Unsupervised Anomaly Detection in Network Traffic using Deep Autoencoders

Daniel Scalettar (scalettar@ucdavis.edu)

ECS 271 - Project Proposal

1 Problem

This project aims to address the problem of detecting novel, "zero-day," network intrusions in high-volume traffic. The core challenge is to build a model that can create a robust, compressed representation of normal network behavior, allowing it to detect anomalies that deviate from this norm as potential intrusions.

2 Motivation

Typical supervised intrusion detection systems (IDS) are limited by their reliance on known attack signatures. The reliance on labeled data limits their ability to detect novel, "zero-day" attacks, which are not represented in the training data. Unsupervised, anomaly-based detection systems address this limitation by learning a robust profile of normal network behavior from unlabeled data, and identify deviations from this profile as potential intrusions. However, simple unsupervised linear methods like Principal Component Analysis (PCA) may fail to capture complex patterns in high-dimensional network traffic data. This project is motivated by the need for more effective unsupervised models, that can learn intricate, non-linear representations of normal network behavior. Deep autoencoders, with their ability to learn hierarchical feature representations, have shown promise in this regard and is an ongoing area of research.

3 Dataset

This project utilizes the **CSE-CIC-IDS2018** dataset [1]. The dataset is a modern, public benchmark for intrusion detection research. It contains over 16 million instances of network traffic, each with over 80 features, covering a wide range of contemporary attack types realistically distributed amongst a background of primarily benign traffic.

4 Methodology

The proposed method is to implement a deep Stacked Autoencoder (SAE).

1. **Data Sampling:** Due to the large dataset size, a representative subset of the data will be sampled. The training set will consist of **only benign traffic** to allow the model to learn normal behavior. The test set will include a stratified sample of both benign and attack traffic that preserves the original class distribution.

2. **Data Preprocessing:** This involves data cleaning by handling missing values (e.g., by imputation), feature selection and augmentation, categorical data encoding (e.g., one-hot encoding for protocol types), and feature scaling (e.g., min-max scaling).

3. **Baseline Model Implementation:** A simple linear PCA model will be implemented first as a baseline.

4. **Core Model Implementation:** A deep Stacked Autoencoder (SAE) will be implemented. The SAE will use a symmetric "bottleneck" architecture with multiple hidden layers. It will be trained on the completely benign training data and validated using a hold-out set also consisting of only benign data to monitor for overfitting. The trained model will be evaluated on the test set containing both benign and attack traffic. MSE reconstruction error between the original test samples and reconstructions will be used to identify anomalies.

5. **Hyperparameter Tuning:** Hyperparameters such as the dimension of the bottleneck layer, addition of dropout layers, L2 regularization, learning rate, etc. will be tuned to optimize performance.

6. **Evaluation:** The performance of the core SAE model and the baseline PCA model will be evaluated using metrics such as precision, recall, F1-score, and most importantly, the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). Visualizations such as Reconstruction Error Histograms and t-SNE plots will also be used to illustrate model performance.

References

[1] Canadian Institute for Cybersecurity. (2018). *CSE-CIC-IDS2018 on AWS (CSE-CIC-IDS2018)*. University of New Brunswick. Retrieved October 6, 2025, from `https://www.unb.ca/cic/datasets/ids-2018.html`.