ARL-TR-9814 • Oct 2023

Online Learning Approach to Predict Value of Information

by Redford Fox Hudson, Jade Freeman, and Mark Dennison

NOTICES

Disclaimers

ARL-TR-9814 ● Oct 2023

Online Learning Approach to Predict Value of Information

Jade Freeman and Mark Dennison
DEVCOM Army Research Laboratory

Redford Fox Hudson
University of California, Irvine

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED START DATE	3. DATES COVERED END DATE
October 2023	Technical Report	01 October 2022	01 September 2023

4. TITLE AND SUBTITLE
Online Learning Approach to Predict Value of Information

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Redford Fox Hudson, Jade Freeman, and Mark Dennison

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
DEVCOM Army Research Laboratory ATTN: FCDD-RLA-IB Adelphi, MD 20783-1138	ARL-TR-9814

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCID IDs: Jade Freeman, 0000-0003-4085-2235; Mark Dennison, 0000-0001-9780-0511

14. ABSTRACT
The modern battlefield environment presents commanders and analysts with an overwhelming amount of information. Only portions of this information are useful at any given moment, often requiring human intervention to parse out what is meaningful and what is not. In an environment where every second counts, methods for accelerating the presentation of only useful information to decision makers is critical. Machine learning is widely used to predict patterns and outcomes in a variety of applications where data structures are complex and high-dimensional. Supervised learning is a traditional machine learning method wherein the algorithm is trained on a large set of data before performing predictions. On the other hand, online learning is a machine learning technique wherein the algorithm learns incrementally or whenever new data and feedback are available. This work seeks to develop a proof of concept for predicting the utility value of incoming sensor data for a user via an online learning method. It also investigates changes in model performance with respect to hyperparameter configuration and the conditions that cause a user to accept that piece of information on each trial presentation via simulated experiments.

15. SUBJECT TERMS
Military Information Sciences, value of information, online machine learning, information overload, information filtering

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	C. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	19

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Jade Freeman	(301) 394-4962

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures **iv**

1. Introduction **1**

2. Cursor-on-Target Messages **1**

3. Data Sets **2**

4. Dynamic Embedding **2**

5. Classification **4**

6. Machine Learning Implementation **4**

6.1 Neural Network Architecture 4

6.2 Synchronic Training Loop 4

6.3 Model Variants 6

7. Results **6**

7.1 Performance Metrics 6

7.2 Model Variation 9

7.3 Conditions and Balance 9

8. Conclusion **10**

9. References **11**

List of Symbols, Abbreviations, and Acronyms **12**

Distribution List **13**

List of Figures

Fig. 1 Overwhelming information density in an information system in 3-D .. 2

Fig. 2 Embedding layer data flow. An input value is mapped to an embedding table and transformed into a vector embedding. 3

Fig. 3 Neural network architecture. A feature vector containing both categorical and numerical data is split, parsed in parallel recombined or concatenated, and fed into a connected layer, ultimately yielding value of information (VoI). 5

Fig. 4 Synchronic training loop. A CoT message is received by the smart dissemination server and assigned a VoI by the neural network via forward propagation. This VoI is then received by the user and assigned a classification label, which is then used to tune the neural network's weights via backward propagation. This process represents one single feedback cycle. 5

Fig. 5 Results for the performance of four model variants across 2,500 feedback cycles for all four data sets and the first two conditions: 1) hostile and 2) both friendly and air 7

Fig. 6 Results for the performance of four model variants across 2,500 feedback cycles for all four data sets and the last two conditions: 1) both friendly and combat and 2) hostile, both friendly and air or both friendly and combat 8

1. Introduction

The modern battlefield environment presents commanders and analysts with an overwhelming amount of information. Only portions of this information are useful at any given moment, often requiring human intervention to parse out what is meaningful and what is not. In an environment where every second counts, methods for accelerating the presentation of only useful information to decision makers is critical. To maintain situational awareness from a common operating picture (Fig. 1), we require a technique that copes with cognitive and systematic information overload. The technique should enable the information system to learn and adapt to dynamic battlefield contexts and provide the most useful and relevant information accordingly.

