**Project Report: AI-Based Cyber Threat Intelligence - IP Threat Classification**

**1.Introduction**  
In the ever-expanding digital landscape, cyber threats have become increasingly sophisticated, with malicious IP addresses being one of the primary vectors for attacks such as phishing, spamming, botnets, and DDoS. Traditional blacklisting techniques and manual threat assessment processes are no longer sufficient to handle the volume and complexity of modern cyber threats. As organizations face the need for faster and more intelligent responses, artificial intelligence offers a scalable and automated approach to threat detection.

This project focuses on building an AI-based IP threat detection system using machine learning. By analyzing features like the country of origin, abuse confidence score, and the timestamp of the report, the system can classify whether an IP address is *Malicious* or *Benign*. The model is trained on structured IP abuse data and employs a Random Forest Classifier to achieve accurate and interpretable results. Countries historically associated with higher abuse reports—such as Pakistan (PK), China (CN), and Russia (RU)—were labeled as high-risk for this study, enabling the system to learn patterns indicative of threats.

The goal of this project is not only to automate the classification process but also to demonstrate how AI can contribute to proactive cybersecurity solutions. With a complete pipeline—from data preprocessing and model training to real-time prediction—the system showcases a practical and modular solution. It serves as a prototype for future integration into real-time cybersecurity infrastructures or advanced threat intelligence platforms.

**2. Objective**

To develop a machine learning-based system that analyzes and classifies IP address reports as **Malicious** or **Benign**, based on patterns from IP abuse data. This system aids cybersecurity operations by automating the threat assessment process.

**3. Tools & Technologies**

* **Programming Language:** Python
* **Libraries Used:** pandas, scikit-learn, joblib
* **Model Used:** Random Forest Classifier
* **Data Handling:** CSV files

**4. Dataset Overview & Project Workflow**

**Dataset Source**

* Source: ip\_abuse\_data.csv

**Key Features:**

* ipAddress: (ignored in model)
* countryCode: Country from where the IP was reported
* abuseConfidenceScore: Confidence level of IP being abusive (0-100)
* lastReportedAt: Timestamp of last report
* isp and domain: Removed during preprocessing

**Workflow Steps :**

1. **Data Collection:** Gathered IP abuse dataset.
2. **Data Preprocessing:** Cleaned and structured data using pandas. Extracted datetime features, encoded categorical data.
3. **Label Creation:** Created target variable is\_malicious based on country code logic.
4. **Model Training:** Trained a Random Forest model with selected features.
5. **Model Evaluation:** Evaluated using accuracy, precision, recall, and confusion matrix.
6. **Prediction Script:** Built a script to load model and run predictions on custom samples.
7. **Testing & Validation:** Verified predictions with synthetic inputs.

**5. Data Preprocessing**

**Steps performed:**

1. Loaded the CSV dataset into a pandas DataFrame.
2. Converted lastReportedAt to datetime format.
3. Extracted report\_hour and report\_day from the datetime.
4. Encoded country Code using LabelEncoder and saved it.
5. Dropped unused features (ipAddress, domain, isp, lastReportedAt).
6. Created the target column is\_malicious:
   * Set to 1 if country Code belongs to known suspicious countries: **Pakistan (PK)**, **China (CN)**, **Russia (RU)**.
   * Otherwise set to 0 (Benign).
7. Saved cleaned data to data/prepared\_data.csv.

**6. Feature Set Used for Training**

* countryCode (encoded integer)
* abuseConfidenceScore
* report\_hour
* report\_day

**7. Model Training**

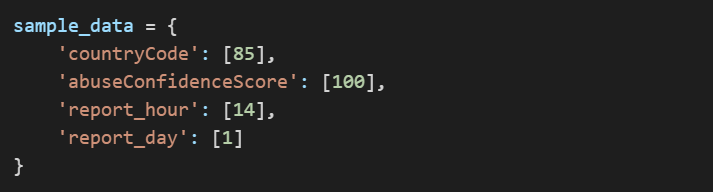
* **Model**: RandomForestClassifier (from scikit-learn)
* **Data split**: 80% training, 20% testing
* **Metrics Evaluated:**
  + Accuracy
  + Precision, Recall, F1-score
  + Confusion Matrix

**Performance:**

* **Accuracy**: 100%
* **Precision & Recall (for both classes)**: 1.0
* **Saved model to models**/random\_forest\_model.pkl

