1. Title Section:

- Anomaly Detection in Network Traffic
- Andrew-Jacob Santos, Daniel Flores, Justin Cruz

2. Abstract (10%):

- This project aims to develop machine learning pipelines that aid in detecting network anomalies/attacks by creating a predictive model that distinguishes the “bad” connections from “good” connections. This is done through data preprocessing, implementation of several machine learning algorithms, and comprehensive evaluation of said models.

3. Introduction (10%):

- Given the prominence of the growing digital world and its relevance to people's daily lives, protecting computer networks from things like unauthorized access and malicious activity is a necessity. Network Intrusion Detection Systems play a crucial role in looking for suspicious or harmful behavior. These systems can detect a wide range of threats, including DOS attacks, unauthorized access, malware, and more. Anomaly detection is especially important because it can detect previously unknown attacks that are yet to be detected, which can make them much more flexible and optimal in many cases. Machine learning algorithms are often used to implement as well as improve these anomaly detection systems by learning normal patterns from data and flagging potential intrusions if abnormal behavior is spotted.
- The 1999 KDD Cup dataset is one that was used for the third International Knowledge Discovery and Data Mining Tools Competition. The task of the competition was to detect network attacks by detecting anomalies in connection. This dataset consists of thousands of distinct connections between two IP addresses. More specifically, each row in the dataset represents a network interaction between the two. Within each connection or row in the dataset, there are over 40 features (columns) that describe them. Each connection within the dataset is ultimately labelled as either a normal connection or one of multiple types of malicious attacks. Overall, this dataset has essentially been used as a benchmark to better understand and evaluate machine learning models' efficiency and usefulness in detecting network anomalies.
- Our main objective for this task is to make use of multiple machine learning models to help us analyze the data within the KDD cup dataset to be able to accurately identify anomalous network activities that may suggest potential security threats. This will be done using 3 different machine learning algorithms along with data reprocessing beforehand to ensure more accurate results from our algorithms, as well as a proper comparison and evaluation between the three models.

4. (Optional) Related Work:

- **Summary of existing research and methodologies in network anomaly detection**
    - Deep learning-based techniques like RNN and CNN have been applied to model intricate patterns in network traffic
    - These models thrive in capturing complex temporal and spatial features to enhance the detection of anomalies

    - Used both unsupervised and supervised techniques, including clustering (K-means, DBSCAN), and classification models to detect point anomalies within network traffic
    - K-Means is effective in partitioning data, while DBSCAN detects anomalies in arbitrary shapes, making it more robust to varying density distributions
    - SVMs classify data by finding an optimal hyperplane that separates normal and anomalous network traffic
- **Discussion on how your approach aligns with or diverges from previous studies**
    - Our approach seems to align with past studies, mainly the model selections they used and what we plan on using. These models, like SVM, will help depict easily noticeable anomalies (attacks) from secure connections by establishing a clear margin.. Many approaches also seem to make use of random forest algorithms to help with accurate labeling of each of the given connections in the dataset. The popularity of the random forest algorithm is mainly due to the algorithm's ability to efficiently handle a large number of features whilst still giving accurate labels and predictions, which is why we decided to use the same algorithm as well.

5. Methodology (60%):

- **Plans and justification for data preprocessing (20%)**
    - We will first make sure to drop any irrelevant or duplicate features with identical data types in order to get better and more accurate outputs from our machine learning models.
    - In order to make a clear consensus of good connections vs harmful ones, we will label the data connections into two categories: normal and attack.
    - We will then encode any features that would be of need using One-Hot Encoding in order to be able to convert data into usable and readable numeric values that can be compared.
    - We will normalize numerical features using a MinMaxScaler for KNN and SVM to ensure a proper distance measurement.
- **Plans and justification for model selection (30%)**
    - One model that we will make use of is an SVM due to the number of features/ dimensions within the given dataset. By using an SVM for at least one of our models, we can attempt to establish a clear margin between attacks/good connections.
    - With an SVM, we would attempt to make use of a nonlinear kernel to help handle any complex patterns in the dataset
    - Using an SVM may be optimal and beneficial for our use since SVM is known for being able to efficiently handle data sets with a larger amount of features. Not only this but an SVM is also known to be good for binary classification, which appropriately aligns with our goals for this dataset, as we are simply trying to make accurate predictions between attacks and normal connections between two addresses.
    - An SVM will create a max margin hyperplane, which helps in clear separation between normal and anomalous activity.
    - To implement an SVM, we can make use of Sklearn.

    - Another model that we plan on using is the KNN algorithm, since it performs well with proper feature scaling and can make classifications based on similar or grouped data points
    - Using the KNN algorithm can help us scale and make use of scaled features.
    - In using the KNN algorithm, with the right distance metric, we can attempt to capture any crucial nonlinear relationships.
    - A final important algorithm that we will make use of is the random forest algorithm, since this algorithm is another one that is suitable for datasets with a large number of features
    - The random forest algorithm is also one that is good for handling datasets with features that include both categorical and numerical features. This is appropriate for our use since the KDD cup data sets contain multiple of each of those data types.
    - This algorithm can also be beneficial for our use since it will allow us to analyze a more raw version of our dataset, since it will not require any heavy scaling of our dataset.
    - Using this algorithm will also be beneficial to us since it naturally makes use of certain features based on their importance in the algorithm's predictions. This means that redundant features will not significantly impact our predictions and outputs when using this model since the random forest algorithm uses random subsets of features for each tree.
    - This method is also known to be able to handle mixed data types well, and can be good for possibly using for feature importance analysis. Not only this, but the algorithm can also be useful since it will be able to reliably resist overfitting.
- **Plans for evaluation metrics (10%)**
    - For each model, we will calculate the accuracy score of our output. This will measure the overall correctness of the model by comparing correct predictions to total predictions.
    - For each model, we will also calculate the recall score of our outputs. This will measure the number of already identified attacks that were correctly identified by the model. This will be important for us in order to compare the amount of false negatives given by our models so that we can choose one that misses the least amount of threats/attacks.
    - For each model, we will also calculate the f1 score of our outputs. This will essentially give us a means of our precision and recall scores. By comparing our models using this score, we can make distinct evaluations of both false positives and false negatives with this score.
    - All of these metrics will be computed for each model to evaluate its ability to distinguish normal connections vs bad connections. We will be comparing each of the scores for all three of the models in order to help us determine which model is most accurate across all means in making these determinations. These metrics can also give us a better understanding of each model's strengths and weaknesses in analyzing the data set to make predictions, which can help us either tweak our model implementations or choose one over the other.

    - We will also make use of a confusion matrix for each model to give us a clear visual representation of how each model stands in its prediction accuracy.

6. Task division (20%):

- **Andrew Santos - Data Preprocessing and Evaluation**
    - Handle the data cleaning, categorical encoding, feature scaling, and handling imbalanced data
    - Provide an in-depth comparative analysis between models by implementing evaluation metrics like interpreting F1 Score, Precision, Recall, and Accuracy
- **Justin Cruz - Documentation and Visuals**
    - Providing the project overview, mainly addressing the problems and objectives
    - Explaining the process/steps taken in data preprocessing, model development, and model evaluation
- **Daniel Flores - Model Selection/Development**
    - Will handle a majority of the implementation of the 3 machine learning algorithms on the KDD cup dataset
    - Ensures that our machine learning models can properly and efficiently give us predicted labels that we can use for evaluation.