Machine learning is widely used to predict patterns and outcomes in a variety of applications where data structures are complex and high-dimensional. Supervised learning is a traditional machine learning method wherein the algorithm is trained on a large set of data before performing predictions. Online learning, on the other hand, is a machine learning technique wherein the algorithm learns incrementally or whenever new data and feedback are available.[1] Hoi et al.[1] discuss a comprehensive survey regarding the online learning method, which includes technical details and advantages of online learning over traditional machine learning methods.

This work seeks to develop a proof of concept for predicting the utility value of incoming sensor data for a user via an online learning method. In this framework, the utility value of a data point is a measure of how likely a user is to accept that data, given the option of either accepting or rejecting it based on their own internal assessment of its usefulness.

This paper describes the development of such a proof of concept. It also investigates changes in model performance with respect to hyperparameter configuration, and the conditions that cause a user to accept that piece of information on each trial presentation via simulated experiments.

2. Cursor-on-Target Messages

Cursor-on-Target (CoT) is an XML-based, tactical messaging schema that supports interoperability within multidomain distributed systems composed of sensor and receiver units.[2] The CoT schema provides an event's type, location, and time. It also provides options for other tactical information such as how the geospatial coordinates of the entity were generated. An event's type can consist of many dimensions and attributes depending on the application and context, but it generally conforms to the military standards for reporting such as MIL-STD-2525C.[3]

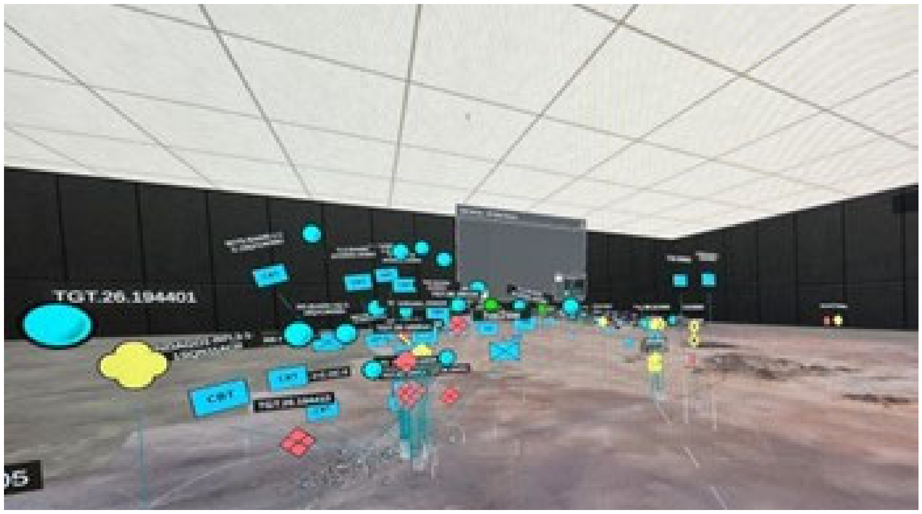

Fig. 1 Overwhelming information density in an information system in 3-D

3. Data Sets

We are interested in developing a model to predict whether a particular incoming CoT message would be useful to the user. To that end, we leverage four data sets, each containing 1,000 CoT messages, which were randomly sampled from an existing CoT data collection from a previous experiment. These messages can represent the incoming sensor data to emulate real-world event monitoring.

For the purpose of this development, the event data have been constrained to contain 20 features: 2 numerical features (range, bearing) and 18 categorical features (type 1–15 and how 1–3). Categorical features of interest included whether the event depicted information from a friendly or hostile entity and whether that entity was a ground, combat, or air-based asset.

4. Dynamic Embedding

The data sets consisted of both categorical and numerical features that cannot be directly used by a neural network for two reasons. The first is that unbounded numerical features in various scales from different domains can result in a slow or unstable learning process, thus disrupting model training. Therefore, data rescaling is implemented as a normalization layer. The second feature is that neurons can only accept numerical data, so the categorical features of the data must be transformed. Although ordinal data (i.e., “high,” “medium,” and “low”) can simply be mapped to a number line, tabular or nonordinal data (i.e., “human,” “car,” and “tank”) are, by definition, a manifold spread across multiple dimensions. Furthermore, they cannot be mapped to a number line without losing information. As a result, a more complex transformation, such as embedding, is required.

