INSyT
Innovative Network Security Technologies

Sandia National Laboratories

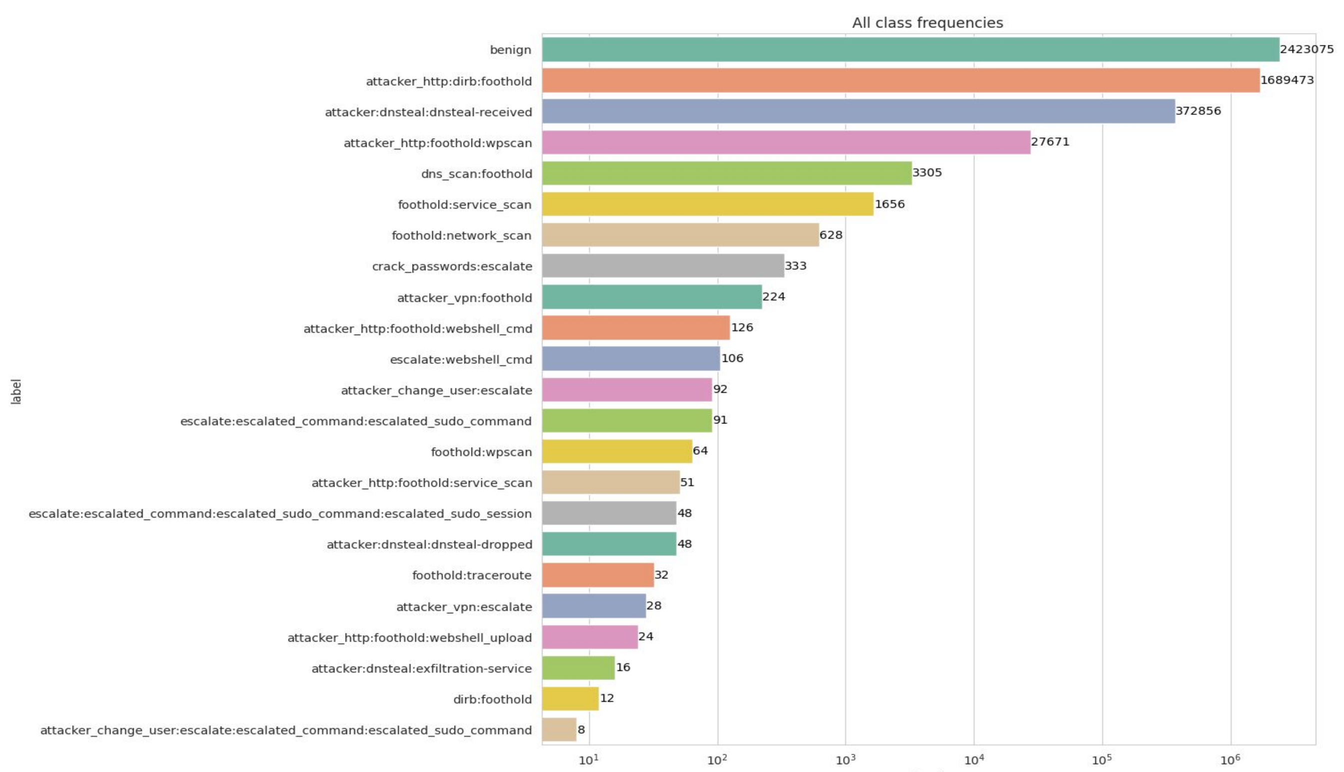
The Problem

Network intrusion refers to unauthorized access or activity on a computer network, often with malicious intent, such as stealing data or disrupting services.

The Data

We get our data from a research group in Australia that simulated attacks over the period of 7 days on 7 different servers (1).

