Network Intrusion Detection System

Meet Our Team

Rupin Ajay	**P N Sanjay**	**Rohith J**
(22011103050)	(22011103050)	(22011103019)
Rudra Panda	**P V S Sukrith**	**Raya Roshan**
(22011103049)	(22011103040)	(22011103045)

INTRODUCTION

This project uses machine learning to identify network attacks from normal network traffic. It offers a simple interface where users can input data to instantly detect potential threats, enhancing network security.

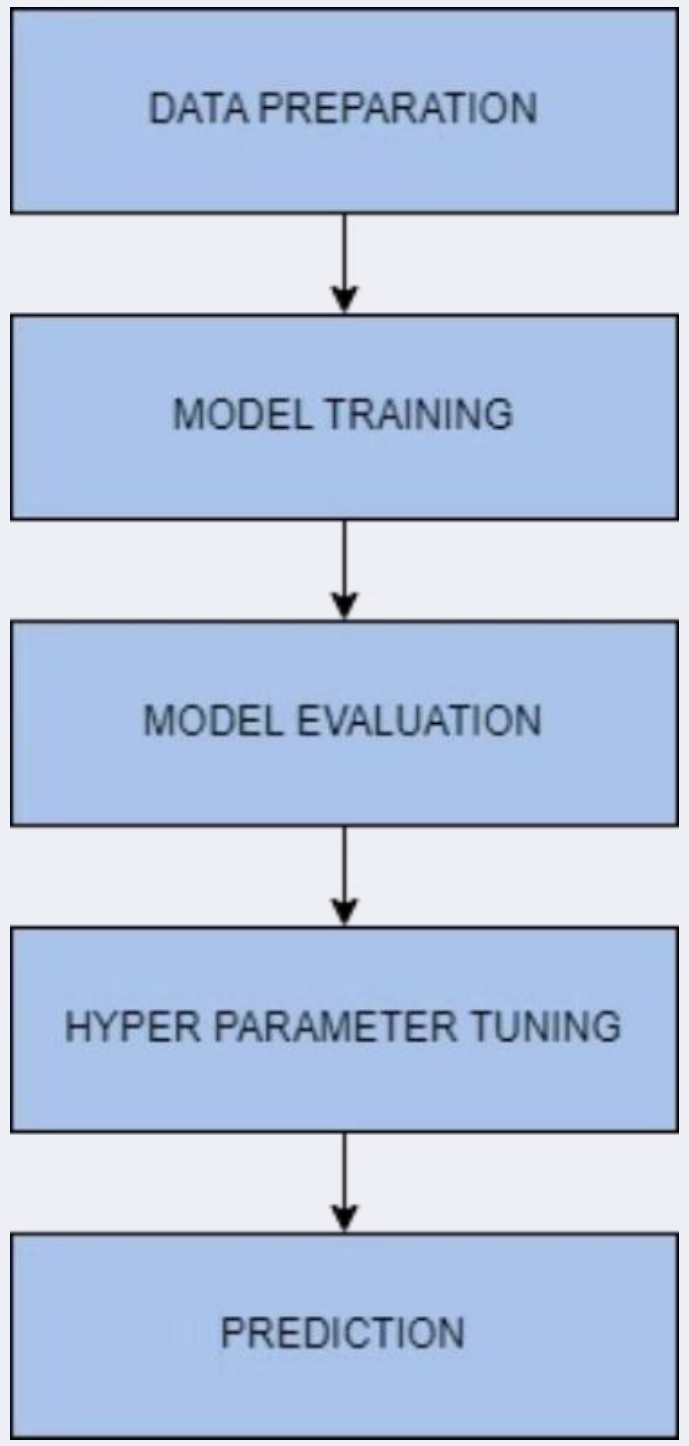
USECASES

1. **Network Security Monitoring**: Provides continuous monitoring of network traffic for potential cyber threats, enabling rapid response to suspicious activity.
2. **Intrusion Detection Systems (IDS)**: Enhances IDS capabilities by accurately classifying network traffic as normal or malicious, reducing false alarms and improving detection accuracy.
3. **Cyber Threat Analysis**: Offers insights for security analysts to analyze attack patterns, develop proactive defense strategies, and mitigate cyber threats effectively.
4. **Incident Response**: Assists incident response teams in quickly identifying the nature of security incidents and implementing containment measures to minimize damage.
5. **Compliance and Regulatory Requirements**: Helps organizations comply with regulatory standards by ensuring effective monitoring and protection of sensitive data within their networks.
6. **Network Forensics**: Aids forensic investigators in reconstructing network events, tracing attack origins, and gathering evidence for legal proceedings against cyber criminals.

TECH STACK USED

- Python
- Pandas, NumPy
- Scikit-learn (Random Forest Classifier, PCA)
- Matplotlib
- Gradio

FLOWCHART


DATA PREPARATION
MODEL TRAINING
MODEL EVALUATION
HYPER PARAMETER TUNING
PREDICTION

WORKFLOW

1. **Input Data**:
   - Users provide network traffic data, which serves as the input for the system.
2. **Data Preparation**:
   - The system preprocesses the data, which involves tasks such as handling missing values, encoding categorical variables, and applying dimensionality reduction techniques like PCA to reduce the number of features while preserving important information.
3. **Model Training**:
   - The preprocessed data is used to train a Random Forest Classifier, a machine learning algorithm capable of classifying network traffic as either normal or malicious based on patterns learned during training.
4. **Evaluation**:
   - The trained model's performance is evaluated using test data, where metrics like ROC curve and AUC are computed to assess its classification accuracy and effectiveness in distinguishing between normal and malicious traffic.

5. **Hyperparameter Tuning**:

   - To further optimize the model's performance, hyperparameter tuning is performed using GridSearchCV, a technique that systematically searches for the best combination of hyperparameters for the Random Forest Classifier.

6. **Prediction**:

   - With the trained and optimized model in place, the system is ready to analyze new incoming data to predict whether the network traffic is normal or potentially malicious.

7. **Output**:

   - Users receive immediate feedback on the classification result, indicating whether the analyzed traffic is deemed safe or if there's a suspicion of malicious activity.If there is an attack, the output shows “attack” otherwise “normal.