Embedding is the process of injectively mapping an input value to a vector embedding—a feature vector containing an arbitrary number of numerical values or weights.[4] This is carried out via an embedding table or dictionary, which has a key-value pair for every possible attribute state. In addition, these weights are tuned during back-propagation and eventually become meaningful representations of each state. By storing an independent embedding vector for every possible state a given variable can occupy, it is possible to represent nonordinal categorical data in a manner that preserves the multidimensionality of that variable. Figure 2 shows the embedding process.

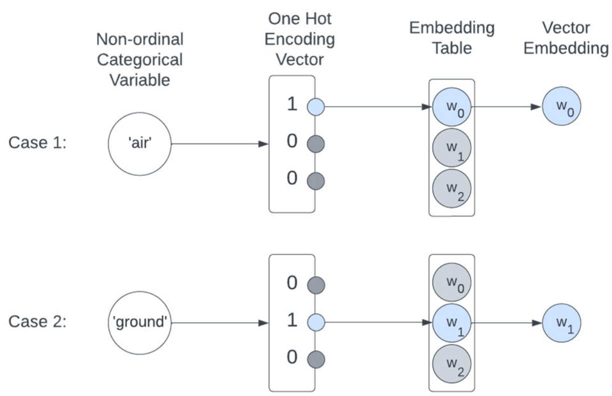

Fig. 2 Embedding layer data flow. An input value is mapped to an embedding table and transformed into a vector embedding.

For every categorical variable, there must be an embedding table. Furthermore, there must be an independent embedding layer for every categorical variable. These embedding layers are placed in parallel with each other at the forefront of the neural network.[5]

Dynamic embedding is an adaptive version of embedding that dynamically expands the embedding table and generates new vector embeddings any time a new attribute state is introduced into the system. (In this implementation, vector embeddings are assigned a single weight that is arbitrarily set to 0 upon creation.) This is useful whenever the number of possible states a variable can occupy is unknown. Because the constraints of the incoming CoT message entity type can vary depending on the application, dynamic embedding is necessary for developing a generic proof of concept.

5. Classification

For every CoT message event, the user can either accept or reject that message to indicate whether this information is of interest to them. This feedback is used as classification labels to train a neural network, where "ACCEPT" messages are encoded as 1 and "REJECT" messages are encoded as 0. In this sense, the classification denotes whether the user regards the message as valuable.

For the purpose of simulating the user's actions, there are four types of events that the user will accept:

1) hostile
2) both friendly and air
3) both friendly and combat
4) any of the above

Three accept/reject conditions will be used in this investigation to simulate the user and generate classification labels. From these user-preferred event constraints, classification labels are generated accordingly.

6. Machine Learning Implementation

6.1 Neural Network Architecture

The 18 categorical features are mapped to numerical values via the embedding layers, as illustrated in Fig. 3. In parallel, the numerical values from the original CoT feature vector are fed through a normalization layer. A new feature vector composed entirely of digestible numerical data is constructed by concatenating the output from all of these transformations. This new feature vector is then fed into a connected layer that outputs a scalar value. This value is passed through a sigmoid function that transforms the output into a probability within the range of (0,1). This final value represents the predicted utility value measuring how likely a user is to accept the incoming data.

6.2 Synchronic Training Loop

This model is not a traditional supervised learning model, because it does not train on batched data; nor is this model a reinforcement learning model, because it does not use a policy gradient or reward function. This model is a hybrid between these two spheres of machine learning, referred to as "online" machine learning, because it conducts incremental learning, wherein the model trains on individual data points as they are received.[6] In other words, the model must "learn on the fly". This allows the model to perform under data-sparse conditions and adapt to changes in the user's demands.