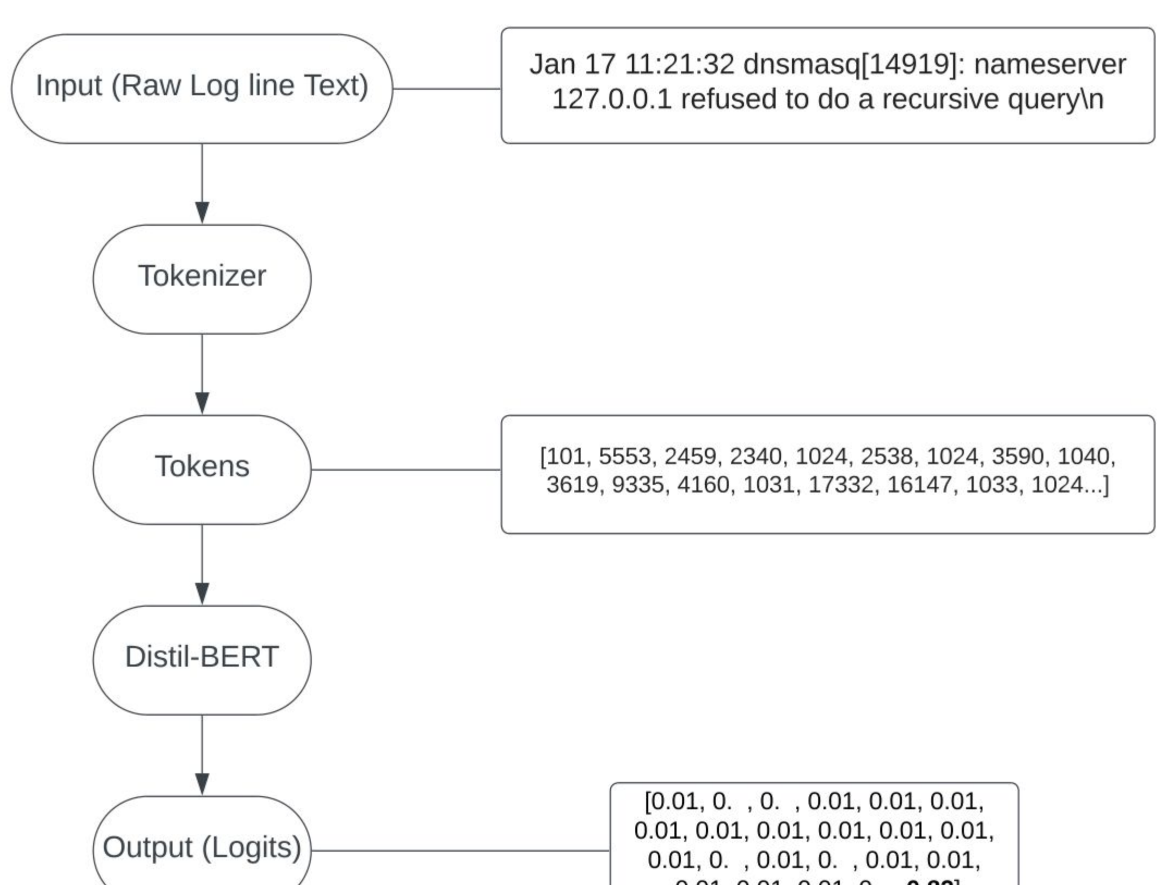
A big problem with the data that we have is the imbalance in the data. Below is a graph of the different label counts, note that the x-axis is log scaled.



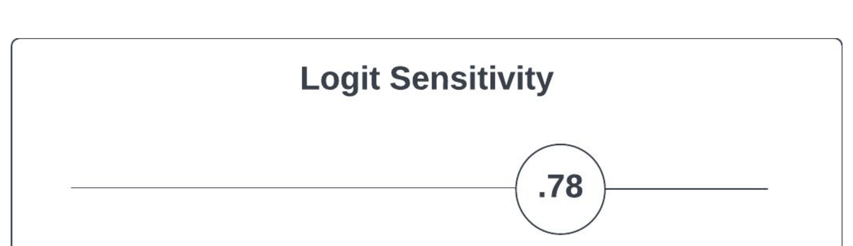
Initially we capped the max number of samples to 100,000 with plans later on to do some data augmentation in order to resolve the data imbalance, but because of the performance we were able to achieve with this data, we didn't see a need for it.

The Model

We utilized distil-BERT, (Bidirectional Encoder Representations from Transformers), distil signifying a smaller model that performs better while doing inference on a CPU.

