CSE 676 – B Deep Learning Spring 2025

SEMESTER PROJECT REPORT

DL for Network Traffic and Cybersecurity Analysis

Team Member	UB Person Number	UBIT
Muhammad Waseem Thameem Ansari	50606269	mthameem@buffalo.edu
Janani Chalapati	50592361	jananich@buffalo.edu

PROJECT DESCRIPTION

This project is on the topic of "Deep Learning for Network Traffic and Cybersecurity Analysis" with the application of state-of-the-art neural network architecture. The aim is to design and experiment with various deep learning methods for network intrusion detection and anomaly detection in network traffic data.

This project entails the implementation and experimentation of three different autoencoder models (Vanilla, LSTM, and Transformer-based) for identifying malicious network activity through learning normal traffic behavior and identifying anomalies. This method enables the identification of both recognized attack signatures and possibly unidentified threats through the detection of behavioral anomalies instead of targeting attack signatures.

DATASET CHARACTERISTICS

Size and Composition:

- Over 2.8 million instances were captured over 5 days (July 3 to July 7, 2017).
- Includes normal traffic and various attacks: Brute Force, Heartbleed, Botnet, DoS, DDoS, Web Attack and Infiltration.
- A highly imbalanced dataset with a majority of records labeled as 'Benign.' (normal traffic)

Data Features:

- 79 columns with 78 numerical features and a categorical 'Label' column.
- Features include network flow characteristics such as flow duration, packet lengths, ports, flags and more.

PROJECT STRUCTURE

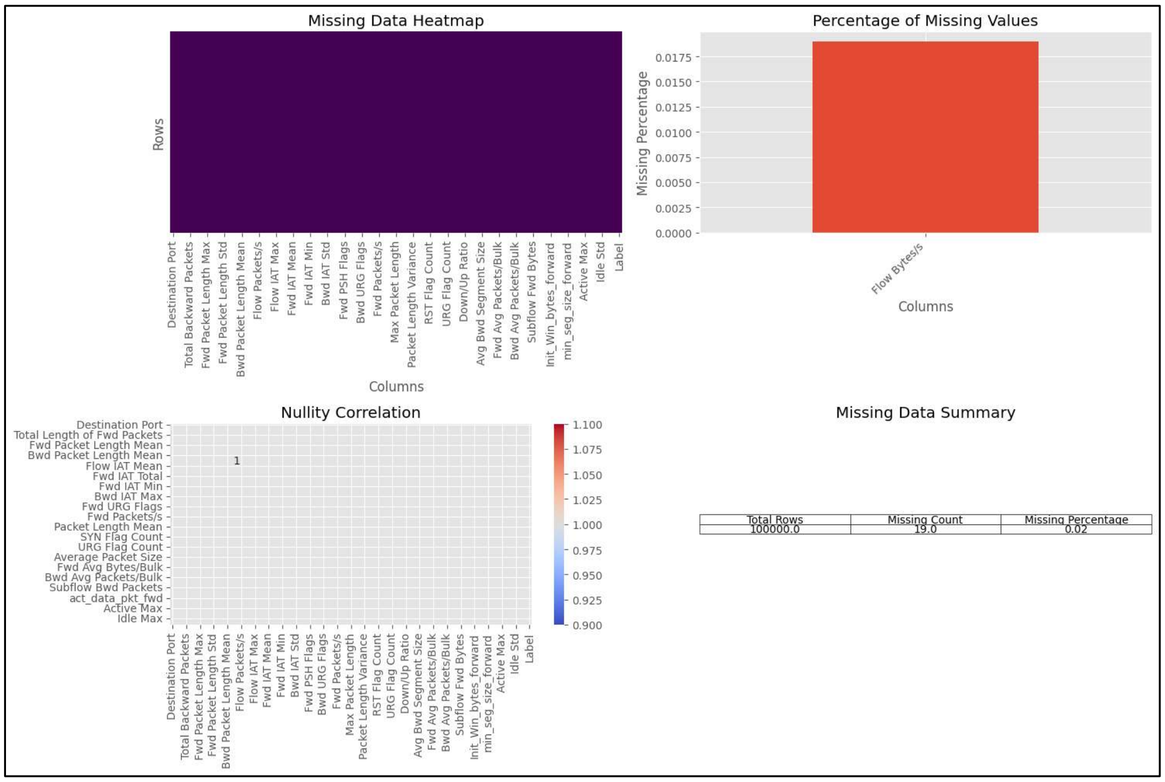
The project is organized into six primary components:

Exploratory Data Analysis	Discovering dataset features and distributions.
Data Preprocessing	Handling missing values, choosing relevant features, and applying data transformation.
Data Analysis	Feature importance determination and statistical analysis.
Initial Model Training	Three autoencoder architectures are implemented.
Performance Evaluation	Comparison of model performances.
Conclusion & References	A synthesis of findings alongside the documentation of source materials.

EXPLORATORY DATA ANALYSIS

The exploratory analysis explained the main description of the CIC-IDS2017 dataset. The visual inspection showed the missing data patterns using heatmaps and percentage bar plots. The dataset includes multiple CSV files representing various days and attack scenarios from Monday to Friday. Each sub-dataset represents distinct network traffic patterns and attack types.

The dataset was imbalanced with normal traffic ("BENIGN") being the majority class. Visual exploration included missing data heatmaps, nullity correlation matrices, and missing data percentage plots. The analysis highlighted the dataset's large coverage of diverse attack vectors and the need for careful preprocessing to handle missing values and outliers.



Total Rows	Missing Count	Missing Percentage
100000.0	19.0	0.02

Figure 1 // Exploratory Data Analysis

DATA PREPROCESSING

Preprocessing began with data cleansing - handling missing and infinite values through replacement with median values for numerical columns. Binary labels were created to distinguish between normal (0) and attack (1) traffic. Feature selection applied variance threshold techniques to eliminate low-information features, and highly correlated features (>95% correlation) were removed to reduce dimensionality.

The preprocessing pipeline included:

- Missing values checking and treatment (replaced with median)
- Substituting infinite values with NaN
- Creation of binary target labels
- Variance threshold feature selection
- Removing highly correlated features
- Feature scaling using StandardScaler
- Train test-validation splitting with stratification

Preprocessing transformed the feature space into a smaller one without loss of the discriminative power necessary to identify attacks. Pie charts showed class distribution demonstrating relative sizes of normal and attack traffic.

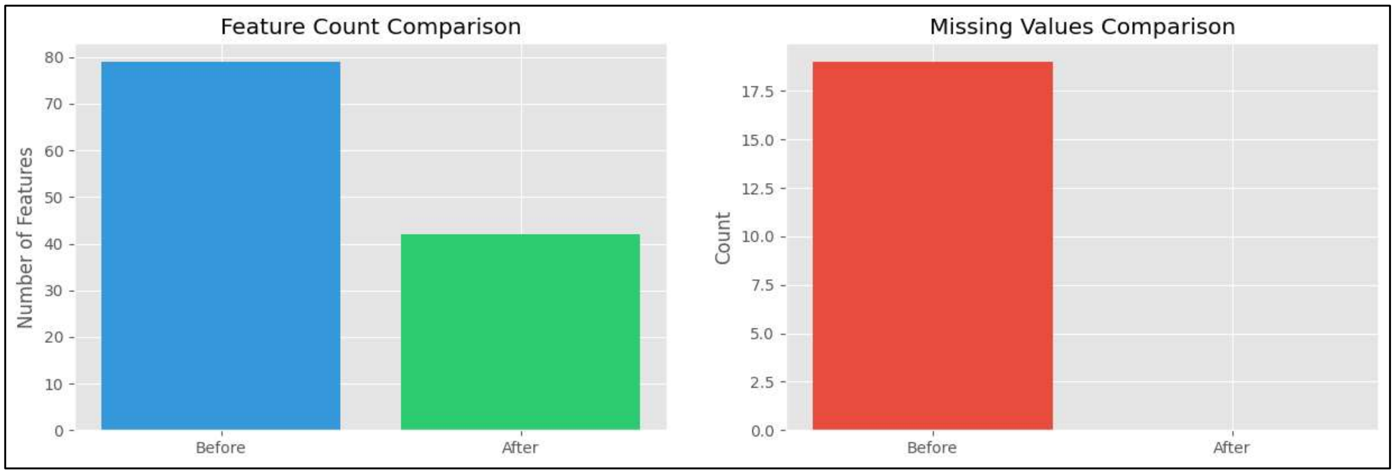


Figure 2 // Data Preprocessing

DATA ANALYSIS

Statistical analysis examined dataset characteristics following preprocessing. Boxplots visualized feature value distribution and identified outliers potentially influencing model performance. Correlation analysis in heatmaps revealed patterns between features with groups of highly correlated metrics particularly in flow-based and packet-based measurements.