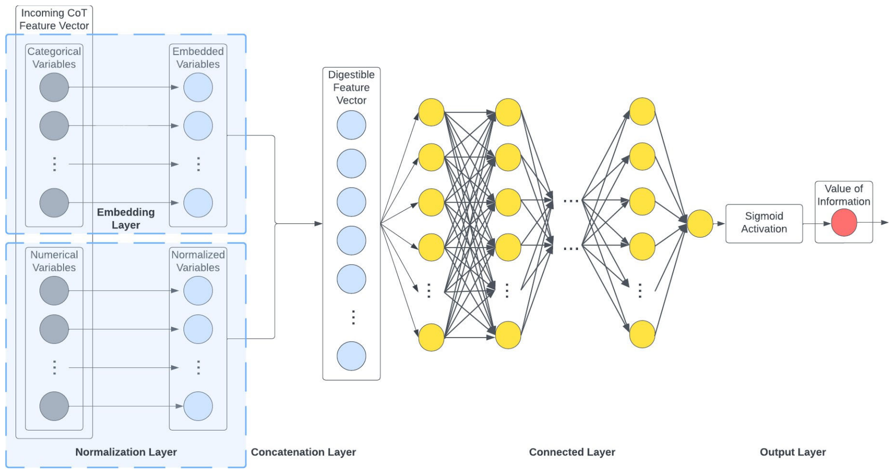

Fig. 3 Neural network architecture. A feature vector containing both categorical and numerical data is split, parsed in parallel recombined or concatenated, and fed into a connected layer, ultimately yielding value of information (VoI).

The model estimates the value of the incoming CoT message via forward propagation, sends this through to the user, and waits for their feedback. When feedback is generated, the model trains incrementally by computing the loss between the feedback (encoded either as 1 or 0) and the corresponding probabilistic estimation, then tuning the weights of the neural network's connected and embedded layers via gradient descent.[7] Thus, the training loop of the neural network is "synchronized" with the user's feedback cycle (Fig. 4).

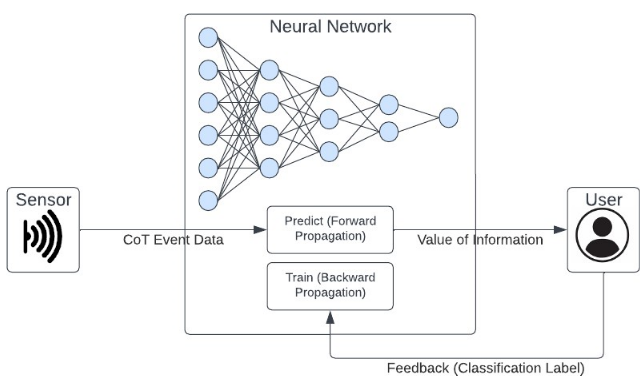

Fig. 4 Synchronic training loop. A CoT message is received by the smart dissemination server and assigned a VoI by the neural network via forward propagation. This VoI is then received by the user and assigned a classification label, which is then used to tune the neural network's weights via backward propagation. This process represents one single feedback cycle.

6.3 Model Variants

To demonstrate the flexibility of this proof of concept, we evaluated the performance of neural network models with various hyperparameter configurations. These models differ with respect to the number of hidden layers, layer dimensions, and number of drop out layers as follows:

1) 1 hidden layer (20 parameters)

2) 2 hidden layers (20 parameters, 10 parameters)

3) 3 hidden layers (20 parameters, 15 parameters, 7 parameters)

4) 1 hidden layer and 1 dropout layer (20 parameters, 25% dropout)

A dropout layer is a layer that randomly removes parameters from the system at a specified rate. This regularizes the data and prevents the model from overfitting. Regularization is very useful for batched training, but it may not enhance performance for online training.

7. Results

7.1 Performance Metrics

We measured the performance of these models across 2,500 feedback cycles. Each cycle represents the presentation of a CoT message to the user, their acceptance or rejection of that message, and the propagation of that feedback to the model. Model performance was measured in terms of error e, or absolute difference between predicted probability p and binary user feedback p^:

$$e = |p - p\hat{}| \tag{1}$$

In addition, the balance b of a data set D, the proportion of CoT messages $m_i \in D$, which satisfy condition c, is a factor that potentially affects the models' performance:

$$b = \frac{|\{m_i \in D | I_c(m_i) = 1\}|}{|D|}, \tag{2}$$

where $I_c(.)$ is 1 if the condition is met, or 0 otherwise.

It is apparent that the error is decreasing for all contexts, indicating that these models are effectively learning to predict an accurate VoI for incoming CoT messages (Figs. 5 and 6). By approximately 750–1,000 feedback cycles, almost all models achieve an error rate less than 5%, after which the model performance plateaus.