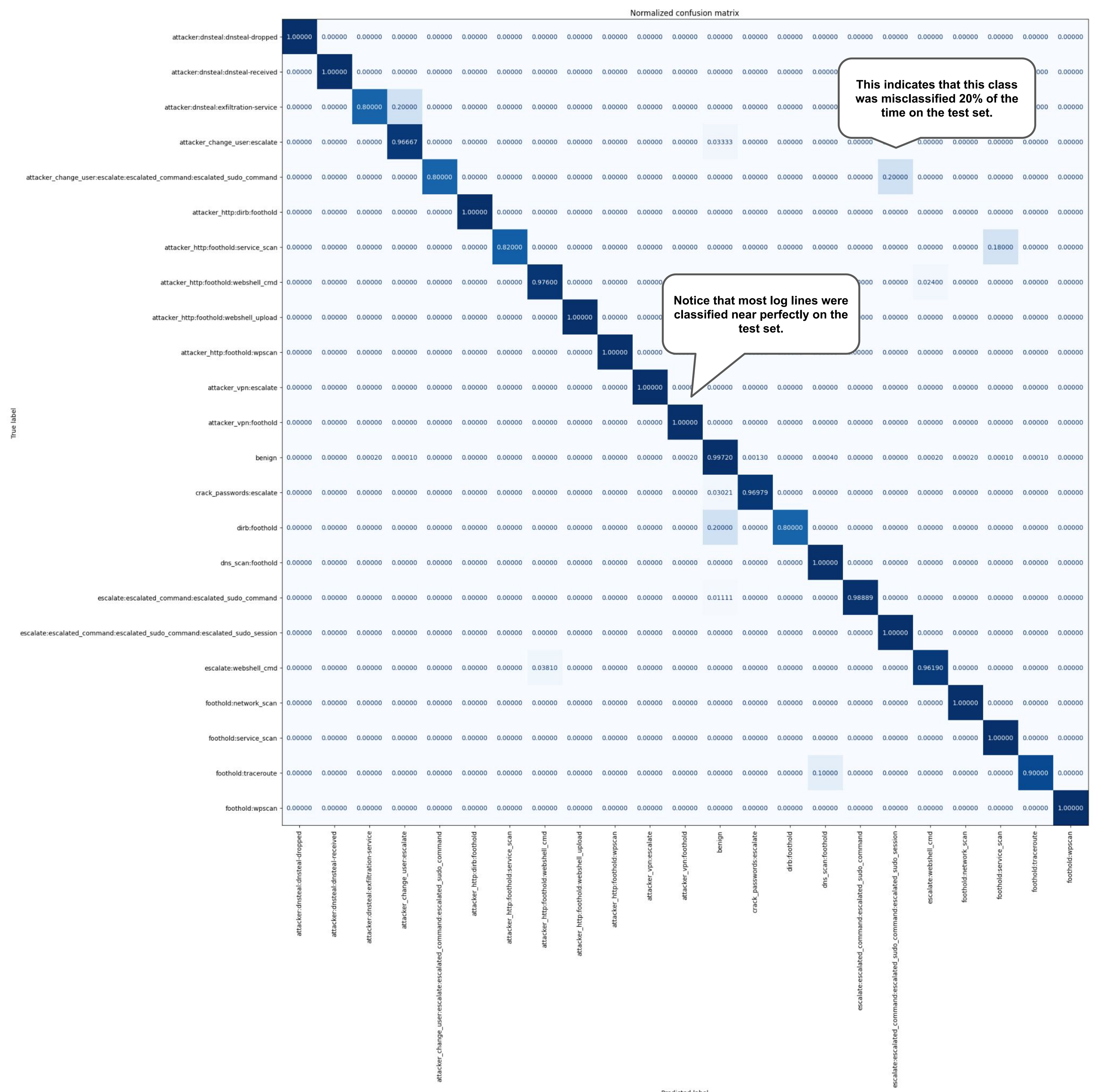


We then select the class associated with the highest value of the output logits for our classification. The output logit can be interpreted as a likelihood of class selection, which can be used to provide more informative metrics. One way we will use these logits outputs is to help manage false positives with a sensitivity slider like below.



The Results

We performed 5-fold cross validation on the data and averaged the accuracies across the five confusion matrices. For the sake of time and resources, the models trained for the validation were limited to 10,000 samples from each label. In order to ensure there was no data leakage when testing our model, we performed stratified sampling to create the train and test sets. The following confusion matrix shows the average results from the classification on the 5 test sets.



As can be seen, even with the imbalance in the data, we ended up getting very good results on the test set.

Hugging Face

The model is currently hosted on Hugging Face, at https://huggingface.co/isaacwilliam4/insyt. This allows us to download the model from Hugging Face and keep track of model versions and performance. You can also utilize Hugging Face's inference api to plug in your own log lines and see how the model labels them.

