**Submission for GSA Artificial Intelligence (AI) Machine Learning (ML) EULA Challenge 2020**

## 1. Contact information for Official Representative:

**Name:** Andrew Mendez
**Email:** interativetech1@gmail.com
**Team Name:** AndrewMendez
**Link to Solution Demonstration:**https://youtu.be/47UOL22l3lQ

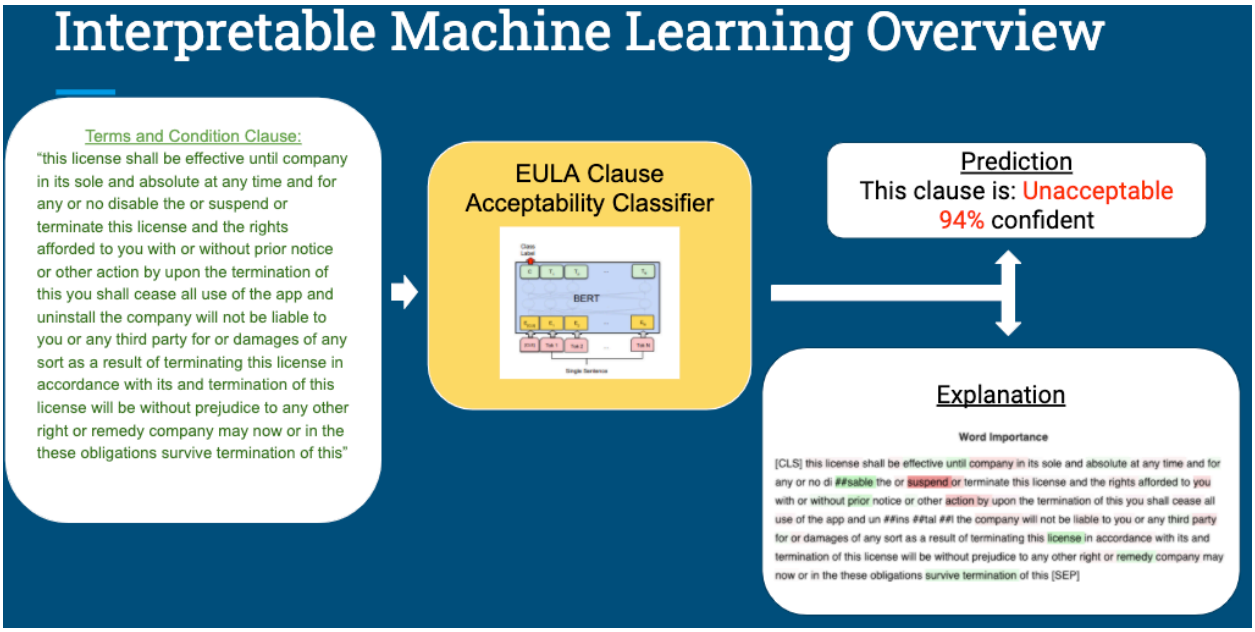
## 2. Names of additional team members:

## 3. Introduction to Team:

My name is Andrew Mendez and I am a machine learning engineer working in the defense industry (CACI Inc.). My passion is applying Machine Learning to national security, government, and cybersecurity problems. I have 2 years of Machine Learning Engineering experiencing at CACI, where I have worked on applying machine learning to our customer's most challenging problems. Before at CACI, I received my masters in information science at Cornell Tech, and a Bachelors in Computer Engineering at the University of Central Florida.