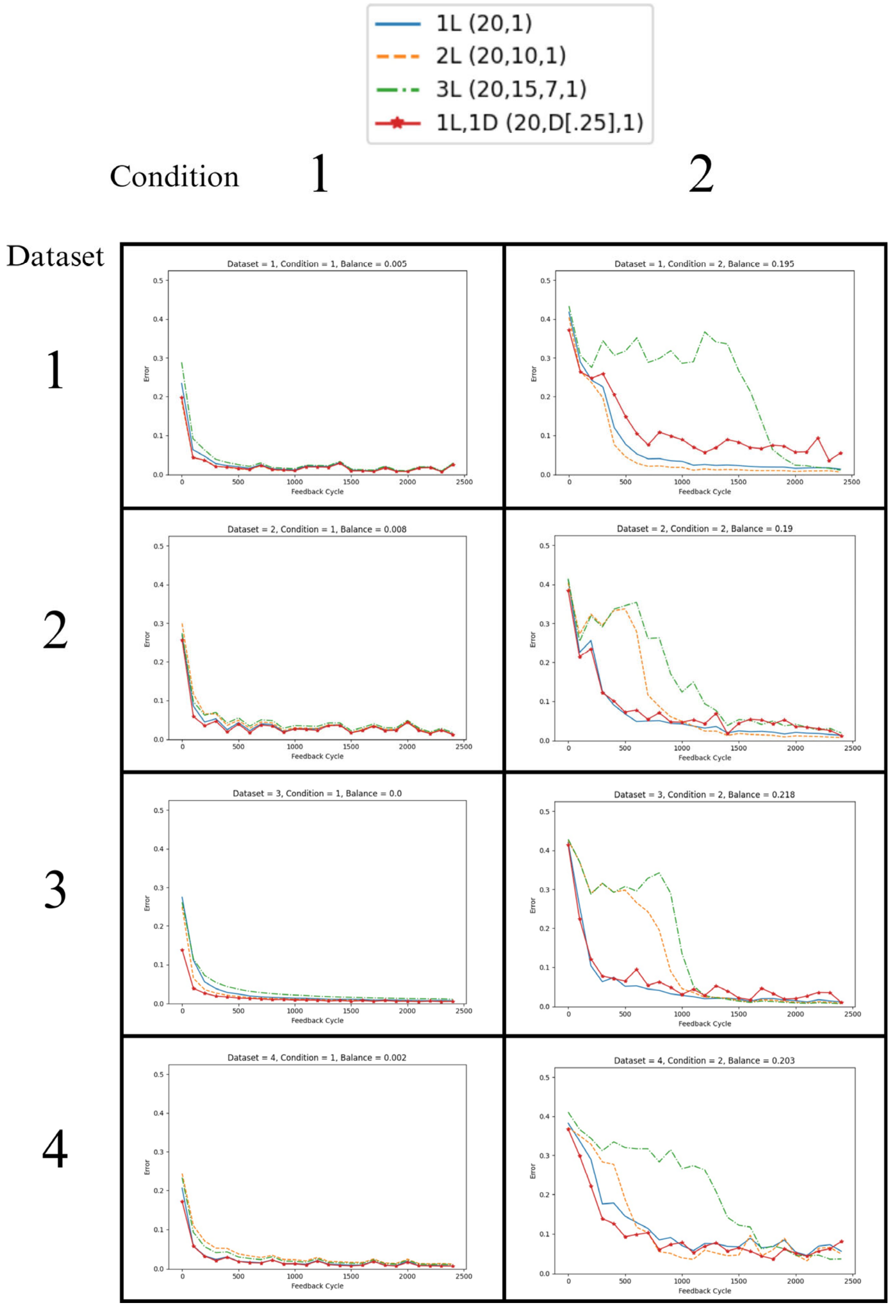

Fig. 5 Results for the performance of four model variants across 2,500 feedback cycles for all four data sets and the first two conditions: 1) hostile and 2) both friendly and air

