AFRL-RH-WP-TR-2020-0134

INTEGRATED DATA DRIVEN SOLUTIONS (I2DS) PROJECT IN THE ACTIVE SOCIAL ENGINEERING DEFENSE (ASED) PROGRAM

Jennifer Neville / Dan Goldwasser / Ninghui Li
Purdue University
401 South Grant St.
West Lafayette IN 47907

AUGUST 2020

FINAL TECH REPORT

AIR FORCE RESEARCH LABORATORY
711th HUMAN PERFORMANCE WING
AIRMAN SYSTEMS DIRECTORATE
WARFIGHTER INTERACTIONS AND READINESS DIVISION
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the AFRL Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RH-WP-TR-2020-0134 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

HANSEN.ERIC.G.1249715502 Digitally signed by HANSEN.ERIC.G.1249715502 Date: 2021.06.03 07:42:11 -04'00'

ERIC HANSEN, DR-III
Mission Analytics Branch
Airman Systems Directorate
711th Human Performance Wing
Air Force Research Laboratory

MURDOCK.WILLIAM.P.1048742161 Digitally signed by MURDOCK.WILLIAM.P.1048742161 Date: 2021.06.10 10:13:14 -04'00'

WILLIAM P. MURDOCK, DR-IV, Ph.D.
Chief, Mission Analytics Branch
Airman Systems Directorate
711th Human Performance Wing
Air Force Research Laboratory

CARTER.LOUISE.ANN.1230249128 Digitally signed by CARTER.LOUISE.ANN.1230249128 Date: 2021.06.21 06:51:41 -04'00'

LOUISE A. CARTER, DR-IV, Ph.D.
Chief, Warfighter Interaction and Readiness Division
Airman Systems Directorate
711th Human Performance Wing
Air Force Research Laboratory

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
15-08-2020	Final	9/17/2018-6/17/2020

4. TITLE AND SUBTITLE
Integrated Data Driven Solutions (I2DS) Project in the Active Social Engineering Defense (ASED) Program

5a. CONTRACT NUMBER
FA8650-18-2-7879

5b. GRANT NUMBER
13000686

5c. PROGRAM ELEMENT NUMBER
HR001117S0050

6. AUTHOR(S)
Jennifer Neville
Ninghui Li
Dan Goldwasser

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
H0X5

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Purdue University
155 S. Grant Street
West Lafayette, IN 47907-2114

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Materiel Command
Air Force Research Laboratory
711 Human Performance Wing
Airman Systems Directorate
Warfighter Interactions & Readiness Division
Wright-Patterson AFB OH 45433

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-RH-WP-TR-2020-0134

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

13. SUPPLEMENTARY NOTES
AFRL-2021-1655; Cleared 27 May 2021

14. ABSTRACT

The Purdue Team's proposal is only for TA1, which focuses on using machine learning models to detect social engineering messages. The Purdue team joined teams led by Berkeley and CMU to form the LASER team. The Purdue team developed techniques to train classification models for social engineering emails, and participated in the dry-run and the evaluations. Three models were developed. Two models analyze the subject and the text in the body. A TF-IDF (term frequency-inverse document frequency) model uses standard term frequency information. A second model extracts motive features from the text to identify the message author's intent (e.g., get information, access social network). A third model is a knowledge and graph model that extracts relation features from the sender and receiver information. An ensemble model aggregates output from the three models to make a prediction, and is comprised of Logistic Regression model and Neural Network model. The team has extensively explored different models, training techniques, and their impacts on accuracy.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Eric Hansen
U	U	U	SAR	26	19b. TELEPHONE NUMBER *(Include area code)*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The Purdue Team's proposal is only for Technical Area (TA) 1, which focuses on using machine learning models to detect social engineering messages. The Purdue team joined teams led by Berkeley and CMU to form the Learning to Automate Social Engineering Resistance (LASER) team. The Purdue team developed techniques to train classification models for social engineering emails, and participated in the dry-run and the evaluations. Three models were developed. Two models analyze the subject and the text in the body. A term frequency-inverse document frequency (TF-IDF) model uses standard term frequency information. A second model extracts motive features from the text to identify the message author's intent (e.g., get information, access social network). A third model is a knowledge and graph model that extracts relation features from the sender and receiver information. An ensemble model aggregates output from the three models to make a prediction, and is comprised of Logistic Regression model and Neural Network (NN) model. The team has extensively explored different models, training techniques, and their impacts on accuracy.

1.0 INTRODUCTION

The Purdue team was led by Principal Investigator (PI) Jennifer Neville, and co-PIs Dan Goldwasser and Ninghui Li. Several graduate students were involved in the project. The Purdue Team's proposal was only for TA1, which focuses on using machine learning (ML) models to detect social engineering messages. The Purdue team joined teams led by Berkeley and Carnegie Mellon University (CMU) to form the LASER team. The team developed techniques to train classification models for social engineering emails, and participated in the dry-run and the evaluations.

We describe the activities we conducted within the project in Section 2.0, Activities, and the classification model we developed in Section 3.0, Model Design. In Section 4.0, Analysis of Results from Last Evaluation, we conducted a detailed analysis of the model with different parameter choices using data from the final evaluation. We conclude in Section 5.0, Conclusions.

2.0 ACTIVITIES

We first describe activities organized by the Active Social Engineering Defense (ASED) Program, including dry-run and two evaluations. We then give an overview of the research on learning and reasoning we carried out in the context of the program, and finally describe activities we proactively carried out to generate training dataset for the program.

2.1. Evaluation-Related Activities

The team participated in all the project meetings and performed the following technical activities for each evaluation.