Citations

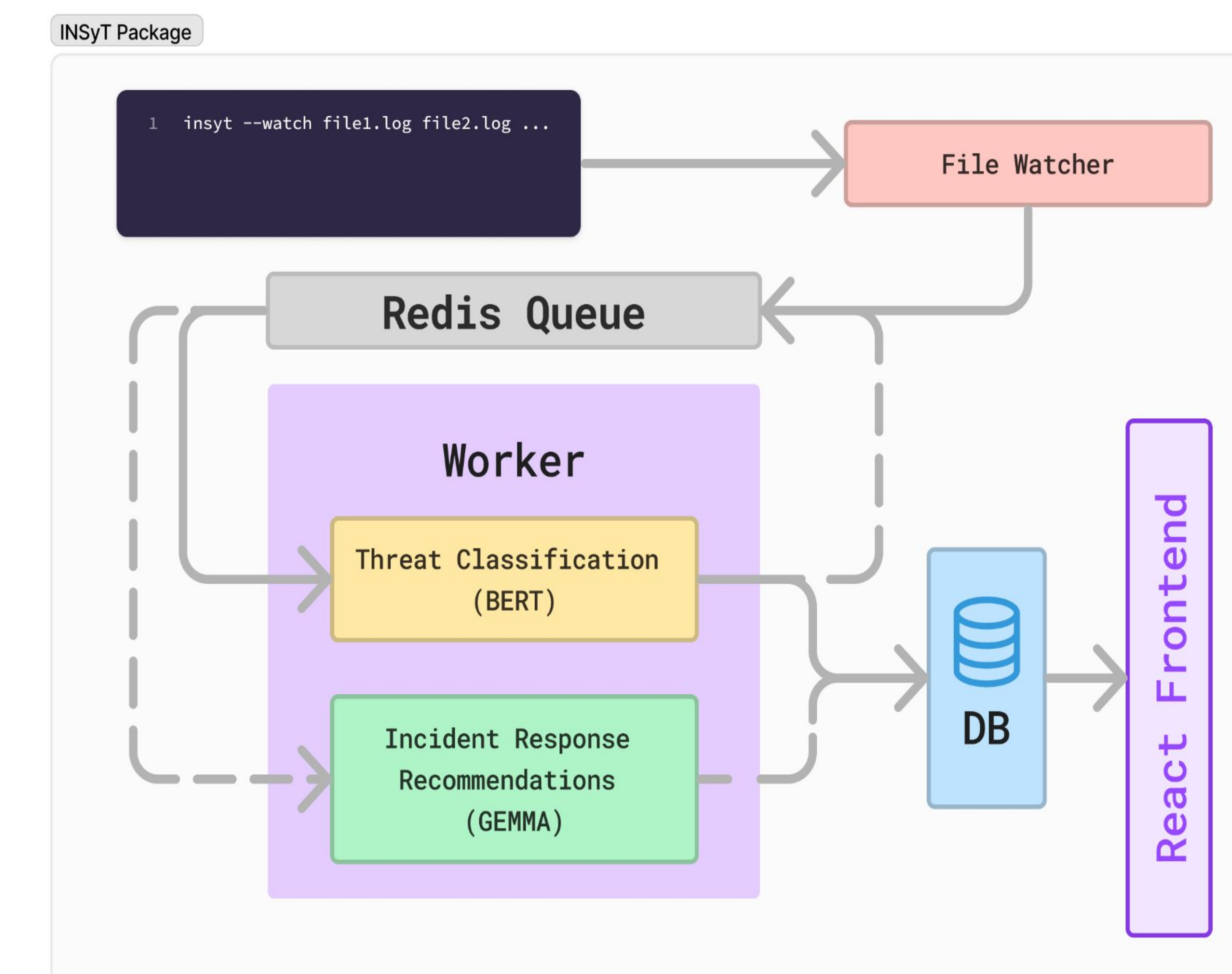
(1) Landauer, M., Skopik, F., Frank, M., Hotwagner, W., Wurzenberger, M., & Rauber, A. (2022). AIT Log Data Set V2.0 (v2_0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.5789064

The Framework/Architecture

To make the model more usable, we built a python package that can be downloaded to analyze log files and track network intrusions.

```
pip install insyt
```

The workflow for our package daemon is set up in three steps: **1)** A file watcher monitors log files and puts every new line into a Redis Queue. **2)** A worker process pulls new lines from the queue and runs it through our classification model, logging results into a SQLite database, and adding the lines identified as threats back into the queue. **3)** The worker pulls classified threats from the queue and processes them with Gemma, offering analysis and response recommendations, and loads analyses into the database.



The User Interface

To demonstrate the classification on the UI, we have used the React frontend from the local DB