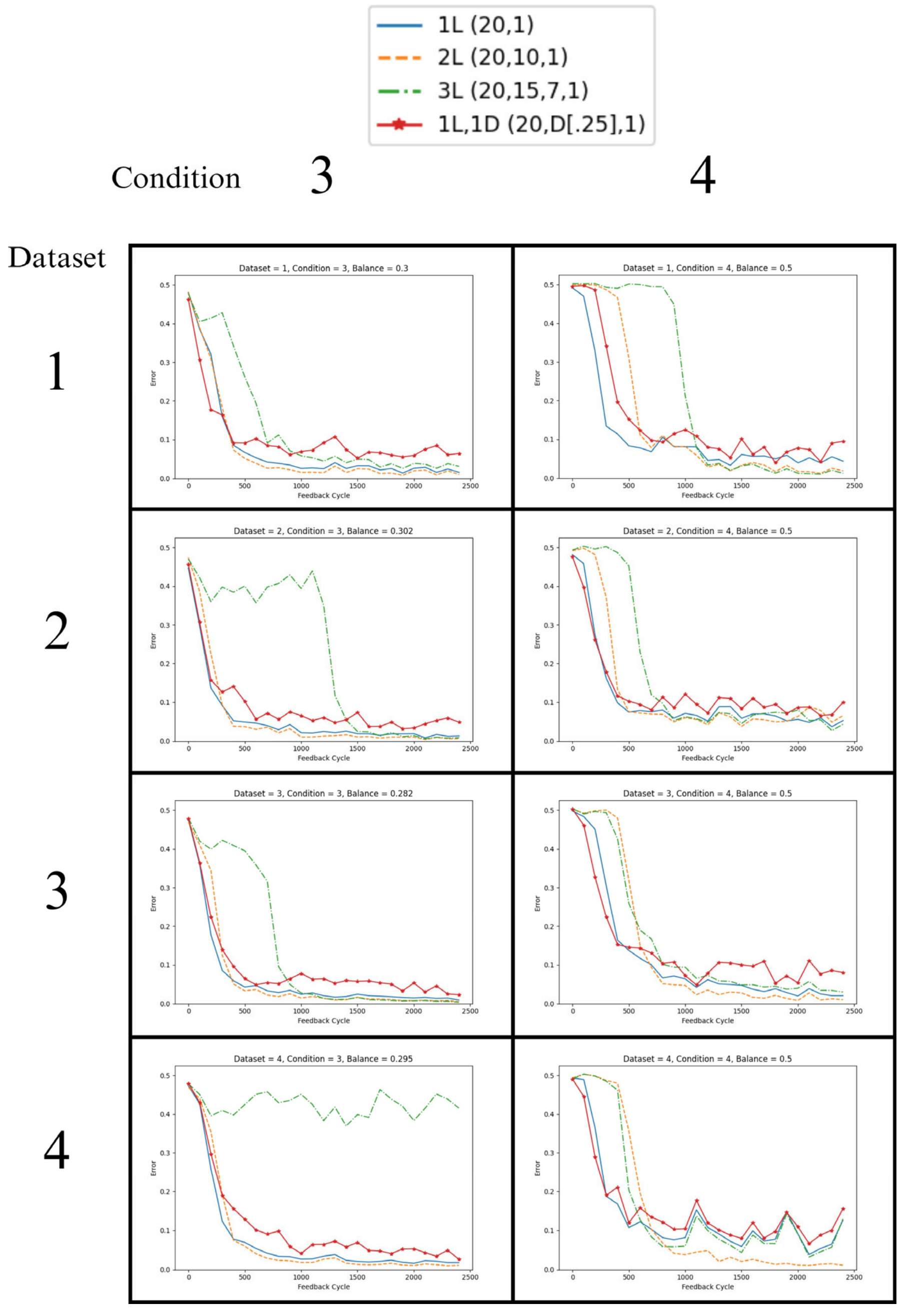

Fig. 6 Results for the performance of four model variants across 2,500 feedback cycles for all four data sets and the last two conditions: 1) both friendly and combat and 2) hostile, both friendly and air or both friendly and combat

7.2 Model Variation

The rate at which the error decreases appears to differ for each model. Initially, we hypothesized that the model with three hidden layers would perform the best, because complexity generally enhances pattern recognition. However, based on these results, the model with three hidden layers (green) is consistently the worst, taking the longest to learn or showing no evidence of improving during the training. Meanwhile, the model with one hidden layer (blue) learns the fastest, and the model with two hidden layers (yellow) has the best long-term performance. These observations suggest that more hidden layers do not necessarily improve accuracy of prediction. This behavior may be because features are independently tested by the conditions, and the simpler models preserve this independence, whereas the more complex models introduce interdependencies between features into the system that actually serve to confuse features and obfuscate the patterns for which the conditions are testing.