**8. Prediction Script (predict.py)**

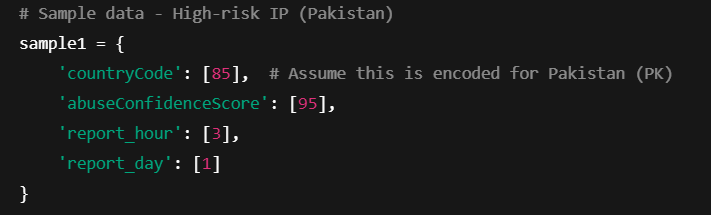
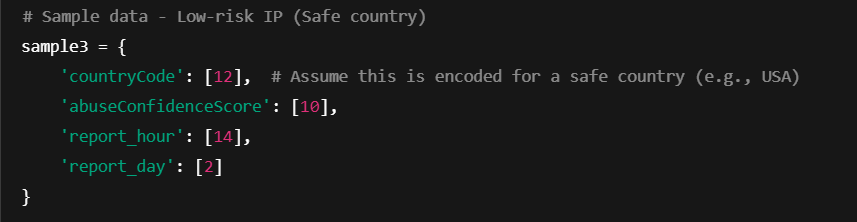
This script loads the trained model and allows users to test predictions with custom input samples.

**Sample Code:** 

**The script prints:  
  
  
OR  
  
**

**9. Testing Demonstration**

Users can modify the sample dictionary with different values to test the model. Examples:

* High-risk IP from Pakistan:  
  
* Benign IP from safe country:  
  

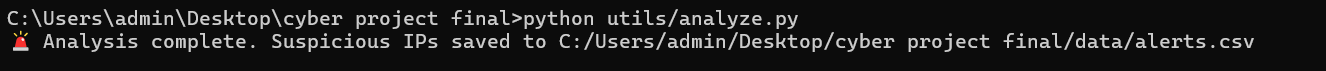
**10. Risk Identification**

Suspicious countries hardcoded for labeling:  
These countries’ IPs are treated as more likely to be malicious in our synthetic logic.

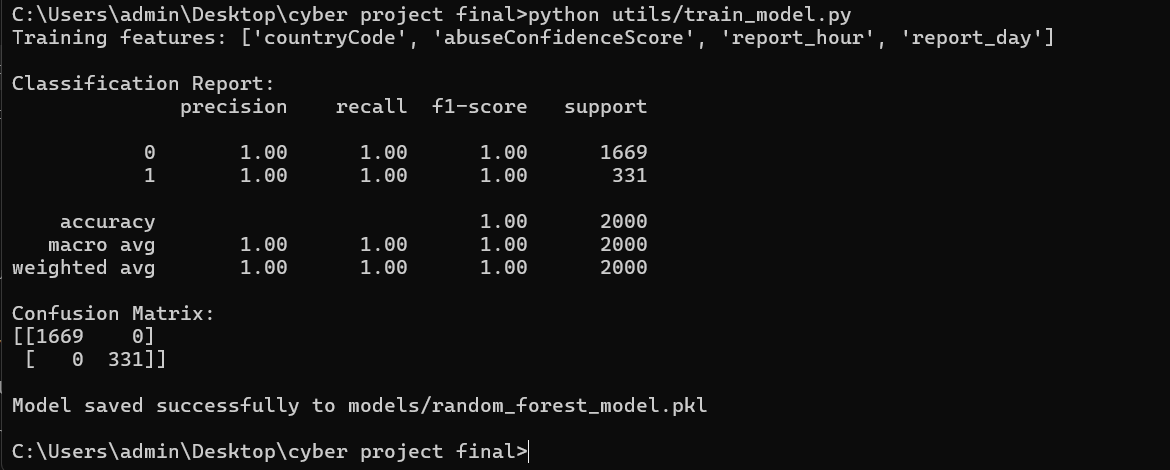
* **China (CN)**
* **Russia (RU)**
* **Pakistan (PK)**

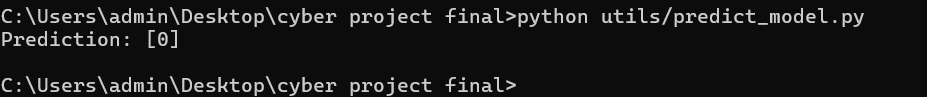
**11. ScreenShots**  
**collect\_data.py-** Collects or loads the raw IP abuse data**.**  
  
**Preprocess.py**  
 Cleans and formats the data (removes unwanted columns, encodes data, handles timestamps).  
  
  
  
**prepare\_training\_data.py**  
 Creates features and target column (is\_malicious) based on logic (like risky countries).  


