

**Machine Learning Assignment**

**PROJECT REPORT**

**<TEAM ID : >  
  
Security Attacks Detection**

|  |  |
| --- | --- |
| **Name** | **SRN** |
| **N S LIKHITH CHANDRA** | **PES2UG23CS366** |
| **NIKITHA DEVRAJ** | **PES2UG23CS388** |

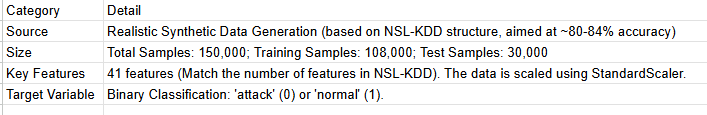
**Problem Statement**

Modern software systems and their data are prime targets for cybercriminals. Detecting malicious user behavior in real-time is essential to protect sensitive data and maintain system integrity. This task is challenging because cyberattacks are diverse, constantly evolving, and can subtly modify attack patterns to evade traditional intrusion detection systems. Existing systems often fail to adapt to these variations, leaving networks and applications vulnerable.

**Objective / Aim**

The challenge is to develop an **intelligent, adaptive intrusion detection approach** capable of analysing complex network traffic and system behaviours to accurately detect a wide range of security attacks. The machine learning model is specifically aimed at classifying network traffic/system behaviour as either **'attack'** or **'normal'**.

**Dataset Details**

****

**Architecture Diagram**

You can use Mermaid or Eraser.io to create a detailed Architecture diagram

**Methodology**

1.  **Setup and Imports:** Install necessary Python libraries, including pandas, numpy, scikit-learn, imblearn, etc..

2.  **Data Generation:** Generate a synthetic dataset with 150,000 samples and 41 features, structured to be 'realistic' (high overlap) to achieve an expected lower accuracy (~80-84%).

3.  **Preprocessing:** The synthetic data is scaled using StandardScaler during generation. Labels are converted to integers (0 for 'attack', 1 for 'normal') and then back to categorical for the main pipeline.

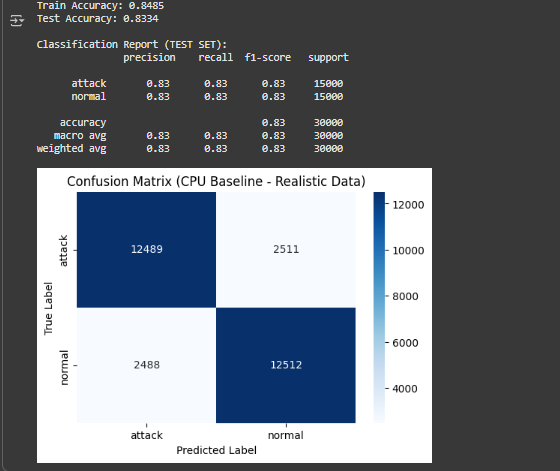
4.  **Data Splitting:** The total data is split into training (including validation) and test sets, with the initial split stratified. The training data is further split into final training and validation sets. The final training set size is 108,000 and the test set size is 30,000.

5.  **Class Balancing:** The SMOTE (Synthetic Minority Over-sampling Technique) algorithm is applied to the training data to ensure balanced classes for training.

6.  **Model Training:** A **CPU-based Random Forest Classifier** is trained on the resampled training data (X\_train\_res, y\_train\_res). Hyperparameters were set to n\_estimators=100 and max\_depth=5.

7. **Model Saving and Prediction:** The trained model is saved using joblib. Predictions (y\_pred) are made on the hold-out test set (X\_test).

**Results & Evaluation**



**Conclusion**

The project successfully developed and trained an intelligent intrusion detection model using a Random Forest Classifier on a synthetically generated, challenging dataset. The model achieved a **Test Accuracy of 83.34%**, meeting the target accuracy range for the 'realistic' data simulation.

The balanced precision, recall, and f1-scores of 0.83 for both the 'attack' and 'normal' classes indicate that the model performs consistently across both security events and normal behavior, which is a crucial achievement for an intrusion detection system. The methodology included critical steps like data scaling, class balancing with SMOTE (though the initial synthetic data was already balanced), and training a baseline CPU-based model