## 4. Executive Summary of Solution:

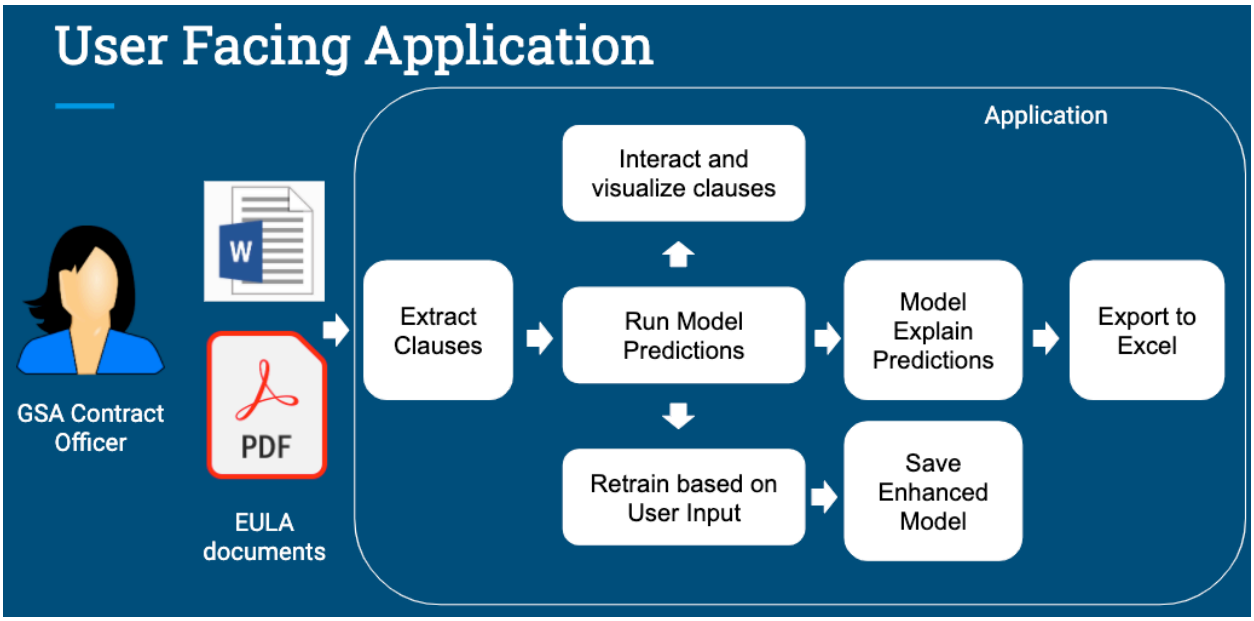
Figure 1




Our team of one developed an Interpretable Machine Learning Model that predicts if a clause is acceptable or unacceptable. We leverage a model agnostic interpretation

method to have the model explain its reason why it classified the input clause as acceptable or unacceptable[3]. Figure 1 shows the overview of the interpretable machine learning model.

Figure 2.




Our team of one also developed an intuitive, interactive application that allows users to input .pdf and .docx EULA documents and predicts the acceptability of the clauses in the EULA document. The app allows the user to see the reasoning behind the models predictions. Our application also supports the user being able to retrain the model based on user input. Figure 2 defines the overall functionality of the application.

To see solution demonstration in action, click the unlisted youtube video: https://youtu.be/47UOL22l3lQ

## 5. Interpretable ML Architecture:

### a. Technology Scope:

- At a high level, what technologies does the solution use? (e.g. language, frameworks).
- We use programming language Python 3.6
- We used Deep Learning framework Pytorch
- We used natural language processing framework HuggingFace[5] to fine-tune SOTA language model BERT
- We used data science libraries Pandas and Matplotlib to load, preprocess data, and visualize data

- We used Model Interpretability library Captum[6] to enable trained model to explain its predictions
- We use python libraries PDFMiner and python-docx to extract clauses from PDF and Word (.docx) documents respectively.

### b. Functionality and User Interface:

- What type of user interface does the solution provide (e.g. web interface, command line interface).
- We provided python web application framework for users to leverage the Machine Learning Model using Streamlit [4]
- The input format supports PDF and MS Word (.docx) documents
- What input formats does the solution support? (e.g. PDF or MS Word).
- Currently the solution processes one PDF file or MS Word file at a time, but the solution can handle batches of excel and .csv files when retraining.

### c. Application of Artificial Intelligence/Machine Learning (AI/ML):

- Provide a description of the ways in which the technology leverages AI/ML. Please specify general approaches (supervised, unsupervised) and conceptual description of how these apply to the challenge.

Our solutions leverages supervised learning to train our Interpretable ML model to predict if a clause is acceptable or unacceptable. Supervised learning is suitable for this challenge because the dataset GSA provides has both input data, and the corresponding ground truth labels generated by a human expert. Supervised Learning works really well for this challenge because supervised learning allows the model to learn from human experts by comparing their predictions to the ground truth labels. Our deep learning based method learns in a supervised manner using gradient optimization. We leverage gradient descent by first measuring the distance between the models predictions to the ground truth labels. The distance or loss gets back-propagated throughout the layers of the network so the model can correct it decisions and do better the next trial.

Another benefit with supervised learning is that we can utilize Transfer learning. Transfer learning is a powerful technique to leverage knowledge from a large dataset to efficiently learn on new, domain specific data. We utilized transfer learning to efficiently learn from the GSA dataset, where the data is limited.