We also observed that the dropout layer did not learn as quickly as other models. We hypothesized that the dropout layer would slow down the learning rate, because regularization is generally a hindrance for online training methods. In addition, feature attributes are to be independently learned in the model, and removing features can diminish model predictive capacity, especially if the removed features are significant in the model. Based on the results, the model with one dropout layer (red) is almost universally worse than its counterpart (blue), which supports our hypothesis.

7.3 Conditions and Balance

Both conditions 2 and 3 test two attributes, so it is not surprising that they yield similar performance curves. However, condition 4 tests four attributes, and yet corresponding graphs do not have any features that distinguish this condition from conditions 2 and 3 with respect to model performance.

We hypothesized that a more balanced data set would enable the models to learn faster, because that would provide the model with equal amounts of negative and affirmative feedback from the user and a richer classification landscape from which to learn. However, there is no obvious distinction between contexts with lower balances (0.2–0.3) compared to contexts with higher balances (0.5). However, the models performing in contexts where the balance is extremely low (0.0–0.1) learn very fast. Intuitively, this is because there are virtually no CoT messages in the data set that warrant affirmative feedback, the model consistently receives negative feedback from the user. It is learning to exploit this pattern by uniformly predicting a low VoI. In this case, this result is successful, but in practice there will not be many contexts with a low balance, so this technique is not useful for further application.

8. Conclusion

In this work, we demonstrated that online machine learning is an effective method for predicting VoI of incoming CoT event data. Online machine learning is useful in real-world applications in which the model lacks existing data sets containing comprehensive knowledge from which to learn. In simulations, most of our models were able to achieve below a 10% error rate within 500 feedback cycles without requiring batched pretraining data.

In our cases, we also observed that simpler models with less parameters perform better than models with more parameters. Dropout layers inhibit learning and should not be used. Model performance is unaffected by condition. Model performance is also unaffected by data set balance, as long as the balance is not extreme ($0.1 < b < 0.9$).

We intend to implement our model into the information filtering system architecture to quantify the values of information. We will continue to investigate how the learning speed and accuracy are impacted by model architecture and other potential factors.

9. References

1. Hoi SCH, Sahoo D, Lu J, Zhao P. Online learning: a comprehensive survey. Neurocomputing. 2021;459.

2. Stevenson G, Naiman M, Valenta AL, Boyd AD. Cursor on target: research for a sensor network. Online J Public Health Inform. 2012;4(3).

3. MIL-STD-2525C. Common warfighting symbology. Department of Defense; 2008.

4. Lopez-Martin M, Carro B, Arribas JI, Sanchez-Esguevillas A. Network intrusion detection with a novel hierarchy of distances between embeddings of hash IP addresses. Knowledge-Based Syst. 2021;219:106887.

5. Aakanksha NS. Deep learning for tabular data using pytorch: on a multi-class classification problem. Towards Data Science. 2020. https://towardsdatascience.com/deep-learning-for-tabular-data-using-pytorch-1807f2858320.

6. Rosasco L, Poggio RT. Machine learning: a regularization approach. MIT-9.520 Lectures Notes, Manuscript. Chapter 7 - Online Learning, Dec. 2017.

7. Loeber P. Training pipeline: model, loss, and optimizer. PyTorch Tutorial. 2019. YouTube video, 00:14:15. https://www.youtube.com/watch?v=VVDHU_TWwUg.

List of Symbols, Abbreviations, and Acronyms

3-D	three-dimensional
CoT	Cursor-on-Target
VoI	value of information
XML	Extensible Markup Language

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLB CI TECH LIB
2 (PDF)	DEVCOM ARL FCDD RLA IB J FREEMAN M DENNISON
1 (PDF)	UCI RF HUDSON