## Model & Dataset Selection Research 🔬

* **Project Title:** AI-Based Cyber Security Threats Prediction
* **Document:** Model Research & Selection Report
* **Author:** Killampalli Yaswanth Vardhan
* **Date:** August 30, 2025

### 1. Objective

The primary objective of this research was to identify the most effective combination of a machine learning model and a dataset for the AI-Based Cyber Security Threats Prediction project. The goal was to find a pairing that provides the highest predictive accuracy while remaining practical for deployment in a real-time system, considering factors like model size and computational efficiency.

### 2. Methodology

To determine the optimal configuration, we conducted a comparative analysis of several potential models and datasets.

#### 2.1 Datasets Evaluated

The following network intrusion detection datasets were considered for training:

* **CSE-CIC-IDS2018:** A comprehensive and modern dataset containing a wide range of benign and malicious network traffic, including attacks like Brute-force, DoS, and web attacks.
* **CSE-CIC-IDS2017:** An earlier version with a different set of attacks and traffic patterns.
* **BCCC-CSE-CIC-IDS2017:** A smaller, more focused dataset.

#### 2.2 Models Evaluated

Three powerful, tree-based ensemble models were selected for evaluation due to their proven effectiveness in classification tasks:

* **XGBoost (Extreme Gradient Boosting):** An optimized gradient boosting library known for its high performance and speed.
* **Random Forest:** An ensemble method that builds multiple decision trees and merges them to get a more accurate and stable prediction.
* **LightGBM (LGM):** A gradient boosting framework that uses tree-based learning algorithms, known for its speed and efficiency with large datasets.

#### 2.3 Evaluation Criteria

The selection was based on the following key metrics:

1. **Predictive Accuracy:** The primary measure of the model's ability to correctly classify network traffic.
2. **Model Size:** The final size of the saved model file on disk. This is a critical factor for deployment, as smaller models consume less memory and have faster load times.
3. **Computational Efficiency:** The general speed of training and inference, crucial for a real-time prediction system.

### 3. Results & Analysis 📊

After training and evaluating each model on the selected datasets, it was determined that the **CSE-CIC-IDS2018 dataset** consistently yielded the most robust and generalizable results. The performance of the models trained on this dataset is summarized below:

| Model | Dataset | Accuracy (%) | Model Size | Notes & Observations |
| --- | --- | --- | --- | --- |
| **XGBoost** | CSE-CIC-IDS2018 | **~95%** | **~3 MB** | **Optimal Choice.** Achieves top-tier accuracy with a practical model size, making it ideal for deployment. |
| **Random Forest** | CSE-CIC-IDS2018 | **~95%** | **~4 GB** | Accuracy is excellent, but the model size is impractically large, leading to high memory usage and slow loading. |
| **LightGBM** | CSE-CIC-IDS2018 | ~89% | Small | While efficient and small, the accuracy was significantly lower than competitors and did not meet the project's requirements. |

The research showed that while both **Random Forest** and **XGBoost** achieved the target accuracy of 95%, the Random Forest model was prohibitively large at approximately **4 GB**. Such a large file would be inefficient to load into memory for the Flask API, leading to slow cold starts and high operational costs.

**XGBoost** provided the perfect balance, delivering the same high accuracy as Random Forest but in a significantly more compact and efficient package. **LightGBM**, despite its reputation for speed, failed to reach the required accuracy threshold.

### 4. Conclusion & Recommendation

Based on the empirical evidence, the combination of the **XGBoost model** trained on the **CSE-CIC-IDS2018 dataset** is the definitive choice for this project.

**Justification:**

* **High Performance:** It achieves the highest classification accuracy (~95%), on par with the best-performing models.
* **Deployment Efficiency:** It produces a model with a practical file size, ensuring fast API response times and lower memory overhead.
* **Proven Technology:** XGBoost is a state-of-the-art algorithm widely used in production environments for its reliability and performance.

This selection ensures the project is built on a foundation that is not only highly accurate but also robust, scalable, and efficient for real-world application.