

Arghyajit Debnath

Member of AI/ML Intern of Team 74

1. Introduction

This project involves building a regression model to predict financial losses caused by global cybersecurity threats from 2015 to 2024. The model uses machine learning techniques to learn patterns from historical data.

2. Dataset Details

- Source: Internal dataset (cleaned by Sudip)
- File: Global_Cybersecurity_Threats_2015-2024.csv
- Time Range: 2015 to 2024
- Key Columns: Attack Type, Financial Loss (in Million \$), Country, Sector

3. Tools & Libraries

- Language: Python
- IDE: Google Colab
- Libraries: pandas, scikit-learn, matplotlib, seaborn

4. Methodology

Data Preprocessing:

- Loaded dataset using pandas.
- Label encoded categorical features using LabelEncoder.

Anomaly Detection:

- Used IsolationForest (contamination=0.05).
- Detected 150 anomalies out of 3000+ entries.

Regression Modeling:

- Used RandomForestRegressor.
- Target variable: Financial Loss (in Million \$)
- Train-test split: 80/20
- Model trained and predictions made on test data.

5. Evaluation

Evaluation Metrics:

- Mean Squared Error (MSE): 860.72
- R^2 Score: -0.064 (indicating poor fit; worse than mean prediction)

6. Interpretation & Suggestions

- Try OneHotEncoding instead of LabelEncoding.
- Remove anomalies before training regression models.
- Consider feature engineering for better representation.
- Use advanced models like XGBoost.
- Apply feature scaling where appropriate.

7. Conclusion

Despite the poor R^2 score, this project successfully implemented a complete ML pipeline for financial loss prediction using cybersecurity data. Improvements in preprocessing and model selection can enhance predictive performance.

8. Appendix

Sample Output:

Anomaly counts:

1 2850

-1 150

Random Forest Regression Results:

MSE: 860.72

R² Score: -0.064

9. Visual Output Screenshots

Medical Diagnosis AI-ML.ipynb - Colab - Train global cyber threat.ipynb - Colab

Train global cyber threat.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

from google.colab import files
uploaded = files.upload()

Choose files Global_Cyb... 15-2024.csv
Global_Cybersecurity_Threats_2015-2024.csv(text/csv) - 251279 bytes, last modified: 24/06/2025 - 100% done
Saving Global_Cybersecurity_Threats_2015-2024.csv to Global_Cybersecurity_Threats_2015-2024.csv

[2] import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import IsolationForest, RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

Load dataset
df = pd.read_csv("Global_Cybersecurity_Threats_2015-2024.csv")
data = df.copy()

Label encode categorical columns
label_encoders = {}
for col in data.select_dtypes(include='object').columns:
 le = LabelEncoder()
 data[col] = le.fit_transform(data[col])
 label_encoders[col] = le

Anomaly Detection

features_for_anomaly = data.drop(columns=['Attack Type', 'Financial Loss (in Million \$)'])
iso_forest = IsolationForest(contamination=0.05, random_state=42)
anomaly_labels = iso_forest.fit_predict(features_for_anomaly)
anomaly_counts = pd.Series(anomaly_labels).value_counts()

Variables Terminal

18:16 T4 (Python 3)

Medical Diagnosis AI-ML.ipynb - Colab - Train global cyber threat.ipynb - Colab

Train global cyber threat.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

[2] features_for_anomaly = data.drop(columns=['Attack Type', 'Financial Loss (in Million \$)'])
iso_forest = IsolationForest(contamination=0.05, random_state=42)
anomaly_labels = iso_forest.fit_predict(features_for_anomaly)
anomaly_counts = pd.Series(anomaly_labels).value_counts()

Regression: Predict Financial Loss

features = data.drop(columns=['Financial Loss (in Million \$)'])
target = data['Financial Loss (in Million \$)']
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)

Random Forest Regressor
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
predictions = rf.predict(X_test)

Evaluation
mse = mean_squared_error(y_test, predictions)
r2 = r2_score(y_test, predictions)

print("Anomaly counts:\n", anomaly_counts)
print("\nRandom Forest Regression Results:")
print(f"MSE: {mse:.2f}")
print(f"R² Score: {r2:.3f}")

Anomaly counts:
1 2850
-1 150
Name: count, dtype: int64

Random Forest Regression Results:
MSE: 860.72
R² Score: -0.064

Variables Terminal

18:16 T4 (Python 3)