Report - Fraud detection challenge

- Data:

1. **Data Exploration and Preprocessing:** As we know the dataset consisted of bash command history segments from 40 users, each with 15,000 commands. We segmented these commands into groups of 100, resulting in 150 segments per user. The first 50 segments were used for training, while the remaining 100 segments contained both genuine and masquerader commands were also used for training. We read and segmented the commands for each user, storing them along with their labels.
2. **Data Augmentation:** To enhance model robustness, we generated random command segments by randomly selecting commands from the existing set, diversifying our training data
3. **Feature Engineering:**
    a. Vectorization: The commands were vectorized using 'CountVectorizer', converting text data into numerical features by counting the frequency of n-grams (combinations of words) within a specified range, which allows the model to capture patterns based on the presence of these n-grams.
    b. N-grams: We utilized n-grams (ranging from 2 to 5) to capture the sequence and context of commands better, improving the detection of masquerader behavior. N-grams help understand patterns and dependencies between commands, as certain sequences may be common for genuine users but uncommon for masqueraders.

- Anomaly Detection Models:

1. **LSTM Autoencoder:** To learn the user command patterns, with higher reconstruction errors indicating potential masquerader segments we built an LSTM.
   The LSTM architecture has hidden layers sizes of 64, a dropout rate of 0.2, and was trained for 25 epochs with a batch size of 32.
2. **Random Forest Classifier:** A Random Forest classifier was trained on vectorized command segments to predict segment anomalies. It used 2000 trees (estimators) and a random state of 25.
3. **Threshold Calculation**: We applied a threshold for each user based on the scores (the model predictions). The threshold was determined using the median and interquartile range (IQR) of the combined anomaly scores. This approach helps identify the most anomalous segments by setting the threshold to the median plus 1.5 times the IQR.

- Evaluation and Inference:

1. **Evaluation:** We used Accuracy, F1, Recall and Precision. The evaluation was per user.
   We divided the 10 first users for 6 users in training and 4 for validation.
2. **User Vectors:** For each user, we computed a vector based on combined anomaly scores. We gathered for each user high entire scores over the samples and including the mean, and standard deviation scores. This helped us understand the user's command patterns. We then used this user vector method in order to match the pattern of the commands to a user.
3. **Inference:** For each user (10-39) we ran his 100 chunks, we got the predictions from the RF model, calculated the user vector, we took the best threshold we calculated in phase 2, sorted the probabilities of the sample to be a masquerade, and then take till the 10 predictions, and classify them as masquerade (1), and the rest as benign (0).