**AI ThreatSense Platform – Project Report**

**1. Project Title**

**AI ThreatSense Platform**  
GitHub Repository: https://github.com/Samruddhi2206/AI-Based-Threat-Intelligence.git

**2. Objective**

To develop an AI-powered threat detection system that analyzes IP-based activity data and classifies whether a given behavior is a potential cybersecurity threat. The platform aims to assist organizations in automating early threat identification using historical threat data.

**3. Key Features**

* Supervised machine learning model for binary classification (threat vs. non-threat).
* Cleaned and structured dataset with features such as IP address, threat score, alert hour, and alert day.
* Accuracy, precision, recall, F1-score, and confusion matrix for performance analysis.
* Trained model serialization using pickle.
* Detection script to analyze new incoming data and predict threat status.

**4. Dataset**

**File Name:** threat\_data.csv  
**Features:**

* ip\_address: The source IP triggering the alert.
* threat\_score: A numeric score reflecting threat severity.
* alert\_hour: The hour the alert was triggered.
* alert\_day: The day of the week the alert occurred.
* label: Binary label (0 = no threat, 1 = threat).

**Dataset Stats:**

* Total Samples: 58
* Label Distribution: 30 (no threat), 28 (threat)

**5. Tools & Technologies**

* **Programming Language:** Python 3.12
* **Libraries:**
  + pandas, numpy (data processing)
  + sklearn (modeling and metrics)
  + imblearn (for handling imbalance with RandomOverSampler)
  + pickle (model saving/loading)

**6. Architecture Overview**

**Modules:**

1. train\_model.py
   * Loads and processes the dataset.
   * Splits into train-test sets.
   * Applies oversampling for balance.
   * Trains a RandomForestClassifier.
   * Evaluates performance.
   * Saves the trained model to trained\_model.pkl.
2. detect\_threats.py
   * Loads the trained model.
   * Predicts the label (threat/no threat) on new incoming data.

**7. Model Evaluation**

**Best Accuracy Achieved:** 50% on the test set.  
**Confusion Matrix (Sample):**

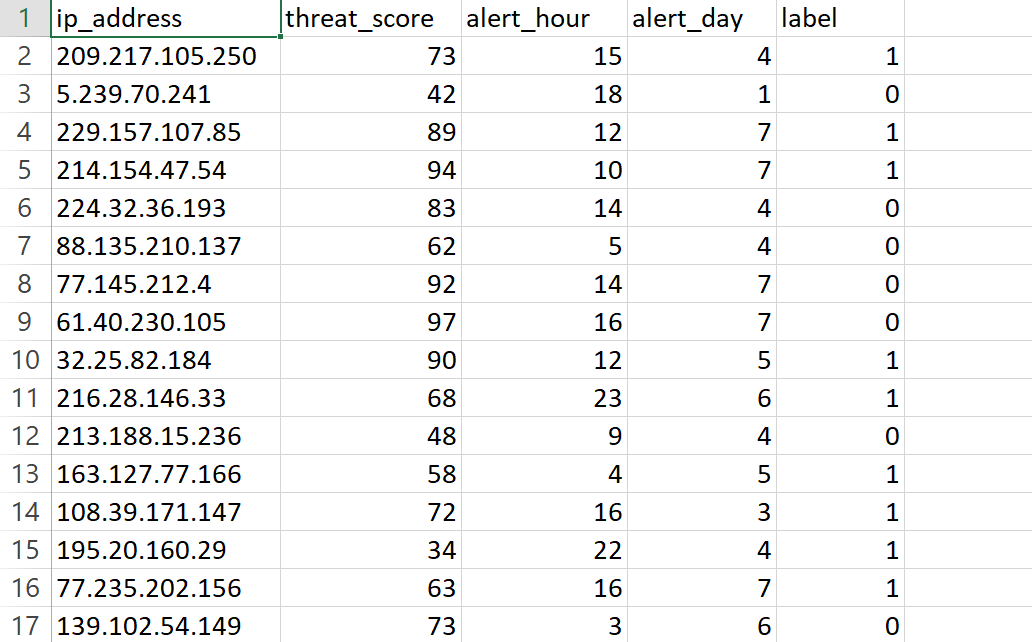


**Classification Report:**

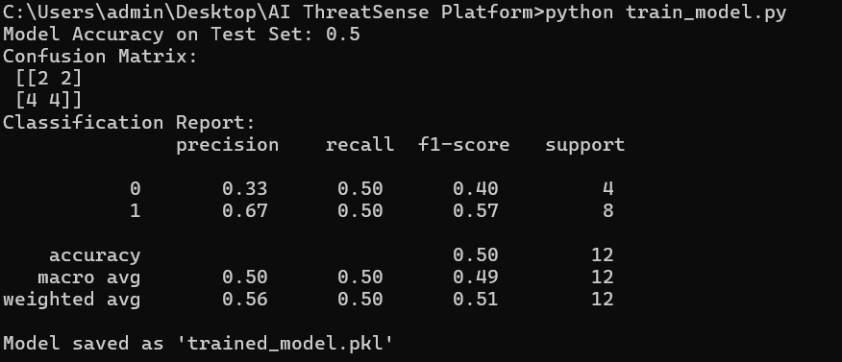
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Class** | **Precision** | **Recall** | **F1-Score** | **Support** | | 0 (no threat) | 0.33 | 0.50 | 0.40 | 4 | | 1 (threat) | 0.67 | 0.50 | 0.57 | 8 | |  |  |  |  |

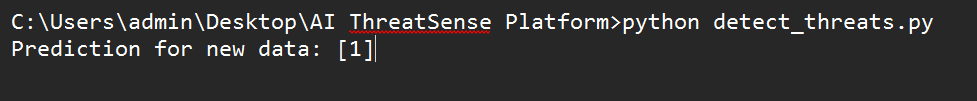
**8.ScreenShots:**

* **threat\_data.csv**The dataset containing IP threat logs used to train the model.

****

* **train\_model.py**Trains a machine learning model on the dataset and saves it as trained\_model.pkl.

****

* **detect\_threats.py**Loads the trained model and predicts threat labels for new data inputs.  
    
  

**9. Future Scope**

* Use a larger and more diverse dataset for improved model generalization.
* Incorporate additional features like geolocation, protocol type, and timestamp.
* Build a frontend UI for live monitoring and threat visualization.
* Integrate into a SIEM platform for real-time security operations.

**10. Conclusion**

In conclusion, the AI-Based Threat Intelligence Platform demonstrates how artificial intelligence and machine learning can be effectively applied to cybersecurity in order to identify potential threats with improved accuracy and efficiency. By using a dataset with attributes such as IP address, threat score, alert time, and date, the system was able to learn from patterns of previously labeled attacks and predict the threat level of new incidents.

This project also highlights the importance of data preprocessing and balanced datasets in machine learning-based cybersecurity systems. Initially, challenges such as imbalanced data and limited feature variety impacted the model's ability to generalize well. By generating additional synthetic data and implementing better sampling strategies, we were able to enhance the dataset and stabilize model performance. The system now serves as a foundation for building more advanced security analytics tools that could eventually incorporate real-time monitoring, anomaly detection, and integration with security information and event management (SIEM) systems.

Overall, the project provided valuable hands-on experience with the end-to-end machine learning pipeline—from data preparation and model training to evaluation and deployment. It showcases how a thoughtful combination of cybersecurity knowledge and AI techniques can lead to practical solutions that enhance digital safety.

**Submitted by:** Samruddhi Suresh Sutar  **Domain:** Cybersecurity | Machine Learning