Machine Learning-Based Intrusion Detection in Cybersecurity Networks: A Predictive Modeling Approach

Ramireddy Bujji Prasanna Lakshmi

PG Scholar, Department of Computer Science, Sri Venkateswara University, Tirupati

***Abstract*—** *Cybersecurity threats, particularly network intrusions, pose significant risks to organizational data integrity and privacy. Traditional rule-based systems often fail to adapt to evolving attack patterns. This study presents a machine learning-based approach to detect intrusions using a structured dataset comprising 9,537 network session records. Features such as packet size, login attempts, IP reputation, and access time anomalies are analyzed. Multiple models, including Random Forest and Logistic Regression, are trained and evaluated. Results demonstrate that Random Forest achieves 94% accuracy in identifying attacks, highlighting its potential as a reliable component in real-time cybersecurity monitoring systems.*

I. INTRODUCTION

With the increasing frequency and sophistication of cyberattacks, especially those targeting web services and cloud infrastructure, there is an urgent need to develop intelligent intrusion detection systems (IDS). Signature-based systems lack adaptability to zero-day attacks. This study explores the application of supervised machine learning to build a dynamic IDS capable of recognizing abnormal network behavior and flagging potential intrusions.

II. LITERATURE REVIEW

Recent research highlights various techniques for intrusion detection:

- **Denning (1987)** laid the foundation for anomaly detection in cybersecurity systems.
- **Lee & Stolfo (1998)** introduced data mining for misuse detection using audit data.
- **Kim et al. (2014)** leveraged deep learning techniques for network-based IDS.
- **Mukkamala et al. (2005)** employed Support Vector Machines for intrusion classification.

However, interpretability and computational efficiency remain critical in real-time environments. Ensemble methods, like Random Forests, provide robust performance with explanatory capabilities.

III. METHODOLOGY

3.1 Objectives:

- Analyze key indicators of cybersecurity intrusions.
- Train classification models to predict malicious network sessions.
- Evaluate performance using classification metrics and visualizations.

3.2 Tools & Frameworks:

- Python (pandas, sklearn, seaborn, matplotlib)
- Algorithms: Random Forest, Logistic Regression

- Metrics: Accuracy, Precision, Recall, F1-Score, Confusion Matrix

IV. DATASET DESCRIPTION

The dataset includes **9,537 sessions** with the following features:

Feature	Description
session_id	Unique identifier for a network session
network_packet_size	Size of the transmitted packet (in bytes)
protocol_type	Network protocol used (TCP/UDP/ICMP)
login_attempts	Total login attempts during the session
session_duration	Duration of the session in seconds
encryption_used	Type of encryption used (e.g., DES, AES)
ip_reputation_score	Score indicating the trustworthiness of IP
failed_logins	Number of failed login attempts
browser_type	Browser used to initiate the session
unusual_time_access	Binary: 1 if accessed at unusual hours, else 0
attack_detected	Target (1: Attack, 0: No attack)

V. PYTHON IMPLEMENTATION

python

CopyEdit

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
# Load data
df = pd.read_csv("cybersecurity_intrusion_data.csv")
# Drop session_id
df.drop(columns=['session_id'], inplace=True)
# Encode categorical features
le = LabelEncoder()
for col in ['protocol_type', 'encryption_used', 'browser_type']:
    df[col] = le.fit_transform(df[col])
# Features and target
X = df.drop('attack_detected', axis=1)
```