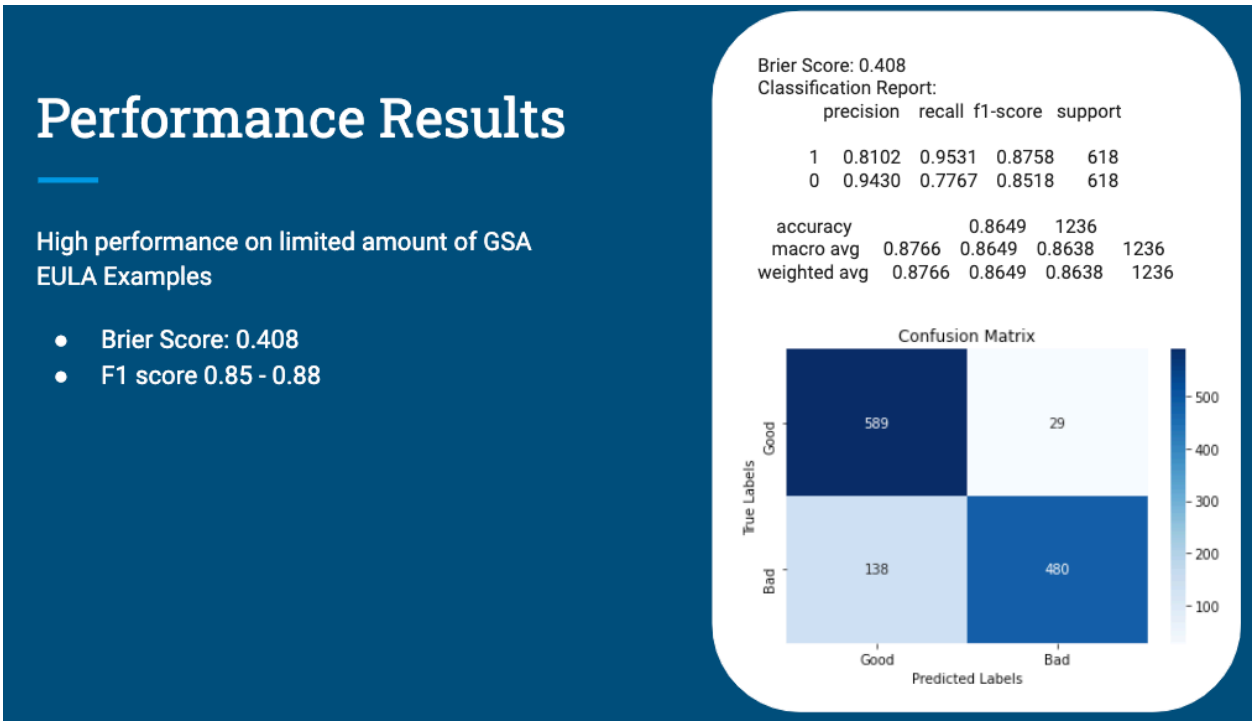
We leveraged a State of the Art language representation model called Bidirectional Encoder Representations from Transformers (also known as BERT[1]) pre-trained on 2.5 Billion records of Wikipedia English articles[2]. We used Transfer Learning to fine-tune our Language Model for binary classification on the EULA Training v1 dataset.

Data Preprocessing and Training Details:

Our training phase is comprised of preprocessing the training data and splitting the dataset into validation and test sets to assess model's generalization. We employ a

function that removes nonsense words, lowercases words, and removes clauses that are less than 2 characters. We have the optional setting to deploy removing stop words and Porter Stemming, however we wanted to keep stop words as stop words like “not” and “unless” could be strong words that determine a clause’s unacceptability. After preprocessing the data, we employ an 80, 15, 5 dataset split where 80% of the dataset goes to training, 15% goes to validation, and 5% goes for a test set. We use the train and validation dataset to tune the model, and use the test set to assess model’s generalization. We do not use the generated test dataset for model improvement or training.

Figure 3.




Below is a Diagram that displays the final performance metrics. The final performance metrics include a Brier Score of 0.408, and F1 score between 0.85 and 0.88. View Figure 3 for F1 metrics details.

Interpretability in our Machine Learning Model:

We leverage an interpretable method to allow the model to explain its prediction. We used an attribution based method called Layered Integrated Gradients[3]. Layer Integrated Gradients allows us assign an attribution score to each word/token embedding tensor in the clause. The method works well for deep learning based

models, and is agnostic to any specific deep learning architecture.

Each token gets an attribution score. Positive Attributions mean that the words/ tokens were positively attributed to the models's prediction. Negative Attributions mean that the words/tokens were negatively attributed to the models's prediction.

User Facing Application:

Figure 2 overviews the functionality of the User Facing application. The trained Interpretable ML model gets loaded into the application, and the user is enabled to upload .docx and .pdf EULA documents to analyze acceptability . The application extracts clauses from the EULA documents and allows a user to interactively explore all the clauses the system extracted from the document. The user then has the option to run the interpretable ML model through the extract clauses, and see the predicted acceptability and the model's confidence in its prediction. The user then has the option to select a predicted unacceptable clause and have the model explain its prediction.

Finally, we developed functionality to allow the user to retrain the model based on user input. The user is required, at this time, to prepare their dataset in a directory. The directory should contain a list of csv files. The csv files must contain the ground truth labels as well. Once the directory is created, the user can input the path to the directory. The user can interactively explore the uploaded data, and can run retraining from end to end. Once the model is trained, the model is saved in a directory the user specifies.

Predictions made by Validation file:

Explaining why model made each prediction on the Validation dataset would be difficult because the model interpretation analysis takes 3 minutes per clause. Ideally, we would of ran the Model Interpretation analysis on all the predictions, and we could see what are common keywords that positively attribute to the model's prediction. For now, there are common terms that attribute to unacceptability. One example of a common keyword that has high attribution with clause unacceptability is "Termination".

References

[1] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[2] https://huggingface.co/bert-base-uncased

[3] Sundararajan, Mukund, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks." *arXiv preprint arXiv:1703.01365* (2017).

[4] https://www.streamlit.io/

[5] https://huggingface.co/

[6] https://captum.ai/