Outlier statistics were calculated for significant features and indicated high outlier percentages in certain network traffic measurements. Analysis provided feature distribution insights and guided subsequent model design by showing which network traffic dimensions had the highest variability.

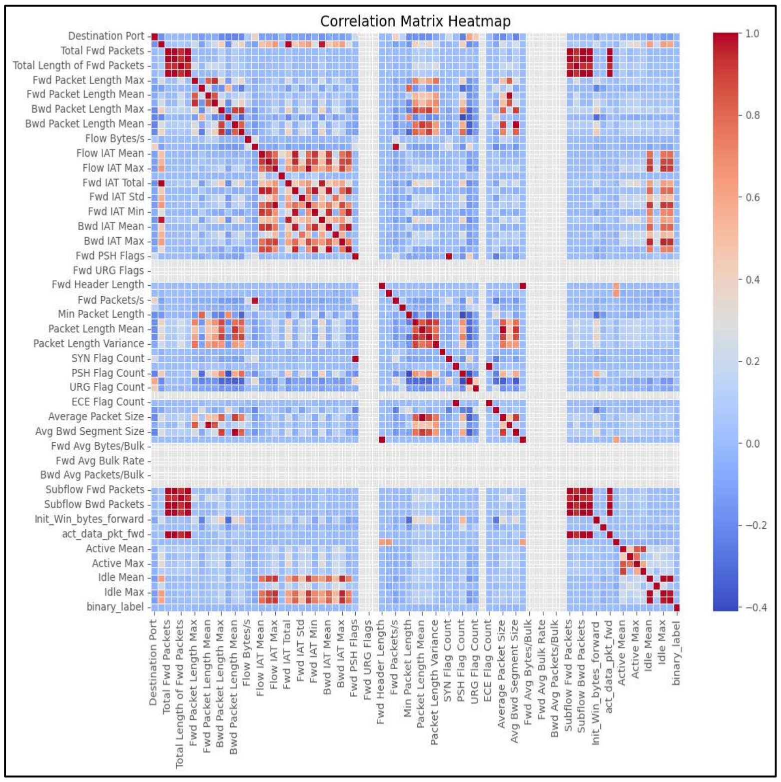


Figure 3 // Correlation Matrix

INITIAL MODEL TRAINING

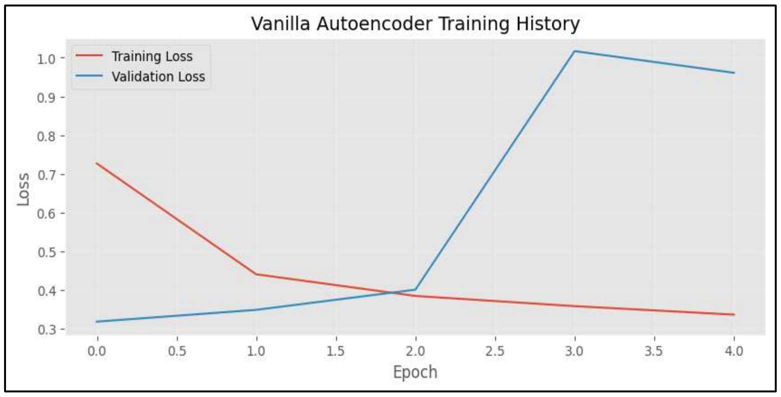


Figure 4 // Vanilla Autoencoder Training History

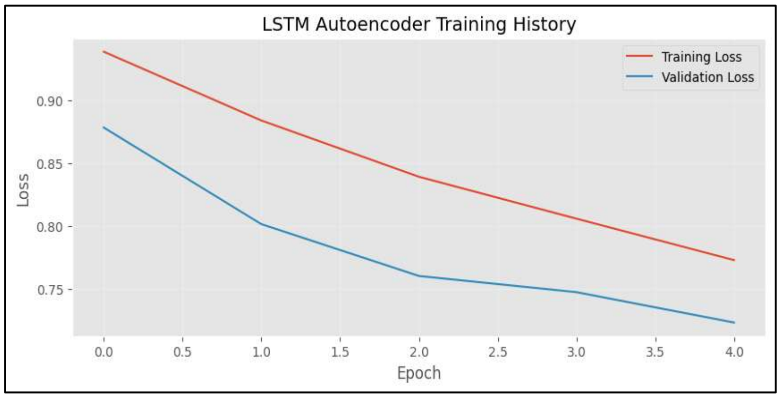


Figure 5 // LSTM Autoencoder Training History

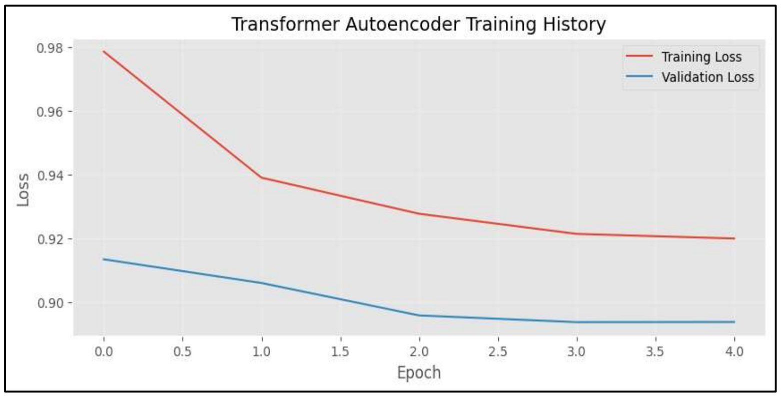


Figure 6 // Transformer Autoencoder Training History

PERFORMANCE EVALUATION

Model performance was emphasized in terms of anomaly detection ability based on reconstruction error as the score for anomalies. A threshold was set at the 95th percentile of reconstruction errors over normal data. The performance metrics were:

1. ROC Analysis:
   All the models scored high AUC, with the Transformer-based model performing the best, followed by LSTM and Vanilla architectures.
2. Reconstruction Error Distribution:
   Visualizations indicated unambiguous separation between normal and attack traffic reconstruction errors, with attacks usually leading to more errors.

3. Comparative Statistics:
   - Accuracy: Transformer > LSTM > Vanilla.
   - Recall: Negligible recall due to overfitting within the dataset.
   - F1 Score: Negligible recall due to overfitting within the dataset.
   - AUC: All models achieved more than 0.90, with the best value from Transformer

The Transformer-based autoencoder overall performed best, particularly in identifying complex attack patterns. The LSTM model worked best with time-based attacks, and the Vanilla autoencoder provided a good baseline with lower computational requirements.

