Transfer Learning for PCAPs Progress November

Completed:

- State of the art analysis, all useful papers that have been used are linked on GitHub
- PCAP conversion into CSV or images of custom size and depth (if seeking multiple channel images, suited for NN analysis)
- Baseline scores with no transfer learning for SVM, KNN, Autoencoder (can do more as the data does not need to be modified to apply other baseline classifiers)
- Initial code for TrAdaBoost algorithm, an instance transfer model to make the most use of the data available (no modification done to the data, simply a boosting algorithm suited for transfer between two domains through statistics of the features present)
- Initial code for domain adaptation using an autoencoder to modify the learned representation across the source and target domain. For the moment this only works for MNIST dataset which is in a .mat (matlab) format.
- Initial code for Transfer Component Analysis (TCA) which is a form of domain adaptation (representation learning)

In Progress:

- Adapting TrAdaBoost, TCA, and TLDA to accept both images and CSV files as input
- Combining the access to these algorithms within a singular file where all preprocessing will occur at the same time to prepare for benchmarking the methods
- Further preprocessing on generated flow information, view all statistical information for the 79 features of the CSV to see if there are some redundant features (eg. if two features have a correlation close to 1.0 we can remove it). This may also lead to some extra information about what the images can represent for a more meaningful data point.

To do:

- Tool to streamline PCAP to CSV? For the moment there are multiple steps across different tools to go from raw PCAP -> separate CSV files
- Latent space representation using Principal Component Analysis to reduce to 2d representation. This is dependent upon the approach being used so it cannot be done first. For example, the latent space representation of an autoencoder would be different to the latent space in a deep neural network.