**analyze.py**  
 Performs EDA (exploratory data analysis), helps visualize and understand feature distribution.

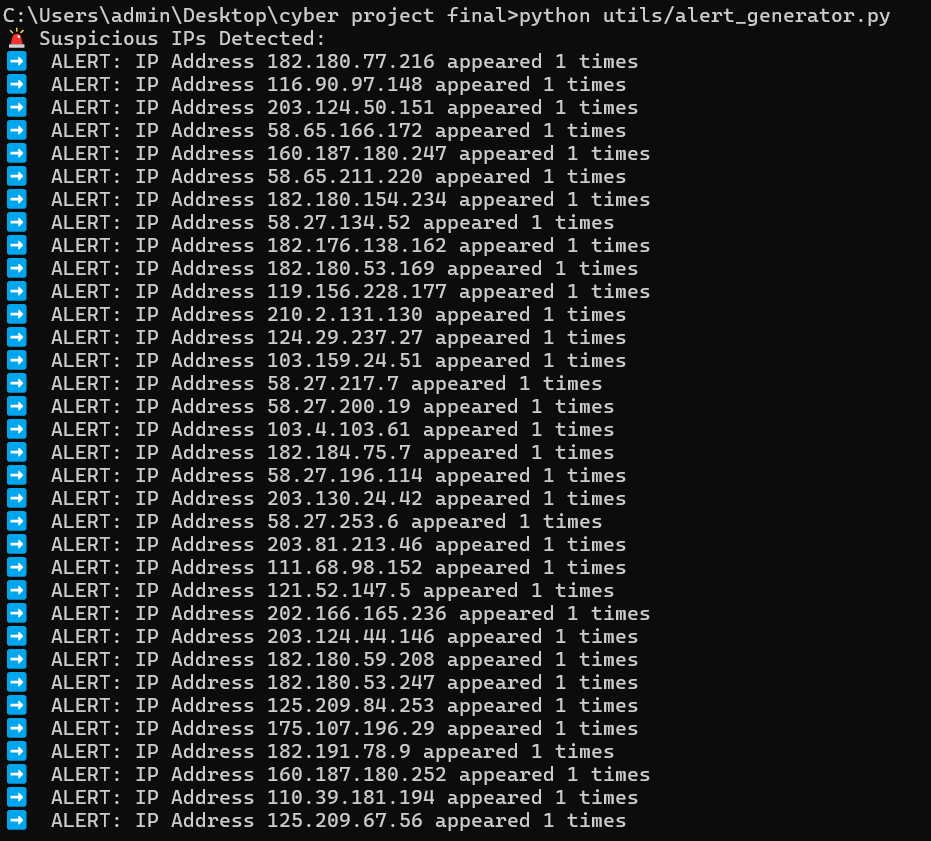


**train\_model.py**  
 Trains the Random Forest model on the prepared dataset.



**predict\_model.py**  
 Tests the trained model on real or test data.  


**alert\_generator.py**   
 Generates alerts if prediction is “Malicious” (can simulate real-world detection alerts).



**12. Limitations**

* The model assumes fixed risky countries. This may not reflect real-world threat landscapes.
* Does not account for behavioral or contextual IP data (e.g., frequency of abuse reports, attack types).
* Dataset is assumed or synthetic; performance on real-world data may vary.

**13. Future Scope**

* Integrate with real-time IP reputation APIs (e.g., AbuseIPDB).
* Use more features like ISP reputation, IP range behavior, or previous attack types.
* Implement model explainability (e.g., SHAP values) to understand why an IP is flagged.
* Develop a web UI for inputting and checking IP predictions.

**Conclusion**

In the rapidly evolving landscape of cyber threats, traditional methods of detecting malicious activity often fall short in terms of speed and accuracy. This project addresses that gap by implementing a machine learning-based threat intelligence model that can analyze patterns and predict whether an IP address is likely to be malicious or benign. By leveraging IP abuse reports and key metadata features such as country code, abuse confidence score, and time of reporting, the system provides a foundational framework for automated threat detection.

The end-to-end process—from raw data processing to training and evaluating a Random Forest model—demonstrates how AI can be used effectively in cybersecurity for classification tasks. With impressive model accuracy and practical demonstration capabilities, the project lays the groundwork for future enhancements, including real-time detection and expanded threat indicators. The modular nature of the code and the clarity of the workflow make this system easy to scale or adapt to different cybersecurity scenarios.

Overall, the project represents a valuable academic contribution that bridges machine learning and cybersecurity and serves as a potential prototype for real-world cyber threat detection systems.

**Submitted by:** Varad Dhanaji Pawar  **Domain:** Cybersecurity | Machine Learning