Real-Time Anomaly Detection in Network Traffic Using Machine Learning

Name of Presenters: Kiruthika Ramadoss & Hinduja Cheela
Degree and program (of presenters): M.S. in Data Analytics and Information Systems

TEXAS STATE THE GRADUATE COLLEGE

Problem Statement

Overview:

Anomaly detection in network traffic is a critical task in cybersecurity. Anomalies often indicate threats like data breaches, malware activity, or insider attacks. This project applies machine learning to detect such anomalies in real-time traffic patterns using synthetic datasets.

Why this Matters?

As cyber threats increase in sophistication and volume, traditional rule-based detection systems fall short. Machine learning offers adaptive detection without needing explicit rules. Detecting anomalies proactively helps organizations:

- Prevent security breaches.
- Protect sensitive information.

Research Questions:

- Can we distinguish anomalous behavior using traffic flow data alone?
- How well do unsupervised vs. supervised models detect anomalies?
- Does data resampling (SMOTE) significantly improve detection?
- Can dimensionality reduction help in visualizing and understanding anomalies?

Data Sources

Dataset Used:

Synthetic Network Traffic Dataset (CSV format) simulating real-world conditions with labeled data for supervised learning.

Key Features in the Dataset:

- Bytes Sent and Bytes Received: Total volume of data in the session.
- Packets Sent and Packets Received: Reflect session communication activity.
- Duration: Time span of a network session.
- Is Anomaly: Binary label (1 = anomaly, 0 = normal).

Why Synthetic Data?

- Allows controlled testing.
- Ensures sufficient representation of rare anomalies.
- Enables experimentation with various modeling approaches

Methods

Data Collection & Cleaning:

- Loaded from CSV file.
- Checked for nulls and handled missing values.
- Standardized features using StandardScaler to normalize magnitudes.
- Engineered new features:
  - Total Bytes = Bytes Sent + Bytes Received
  - Total Packets = Packets Sent + Packets Received

Testing Statistical Assumptions:

- Visualized feature distributions using histograms and boxplots.
- Correlation matrix used to identify strong feature pairs.

Predictive Modeling:

- Unsupervised: Isolation Forest to detect anomalies without labels.
- Supervised: Random Forest trained on SMOTE-balanced dataset.
- Dimensionality Reduction: PCA used to explore feature space.

Fairness & Balance Metrics:

- SMOTE addressed class imbalance, improving detection of minority class.

Implications

Real-World Use Cases:

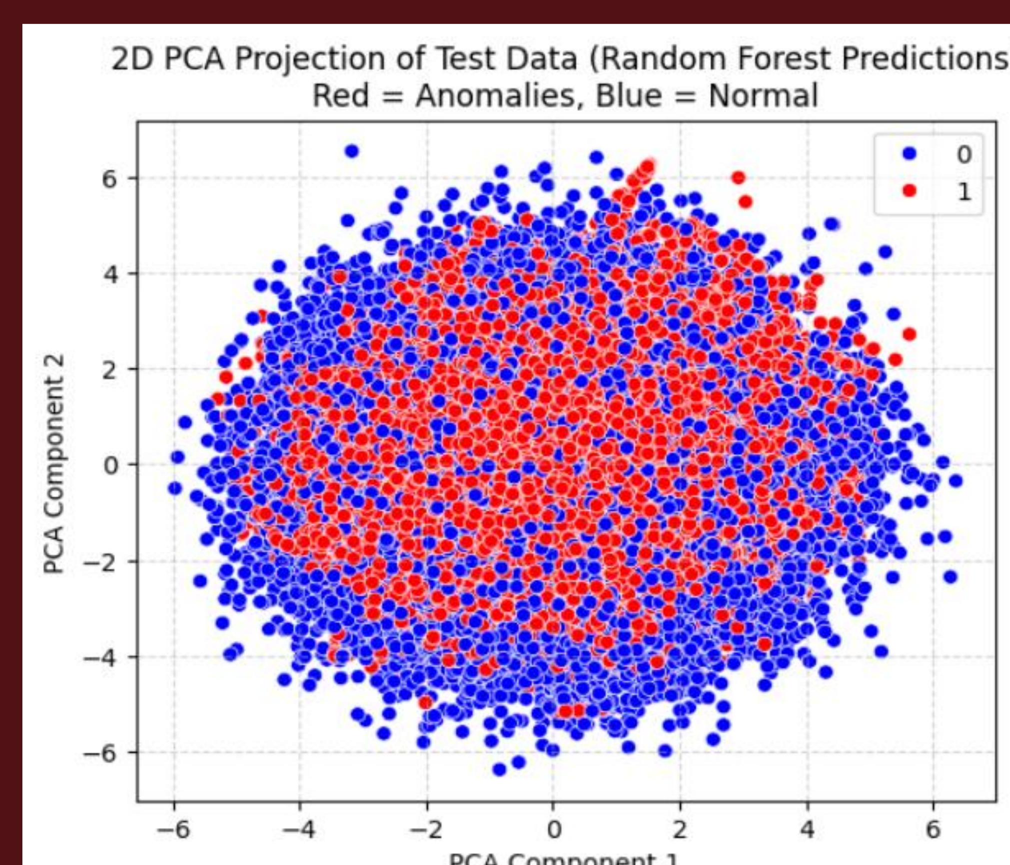
- Intrusion Detection Systems (IDS): Can integrate models to flag suspicious sessions.
- Zero-Day Attack Detection: Models can catch patterns not seen in traditional logs.
- Scalable Security: Potential to automate real-time threat detection in cloud environments.