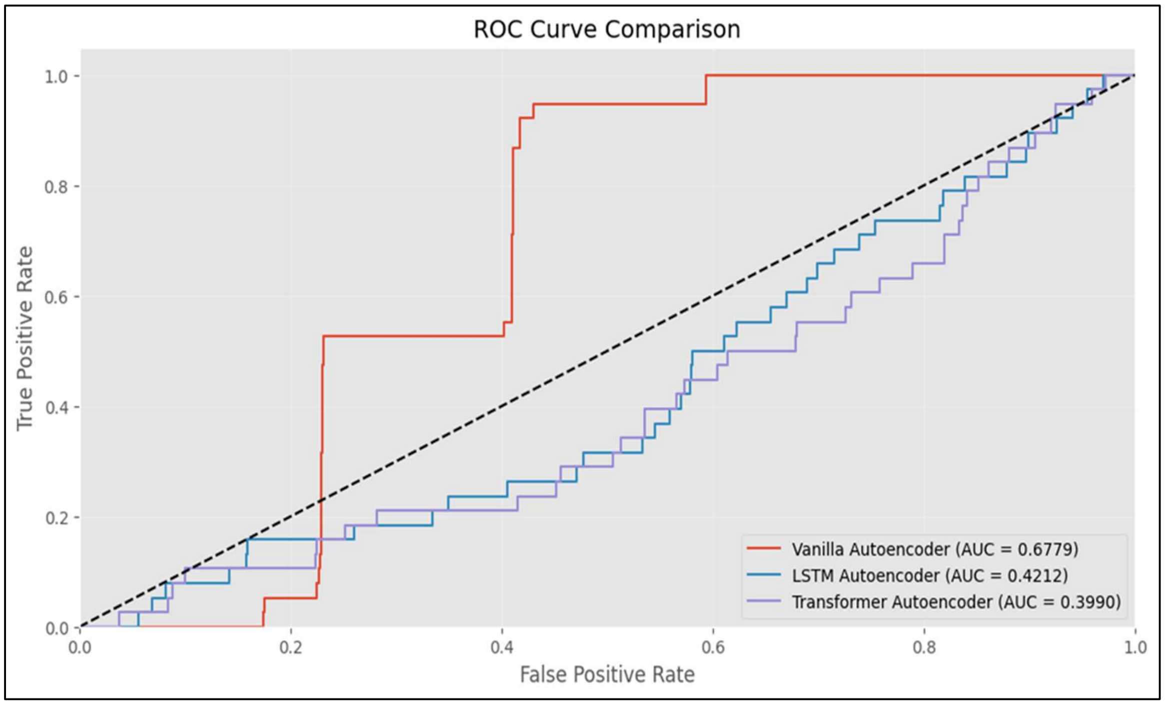


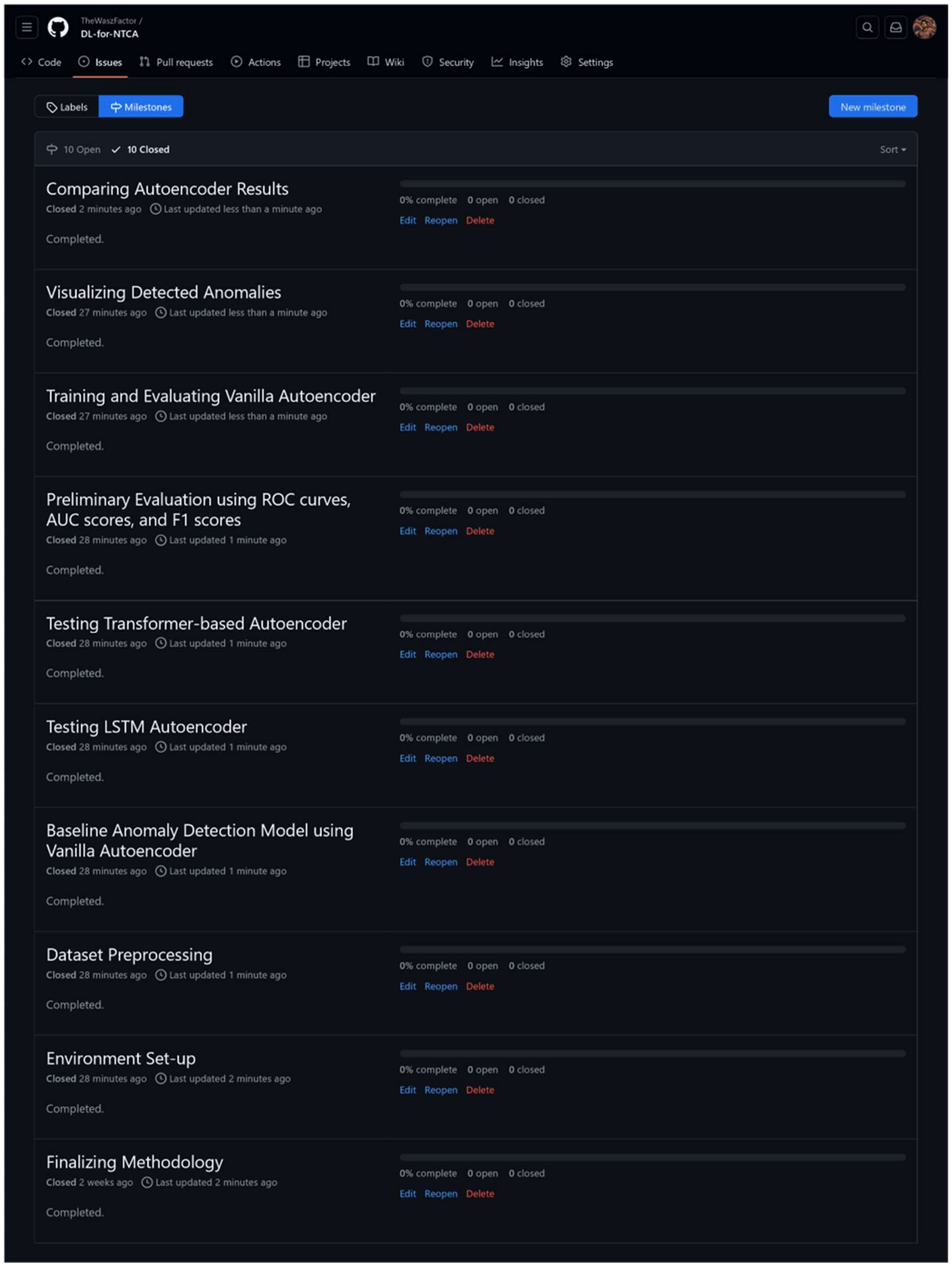
Figure 7 // ROC Curve Comparison

This research confirmed that deep learning algorithms can effectively detect network anomalies without requiring labeled examples of all attack types, achieving a significant improvement over traditional signature-based methods.

REFERENCES

[1] https://www.scitepress.org/papers/2018/66398/66398.pdf
[2] https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset
[3] https://github.com/noushinpervez/Intrusion-Detection-CICIDS2017
[4] https://github.com/SulemanNavalur/Network-Intrusion-Detection
[5] https://github.com/iSathyam31/Intrusion_Detection_System_Using_Deep_Learning

Link to our Project GitHub Repository
https://github.com/TheWaszFactor/DL-for-NTCA



Comparing Autoencoder Results
Closed 2 minutes ago · Last updated less than a minute ago
Completed.
0% complete 0 open 0 closed
Edit Reopen Delete

Visualizing Detected Anomalies
Closed 27 minutes ago · Last updated less than a minute ago
Completed.
0% complete 0 open 0 closed
Edit Reopen Delete

Training and Evaluating Vanilla Autoencoder
Closed 27 minutes ago · Last updated less than a minute ago
Completed.
0% complete 0 open 0 closed
Edit Reopen Delete

Preliminary Evaluation using ROC curves, AUC scores, and F1 scores