2.1.1 Dry-Run

- Workshop and PI meeting time: 03/11/2019 - 03/15/2019
- Evaluation time: 03/23/2019-03/29/2019
- Technical activities:
  - Designed a prototype phishing email classifier based on Bag-of-Words model as Purdue team's TA1 component.
  - Designed and implement a prototype system for LASER, consisting of Purdue team's TA1 component, Berkeley team's TA1 component and CMU team's TA2 component.
  - Verified the feasibility and reliability of our prototype system within the evaluation environment.

2.1.2 Fall Evaluation

- Workshop and PI meeting time: 08/19/2019 - 08/23/2019
- Evaluation time: 09/30/2019 - 10/21/2019
- Technical activities:
  - Designed a new Purdue TA1 component based on the Global Vector (GloVe) model and classify messasges based on different motives.
  - Delivered an integrated system that supported more interactions between TA1 and TA2 components of LASER team. Enabled more flexible aggregation for final prediction results of Purdue and Berkeley TA1 components.

2.1.3 Winter Evaluation

- Workshop and PI meeting time: 01/13/2020 - 01/17/2020
- Evaluation time: 02/03/2020 - 02/24/2020
- Technical activities:
  - Upgraded Purdue TA1 component to an ensemble model consisting of motive classifier with GloVe model, knowledge graph based classifier and rule-based whitelist classifier.
  - Rolled out a more robust and efficient system for evaluation.

2.1.4 Research on Learning and Reasoning over Narrative Text

One promising approach to identify social engineering communication is to correctly identify the intent of the messages. Identifying intent in natural language communications requires models that

can reason about the content and automatically infer the connotations and impact of the information conveyed in the text. To accomplish this goal we study, in a sequence of works, different approaches for representing *events* described in text.

An event is a structured object, associating activities and situations with their participants using typed *roles*. For example the text *"Mr. Smith, this is your tax advisor, send me your credit information"* describes two participants, Mr. Smith and tax-advisor, each associated with a different role in the context of a send event. Our models are designed to answer the following question *–in what **context** is this a legitimate request?*

Intuitively, answering this question requires placing the event in the context of prior interactions. We formulate this problem as script learning, i.e., building models that can automatically infer the relationship between events, and identify meaningful event sequences corresponding to intentional behaviors in a given scenario. We tackled three challenges.

2.1.4.1 Multi-Relational Script Learning for Discourse Relations.

Our work studying multi-relational script learning was published in Association for Computational Linguistics (ACL)'19 [3]. Current statistical script learning approaches embed the events, such that their relationships are indicated by their similarity in the embedding. While intuitive, these approaches fall short of representing nuanced relations, needed for down-stream tasks. For example, given an event described in text ("Jenny went into a restaurant"), we would like to identify the reason for the activity, as described in the figure below. In this paper, we suggest to view learning event embedding as a multi- relational problem, which allows us to capture different aspects of event pairs. We model a rich set of event relations, such as Cause and Contrast, derived from the Penn Discourse Tree Bank. We evaluate our model on three types of tasks,the popular Mutli-Choice Narrative Cloze and its variants, several multi-relational prediction tasks, and a related downstream task—implicit discourse sense classification.

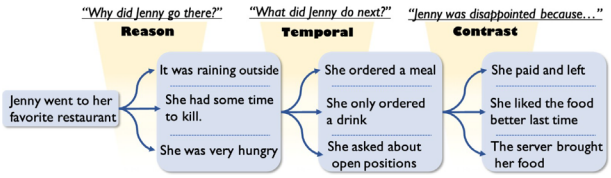


Figure 1. Reason for Activity.

2.1.4.2 Weakly-Supervised Modeling of Contextualized Event Embedding for Discourse Relations.

This work was published at Empirical Methods in Natural Language Processing (EMNLP) '20 [4]. Representing, and reasoning over, long narratives requires models that can deal with complex event structures connected through multiple relationship types. This paper suggests to represent this type of information as a narrative graph and learn contextualized event representations over it using a relational graph NN model.

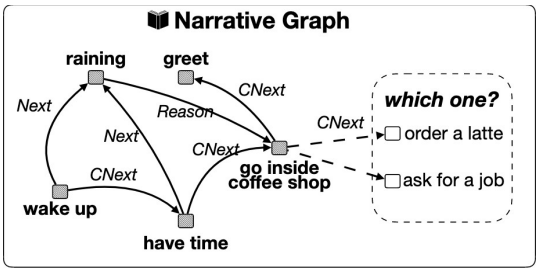


Figure 2. Narrative Graph

We train our model to capture event relations, derived from the Penn Discourse Tree Bank, on a huge corpus, and show that our multi-relational contextualized event representation can improve performance when learning script knowledge without direct supervision and provide a better representation for the implicit discourse sense classification task.

2.1.4.3 Modeling Human Mental States with an Entity-based Narrative Graph (ENG).

This work has been accepted for publication in the North American Chapter of the ACL (NAACL)'21 [5]. We propose an ENG to model the internal-states of characters in a story. We explicitly model entities, their interactions and the context in which they appear, and learn rich representations for them. We experiment with different task-adaptive-pretraining objectives, in-domain training, and symbolic inference to capture dependencies between different decisions in the output space. We evaluate our model on two narrative understanding tasks: predicting character mental states, and desire fulfillment, and conduct a qualitative analysis

2.2. Message Labeling

When we learned from the Fall '19 evaluation that linked-in would be included in the later parts of the program, we realized that lacking training data would be an obstacle to the program. We thus decided to generated labeled emailed and linked-in messages, hoping that the datasets would be useful for the whole program. In the process, we also aim to generate finer-grained label, where messages are labelled with intent so that one can train intent inference models.