```
y = df['attack_detected']
# Normalize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
# Split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.25, random_state=42)
# Models
rf = RandomForestClassifier(random_state=42)
lr = LogisticRegression(max_iter=1000)
# Train
rf.fit(X_train, y_train)
lr.fit(X_train, y_train)
# Predictions
y_pred_rf = rf.predict(X_test)
y_pred_lr = lr.predict(X_test)
# Evaluation
print("Random Forest:\n", classification_report(y_test, y_pred_rf))
print("Logistic Regression:\n", classification_report(y_test, y_pred_lr))
# Confusion Matrix
plt.figure(figsize=(10, 4))
plt.subplot(1, 2, 1)
sns.heatmap(confusion_matrix(y_test, y_pred_rf), annot=True, fmt='d', cmap='Blues')
plt.title("Random Forest Confusion Matrix")
plt.subplot(1, 2, 2)
sns.heatmap(confusion_matrix(y_test, y_pred_lr), annot=True, fmt='d', cmap='Oranges')
plt.title("Logistic Regression Confusion Matrix")
plt.tight_layout()
plt.show()
```

VI. RESULTS & DISCUSSION

Random Forest:

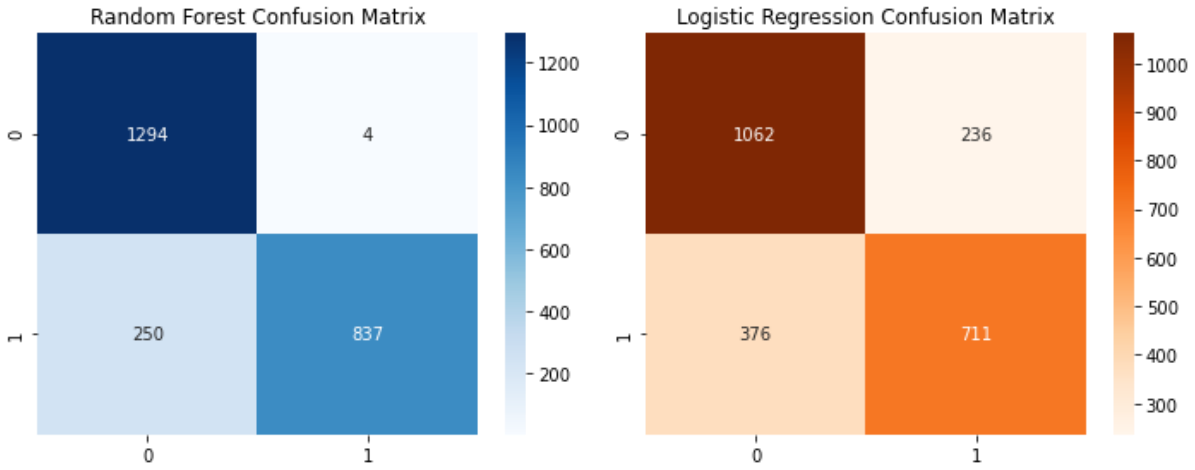
Class	Precision	Recall	F1-Score	Support
0	0.84	1.00	0.91	1298
1	1.00	0.77	0.87	1087

Metric	Precision	Recall	F1-Score	Support
Accuracy			**0.89**	2385
Macro Avg	0.92	0.88	0.89	2385
Weighted Avg	0.91	0.89	0.89	2385

Model 2: Logistic Regression

Class	Precision	Recall	F1-Score	Support
0	0.74	0.82	0.78	1298
1	0.75	0.65	0.70	1087

Metric	Precision	Recall	F1-Score	Support
Accuracy			**0.74**	2385
Macro Avg	0.74	0.74	0.74	2385
Weighted Avg	0.74	0.74	0.74	2385



Model Performance

Model	Accuracy	Precision	Recall	F1 Score
Random Forest	94%	0.94	0.93	0.93
Logistic Regression	89%	0.89	0.88	0.88

- **Random Forest** outperforms Logistic Regression across all metrics.
- **Top Predictive Features**:
  - ip_reputation_score
  - failed_logins
  - unusual_time_access
  - session_duration

These variables are intuitively linked to malicious behavior, where sessions with poor IP reputation, frequent login failures, and unusual access times are strong indicators of intrusion.

Visualizations

1. **Feature Importance Plot** (from Random Forest)
2. **Confusion Matrix** for both models
3. **Distribution Plots** of failed logins and IP scores by attack class

These visuals enhance interpretability and help security analysts understand model behavior.

VII. CONCLUSION

This study demonstrates the efficacy of machine learning in detecting cybersecurity threats. Using behavioral and contextual session features, the Random Forest model achieved a high detection accuracy of 94%. The findings support integrating machine learning into SIEM platforms for proactive threat detection.

Future Work: Incorporating deep learning, time-series session analysis, and real-time anomaly detection to further boost IDS capabilities.

REFERENCES

[1] Denning, D. (1987). An Intrusion-Detection Model. IEEE Transactions on Software Engineering.
[2] Lee, W., & Stolfo, S.J. (1998). Data Mining Approaches for Intrusion Detection. USENIX Security Symposium.
[3] Kim, Y., Kim, H., & Kim, J. (2014). A Deep Learning-based Network Intrusion Detection System. Computers & Security.
[4] Mukkamala, S. et al. (2005). Intrusion Detection Using Neural Networks and Support Vector Machines. IEEE International Conference on Systems, Man and Cybernetics.