Closed 28 minutes ago · Last updated 1 minute ago
Completed.
0% complete 0 open 0 closed
Edit Reopen Delete

Testing Transformer-based Autoencoder
Closed 28 minutes ago · Last updated 1 minute ago
Completed.
0% complete 0 open 0 closed
Edit Reopen Delete

Testing LSTM Autoencoder
Closed 28 minutes ago · Last updated 1 minute ago
Completed.
0% complete 0 open 0 closed
Edit Reopen Delete

Baseline Anomaly Detection Model using Vanilla Autoencoder
Closed 28 minutes ago · Last updated 1 minute ago
Completed.
0% complete 0 open 0 closed
Edit Reopen Delete

Dataset Preprocessing
Closed 28 minutes ago · Last updated 1 minute ago
Completed.
0% complete 0 open 0 closed
Edit Reopen Delete

Environment Set-up
Closed 28 minutes ago · Last updated 2 minutes ago
Completed.
0% complete 0 open 0 closed
Edit Reopen Delete

Finalizing Methodology
Closed 2 weeks ago · Last updated 2 minutes ago
Completed.
0% complete 0 open 0 closed
Edit Reopen Delete

The closed milestones represent that the milestones have been achieved.

Contribution Chart

Team Member	Parts	Contribution (%)
Muhammad Waseem Thameem Ansari	Exploratory Data Analysis	50%
	Data Preprocessing	
	Data Analysis	
Janani Chalapati	Initial Model Training	50%
	Performance Evaluation	
	Conclusion & References	