- Labeling emails based on intents: 11/2019-03/2020
  - Labeled emails from dry-runs based on their intents, instead of malicious or not. Intents are classified into three main categories and 14 sub-categories in a hierarchy structure.
- Mechanical Turk (MTurk) Labeling for LinkedIn phishing messages: 10/2019-03/2020
  - Designed questionnaire and asked volunteers to generate LinkedIn messages and try to build connection with targeted users.
  - Messages were required with different combinations of motives, strategies and focusing on different information of targeted user's profile i.e., job, education, past experiences, etc.

3.0 MODEL DESIGN

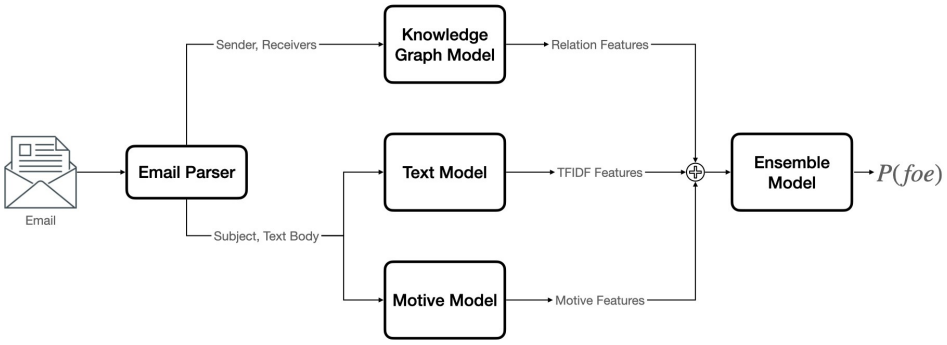


Figure 3. Email Classifier Framework.

The LASER TA1 model is an ensemble of the final Purdue and Berkeley models. The Purdue model takes an email as input and outputs a predicted probability of foe (P(foe)). To produce a classification, we use a decision threshold on the predicted probability, i.e., $\mathbf{if}\, P$ (foe) > threshold then label=foe. In our final model, we use threshold=0.5.
The LASER ensemble uses the following voting procedure:

- If the Berkeley and Purdue models agree, then return the predicted label
- If the Berkeley and Purdue models do not agree, then
  - If one of the models predicts friend, return friend
  - If one of the models predicts uncertain or null, return the other prediction

To support the classification tasks, the Purdue team developed several component models for TA1, including models based on:

1. Text analysis of the words in the email
2. Knowledge graph analysis of sender/receiver interactions in the past
3. Motive analysis using features from text-based models that were learned to identify the author's *intent* (e.g., get information, access social network)

We learned these models based on labeled training data from the Jet Propulsion Laboratory (JPL) Abuse dataset, the fall evaluation, and summer campaigns. The final Purdue model is itself an ensemble of our three component models, which averages the predicted probability from each of the components.

The overall framework is shown in Figure 3. When a new email comes, our email parser extracts sender, receivers, subject, and text body from the email. The knowledge graph model analyzes the interactions between sender and receivers and outputs the relation features based on these interactions. TF-IDF model and motive model facilitate several Natural Language Processing (NLP) techniques to identify keywords and the intent of this email. Finally, the ensemble model concatenates these features as input and infers the probability that the sender is a foe. The ensemble model is comprised of Logistic Regression model and NN model.

3.1 Knowledge Graph Model

The fundamental architecture behind the knowledge graph model is a graph database. We implement Neo4j[1] as our graph database. The graph database reveals data relationships between entities, like sender, receiver, and company. These relationships cannot quickly discover from the traditional relational databases. The graph database can help us answer some questions, like how many times these two users interact with each other in a certain period or how many mutual receivers two senders have in common in the past month. The traditional databases usually have a hard time to fulfill these queries.

We build a sender-receiver graph to facilitate the power of the graph database and discover malicious patterns in the graph to further improve our model accuracy. The graph is shown in Figure 2. The sender-receiver graph is a directed graph. Each node in the graph can be both a sender or a receiver. If a sender sends an email to a receiver, an edge will exist between them.

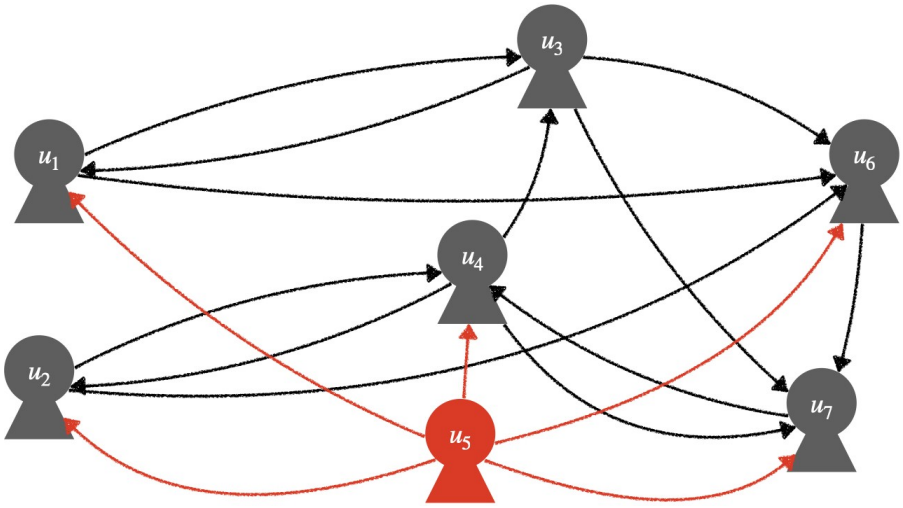


Figure 4. Malicious Pattern in Knowledge Graph.

We observe a malicious pattern that usually happens with malicious senders in our training data. A malicious sender usually sends lots of emails to receivers who do not interact with him/her. For example, user *u*5 in Figure 4 sends emails to many users who never interact with it. Therefore, we use the number of interactions between a sender and receivers as our knowledge graph features. For each email sent or received, we will record the email exchanges in our graph database. Our model will query the graph database every time it needs to predict whether a user is a friend or foe.

3.2 Text Model

Each email in our training dataset contains rich text information. The Text Model can extract keywords from the contents in emails. These extracted keywords may provide some information to classify a sender. It first tokenizes the content into bags of words and calculates the statistical importance of each term across the corpus (all of the email contents). The output TF-IDF feature is the importance scores of the keywords in each email.

TF-IDF shows the statistical importance of a term. The term frequency [6] is as follows:

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}},$$

where $f_{t,d}$ is the frequency of term t in the document d. The term with higher TF values indicates that it is used several times in the document and this might suggest that the term is important in this document. However, there are some common terms, like "this," "tha," "is," "not," etc. which show up frequently in all documents, but they usually provide low information. Therefore, we need inverse document frequency [2] value to reduce the importance of these common terms. IDF is calculated as follows:

$$idf(t) = \log(\frac{N}{n_t}) + 1,$$

where N is the number of documents and n_t represents that how many document has term t. IDF reduces the importance of those common terms because they are in most of the documents. Finally, the statistical importance of each term t given a document is defined as follows:

$$TFIDF = tf(t, d) \times idf(t)$$

Text Model provides the TF-IDF score for each term in a given document as features.

3.3 Motive Model

The email content might reveal the possible motives of senders. The malicious motives may be acquiring the credential information, spreading malware, phishing scam, etc. The motive model targets the motives behind emails and output motive scores as motive features. The final ensemble model adopts these motive features to classify whether a sender is a friend or foe. The 14 malicious motives are shown in Table 1.

Our labeling website1 provides an interface for users to label email motive at the sentence level. Users highlight the sentences that might contain malicious intent and label the malicious sentences with the related motive. Motive Model can learn what the intent is behind each sentence through the labeling result from our website. However, we lack negative samples for each motive. For example, the sentence "please verify your password through the following link" is the positive sample for acquiring the credential. However, the model does not know how a sentence looks like when the sentence is a negative sample for acquiring the credential.

We propose a clustering method to identify the negative sample to solve the negative sample problem. The clustering method focuses on searching those negative samples (sentences) remarkably similar to the positive samples to build a robust classifier. First, we use GloVe [7] to represent each positive sentence having a given motive as sentence vectors in the embedding space. Likewise, we transform the rest sentences in the training data into sentence vectors by GloVe. Then, we calculate the centroid within the group of positive

sentence vectors and identifies top-*k* negative sentences with the sentence vector close to the centroid of the given motive. We train a Binary Logistic Regression model for each motive to output the motive score as the motive feature.

4.0 ANALYSIS OF RESULTS FROM LAST EVALUATION

This section describes the Purdue team's analysis of TA1 performance in the Spring '20 evaluation, both within the LASER system, as well as independently.

Our experience in the fall evaluation indicated that (in lieu of a large labeled training set from the evaluation participants) we should evaluate each of our models components during the winter evaluation in order to assess performance individually. This will inform development of our ensemble methods by providing information about the strengths and weaknesses of each approach, as well as allowing us to assess correlations among their predictions. (Ensembles provide larger reductions in error when the various components have lower correlations between predictions.)

To facilitate this type of analysis, we not only output the prediction of our final ensemble model in our custom stix tag, we also output predictions for 17 alternative models. 15 of these models correspond to different combinations of our components. (See Figure 6.)

Two additional models were included to assess the effects of retraining our final model on the historical data of participants in the winter eval. In this case, we still used the previous labeled data from the program (described above). We simply added to that data a sample of the historical data provided during the evaluation and labeled it as friend. One of the retrained models was designed to use a random sample of historical data. The other was designed to prioritize samples that our previous models labeled as foe with high probability (i.e., difficult examples that were confusing to our model). Due to technical difficulties, when the LASER system went down during evaluation, we lost the second retrained model. In addition, we only recorded partial results for the first retrained models, which used a random sample of historical messages.

We include results for this retrained model plus the 15 components models in our analysis next. See python notebook 2020 Winter Eval Result.ipynb2 to calculate these results directly from JPL output (stix bundles, ledger, and labeled foes).

4.1 Performance Analysis

Overall TA1 performance of the LASER system is reported in the table in Figure 5. We compare the following models:

1. LASER (JPL): Numbers from ASED Winter 2020 Eval Metrics
2. LASER (bundles): Numbers are calculated from ased-laser-ta1 bundle-evaluation 2.json
3. Purdue: Numbers are calculated from the stix output of the final Purdue model (labeled Purdue in this report).

model	total_foe_message	total_foe_prediction	correct_foe_predictions	precision	recall	F1	total_friendly_message	incorrect_foe_prediction	false_alarm
LASER (JPL)	524	11167	221	2%	42%	0.040	50261	10946	22%
LASER (bundles)	440	11067	215	2%	49%	0.037	50095	10852	22%
Purdue	440	24683	223	1%	51%	0.018	50095	24460	49%

Figure 5. Overall Performance.

[1]https://nld.cs.purdue.edu/laser/motive/instruction/

4.1.1 Observations

- The Purdue model has 51% recall, which is quite high compared to the other models
- However, the Purdue model has very high false alarm rate (49%). This is due to a poor choice of decision threshold (threshold=0.5), which we will explore below in Section 4.3.

4.2 Alternative Purdue Models

As described earlier, we evaluated the performance of 16 different models, which correspond to various combinations of our component models, using both NN and logistic regression (Logistic). These experimental results are extracted and calculated from our stix bundles in ased-laser-ta1 bundle-evaluation 2.json.

Table 1. Motive Categories

Acquire Information	Provide Information	PrecisionRequest Actions
Acquire personal identity	Share social network	Click links
Acquire credential information	Share general information	Install malware
Acquire social Network	Comment	Open attachment
Gather general information	Notification	Get money
		Do favor
		Accept social network request

The model names are listed along with the associated type of model (Model) and its components (columns 3-5) in in Figure 4. Note that Purdue model in the first row corresponds to our final model. The Text column refers to whether the model used text-based features (i.e., TF-IDF, unigram and bigrams). The Knowledge Graph column refers to whether the model used counts of previous sender-receiver interactions as features. The Motive column refers to whether the model used intent predictions as features (see Table 1 for categories).

[2]https://drive.google.com/drive/folders/1M D9KN7L-UcZv2btrsh-FF hOsF1pc5u?usp=sharing

	Model	Text	Knowledge Graph	Motive
Purdue	NN+Logistic	True	False	True
laser_winter_logistic	Logistic	True	True	True
laser_winter_logistic_default	Logistic	True	False	False
laser_winter_logistic_kg	Logistic	True	True	False
laser_winter_logistic_kg_motive	Logistic	False	True	True
laser_winter_logistic_kg_only	Logistic	False	True	False
laser_winter_logistic_motive_only	Logistic	False	False	True
laser_winter_logistic_no_kg	Logistic	True	False	True
laser_winter	NN	True	True	True
laser_winter_default	NN	True	False	False
laser_winter_kg	NN	True	True	False
laser_winter_kg_motive	NN	False	True	True
laser_winter_kg_only	NN	False	True	False
laser_winter_motive_only	NN	False	False	True
laser_winter_no_kg	NN	True	False	True
laser_winter_ensemble_mean_no_kg_no_strategy	NN + Logistic	True	False	True

Figure 6. Final and Alternative Model Settings.

Figures 7-8 report the precision/recall/F1/false-alarm performance of each alternative model using our default decision threshold of 0.5.

4.2.1 Observations

- The laser winter model had better performance in general (recall 54% and false alarm 16%)
- The laser winter logistic kg motive model had very high recall (94%), but with higher false alarm rate (32%).

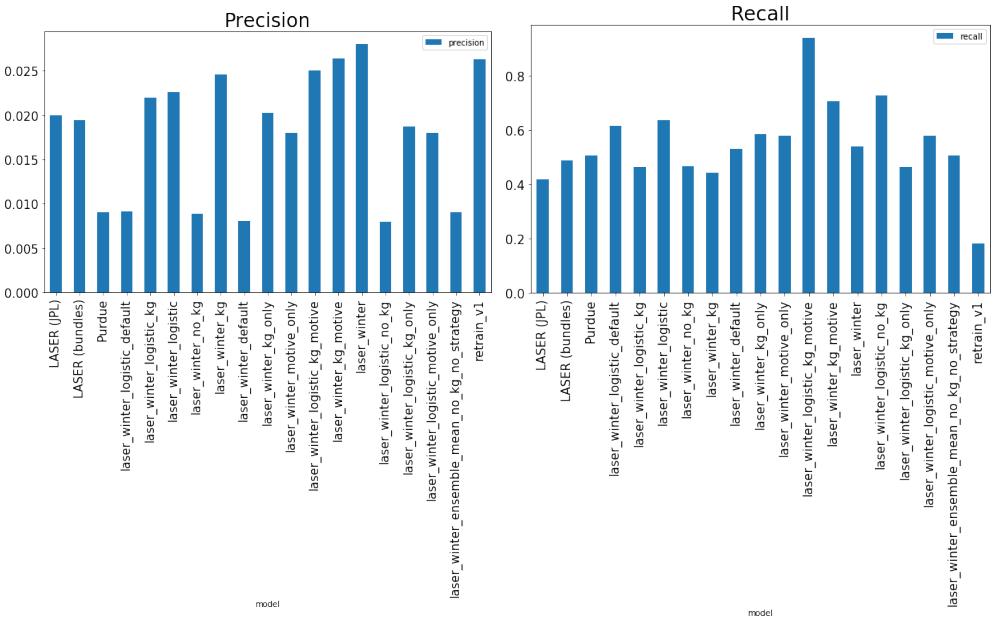
To reproduce the results in Section 4.1 from the python notebook, run (1) Initialization, (2) JPL Result Loading, (3) LASER Result Calculating, (4) Purdue Result Calculating, (5) Output Performance Analysis Table, (6) Load data for 16 models, (7) Output performance figures for 16 models.

4.2.2 Adjustment of Decision Threshold

The above results used a decision threshold of 0.5. However, since all our models all models output *P* (foe), we can explore how adjusting the decision threshold changes the performance of the models. Adjustments to the threshold can be set by (1) prior knowledge of the expected ratio of benign/attack emails, and/or (2) optimizing the threshold on domain data taking into account and costs associated with false positives/false negatives.

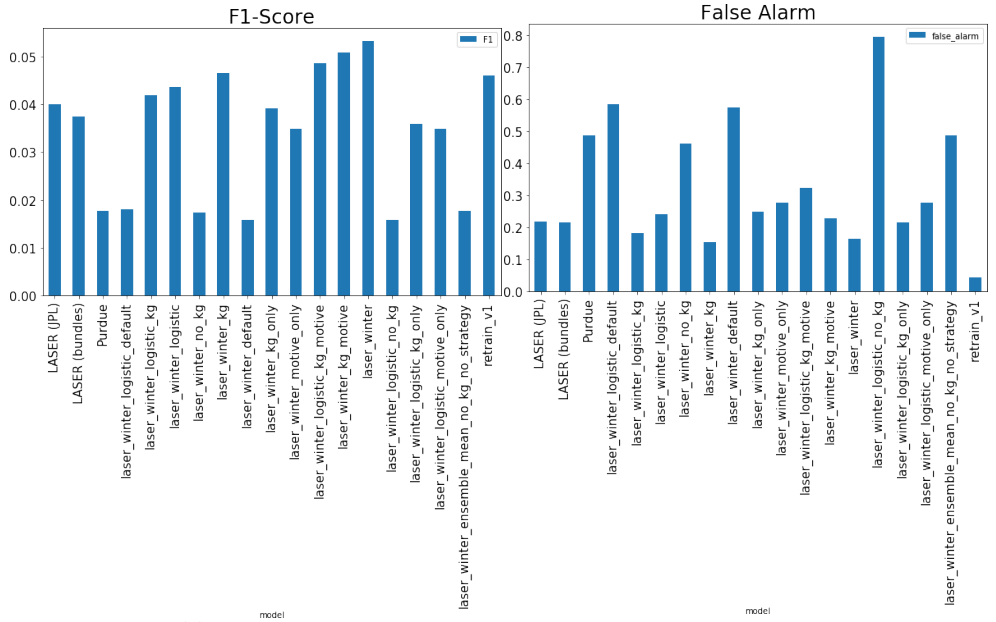
4.2.3 Receiver Operating Characteristic (ROC) Curves

ROC curves plot the true positive rate and false positive rate for all possible decision thresholds. Generally, the shape of the ROC curve shows the quality of the ranking induced by the models predicted $P(\text{foe})$. If a low false positive rate is the priority, one should look at the lower left corner of the plot and select a model with a steep initial slope



(a) Precision. (b) Recall.

Figure 7. Precision and Recall.



(a) F1 Score. (b) False Alarm Rate.

Figure 8. F1 Score and False Alarm Rates.

This indicates that the top of the ranking is very accurate. If recall is the top priority, one should look at the top of the plot and select a model that maximizes true positive rate (TPR) with minimum false

positive rate (FPR). Figure 9 shows the ROC curves for our 16 models.

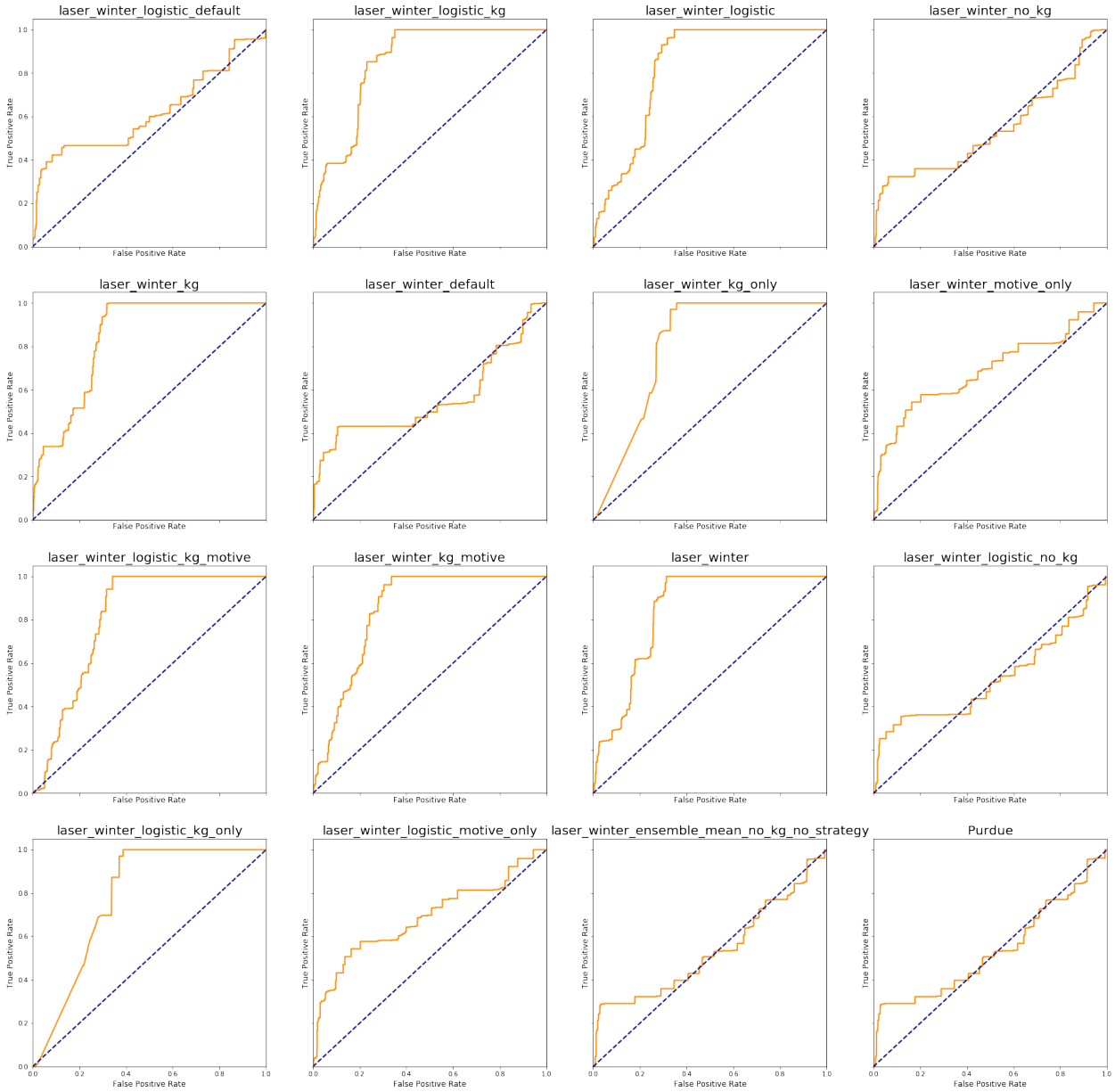


Figure 9. ROC Curves of All Models.

- The dashed diagonal lines show baseline performance (i.e., random predictions).
- The orange curve shows true positive rate and false positive rate for the given model, as the decision threshold is varied.
  - For example, for laser winter kg with threshold=0.99, the true positive rate is 17% and false positive rate is 2%. On the other hand, with threshold=0.90 the true positive rate is 34% and the false positive rate is 5%.
- Larger area under the ROC (orange curve) indicates better performance.

4.2.4 Observations

- Figure 9 shows that our component models have different performance

profiles. This indicates that we could produce a more effective ensemble by weighting the contribution of each component model. (See Section 4.5.) For example,

- laser winter logistic default has a very steep initial slope
- laser winter kg only has 100% recall with moderate FPR
- laser winter motive only has large area early on, indicating a reasonable tradeoff between TPR and FPR

- Figure 10 shows the performance of our Purdue model using an adjusted threshold (chosen to maximize F1). This shows that (without relearning) the model can produce a low false alarm rate, at the expense of reducing recall.

To reproduce the results in Section 4.3 from the python notebook, run (1) Initialization, (2) JPL Result Loading, (3) LASER Result Calculating, (4) Purdue Result Calculating, (5) ROC Curve for 16 models, (6) ROC curves for the retrained model vs Purdue model, (7) Purdue model after threshold adjustment.

model	total_foe_message	total_foe_prediction	correct_foe_predictions	precision	recall	F1	total_friendly_message	incorrect_foe_prediction	false_alarm
LASER (JPL)	524	11167	221	2%	42%	0.040	50261	10946	22%
LASER (bundles)	440	11067	215	2%	49%	0.037	50095	10852	22%
Purdue	440	24683	223	1%	51%	0.018	50095	24460	49%
Purdue (Threshold Adjusted)	440	1059	94	9%	21%	0.125	50095	965	2%

Figure 10. Comparison after Threshold Adjustment.

4.3 Online Retraining with Historical Data

As discussed earlier, we attempted to retrain two models online during the evaluation. Specifically, the Purdue model was trained with (i) 2017 JPL Abuse Emails and (ii) 2019 Dryrun Emails. In our retraining phase, we use (i) and (ii), but also add to that a sample of (iii) 2020 Winter Eval Messages in the first week, which we label as friend.

The retrained model was not preserved properly when the LASER system went down, so we only have predictions for the retrained model on 15, 640 of the evaluation emails. To assess the impact of retraining, we can compare performance to the Purdue model only on those 15, 640 emails in Figure 13.

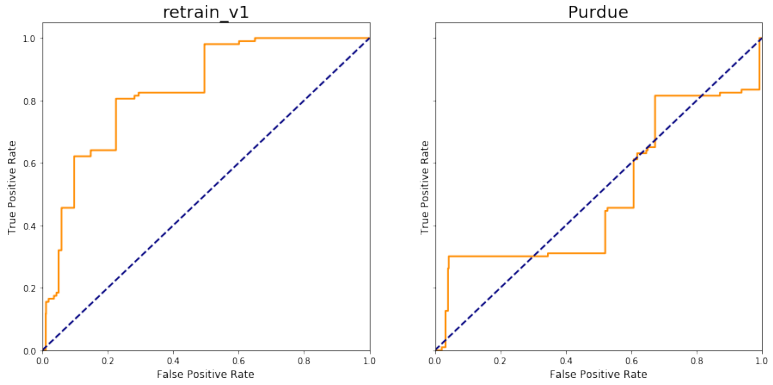


Figure 11. Comparison of Online Learning with Historical Data to Purdue Model Learned Offline.

4.3.1 Observations

- The retrained model has much larger area under the ROC curve compared to the model learned offline. This indicates that learning with a larger dataset that reflects the target users (even if we only add benign examples) can significantly improve model performance.
- We did not evaluate the impact of online learning for the various model components, but we expect that some components will be impacted more than others. For example, the knowledge graph features are more accurate as more interactions are observed.

To reproduce the results in Section 4.4 from the python notebook, run (1) Initialization, (2) ROC curves for the retrained model vs Purdue model.

4.4 Reweighted Ensemble

In Section 4.3, we discussed that the results indicated that a *weighted* ensemble of the 16 components models should improve performance.

To explore this, we trained another model to *learn* the weights and combine the components in an ensemble. Specifically, we used the predicted probabilities for each of the 16 models (extracted from the stix bundles) as features, for the winter evaluation messages in the last two weeks. We considered both logistic regression and decision tree models for the ensemble, and evaluated the new ensemble model with 10-fold cross-validation (using the labels provided by JPL). The average performance is reported in Table 2

Table 2. Reweighted Ensemble Model Trained with 2020 Winter Eval Data.

Ensemble Model	Precision	Recall	F1	False Alarm
Logistic Regression	5%	98%	0.1	16%
Decision Tree	86%	89%	0.87	0.13%

4.4.1 Observations

- The new logistic ensemble model has recall 98% and false alarm rate 16% in average.
- The new decision-tree ensemble model has recall 89% and false alarm rate 0.13%

While the performance of the reweighted decision tree ensemble is very high (recall=89%, false alarm=0.13%), we should note that the model weights were learned using the labels of the winter evaluation data–which we wouldn't have had beforehand. To assess this effect, we also learned a weighted ensemble with only historical data. Specifically, we only use (i) 2017 JPL Abuse (abuse), (ii) 2019 Dryrun (dryrun), and (iii) 2020 Winter Eval Messages from the first week (new historical) to learn the ensemble weights. Table 3 shows the details of the models, training data, and associated performance on the Winter evaluation data.

Table 3. Reweighted Ensemble Models Trained with Historical Data.

Ensemble Model	Data	Precision	Recall	F1	False Alarm
Logistic Regression	**abuse+dryrun**	**3%**	**50%**	**0.05**	**16%**
Decision Tree	**abuse+dryrun**	**2%**	**44%**	**0.04**	**19%**
Logistic Regression	**abuse+dryrun+new historical**	**10%**	**36%**	**0.16**	**3%**
Decision Tree	**abuse+dryrun+new historical**	**6%**	**12%**	**0.08**	**2%**

4.4.2 Observations: Table 3

- Performance is lower for these ensembles, compared to those learned from the true attack data.
- However, the logistic ensemble model achieves reasonable recall of 36% with a low false alarm rate of 3%. We can investigate the weights assigned to each model in the regression to understand the impact of each component.

Figure 12 shows the weights for each model in the logistic regression ensembles. The column old model reports the weights when the ensemble is trained with: 2017 abuse + 2019 dryrun. The column old model+ reports the weights when the ensemble is trained with: 2017 abuse + 2019 dryrun + **2020 historical**. The column new model reports the weights when the ensemble is trained with: 2020 winter eval messages (messages in ased-laser-eval W20 ledger v1.csv).

	old model	old model+	new model
laser_winter_logistic_default_score	2.49	15.17	1.87
laser_winter_logistic_kg_score	4.69	10.33	15.72
laser_winter_logistic_score	-1.74	-8.40	0.12
laser_winter_no_kg_score	-0.47	4.44	7.39
laser_winter_kg_score	6.32	8.28	-2.98
laser_winter_default_score	5.82	3.61	1.08
laser_winter_kg_only_score	0.40	1.34	-0.92
laser_winter_motive_only_score	0.80	3.73	1.49
laser_winter_logistic_kg_motive_score	-1.43	9.46	6.57
laser_winter_kg_motive_score	-1.11	-3.16	1.59
laser_winter_score	-2.71	-4.29	-1.52
laser_winter_logistic_no_kg_score	-1.68	-20.10	-16.81
laser_winter_logistic_kg_only_score	-1.81	-10.46	-10.74
laser_winter_logistic_motive_only_score	0.80	3.73	1.49
Purdue_score	0.03	-1.65	-4.05
laser_winter_ensemble_mean_no_kg_no_strategy_score	0.03	-1.68	-4.21

Figure 12. Comparison Weights of each Model.

4.4.3 Observations: Figure 12

- The relative weighting of the models changes across the ensembles trained from different data. That indicates a possible shift in concept across the various datasets.
- Models are weighted disproportionately in each ensemble, indicating potential correlations and varying impact among their predictions. Since performance is high when the ensemble is estimated on the evaluation data, this indicates that the concept is learnable.

To reproduce the results in Section 4.5 from the python notebook, run (1) Initialization, (2) Logistic Regression ensemble model, (3) Decision Tree ensemble model, (4) Calculate performance of old models, (5) Comparison between old models and new models (6) Output weights of logistic models.

5.0 CONCLUSIONS

The Purdue Team has conducted research in learning and reasoning in natural language text, developed ML models for detecting social engineering attacks, and analyzed the approach in detail. The problem of ASED is important yet challenging. Since the effectiveness of machine learning depends in large part on the quality of data, a major challenge is obtaining effective training and testing datasets. In practice, any such defense techniques and models must evolve over time as the situation changes.

6.0 REFERENCES

[1] Neo4j. https://neo4j.com.

[2] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[3] I.-T. Lee and D. Goldwasser. Multi-relational script learning for discourse relations. In *Association for Computational Linguistics (ACL)*, 2019.

[4] I.-T. Lee, M. L. Pacheco, and D. Goldwasser. Weakly-supervised modeling of contextualized event embedding for discourse relations. In *Findings of the Empirical Methods in Natural Language Processing (EMNLP findings)*, 2020.

[5] I.-T. Lee, M. L. Pacheco, and D. Goldwasser. Modeling human mental states with an entity-based narrative graph. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021.

[6] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4):309–317, 1957.

[7] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL, 2014.

7.0 LIST OF ABBREVIATIONS, ACRONYMS AND SYMBOLS

ACL	Association for Computational Linguistics
ASED	Active Social Engineering Defense
CMU	Carnegie Mellon University
EMNLP	Empirical Methods in Natural Language Processing
ENG	Entity-Based Narrative Graph
FPR	False Positive Rate
GloVe	Global Vectors (for word representation)
JPL	Jet Propulsion Laboratory
LASER	Learning to Automate Social Engineering Resistance
ML	Machine Learning
MTURK	Amazon Mechanical Turk
NAACL	North American Chapter of the ACL
NLP	Natural Language Processing
NN	Neural Networks
PI	Principal Investigator
ROC	Receiver Operating Characteristic
TA	Technical Area
TF-IDF	Term Frequency-Inverse Document Frequency
TPR	True Positive Rate