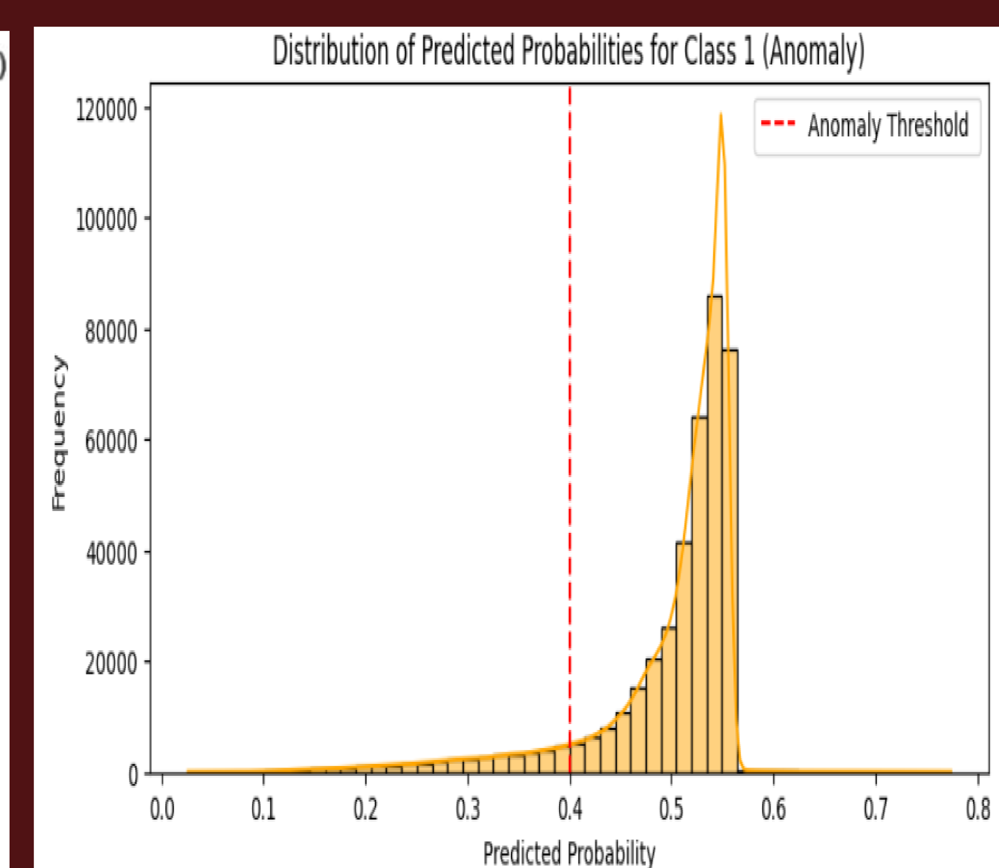
Benefits:

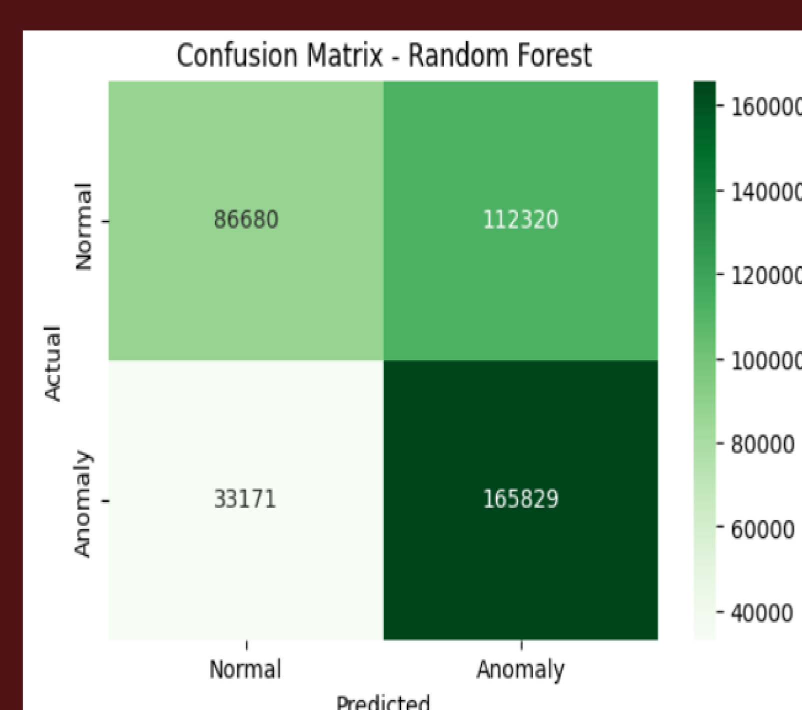
- Enhances proactive defense mechanisms.
- Reduces dependence on static rule sets.
- Adaptable to evolving attack vectors.

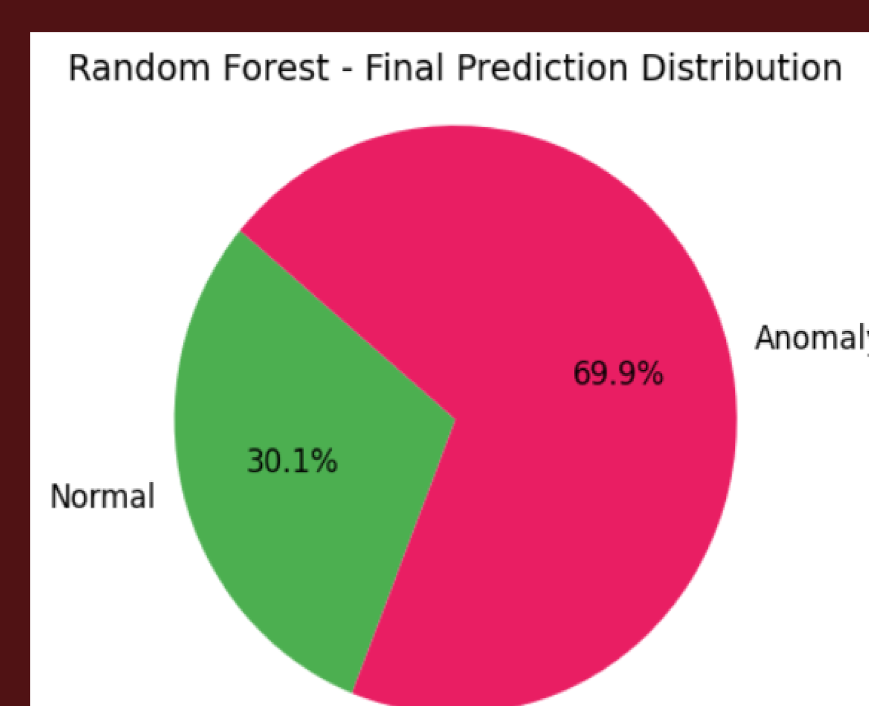
Limitations & Considerations:

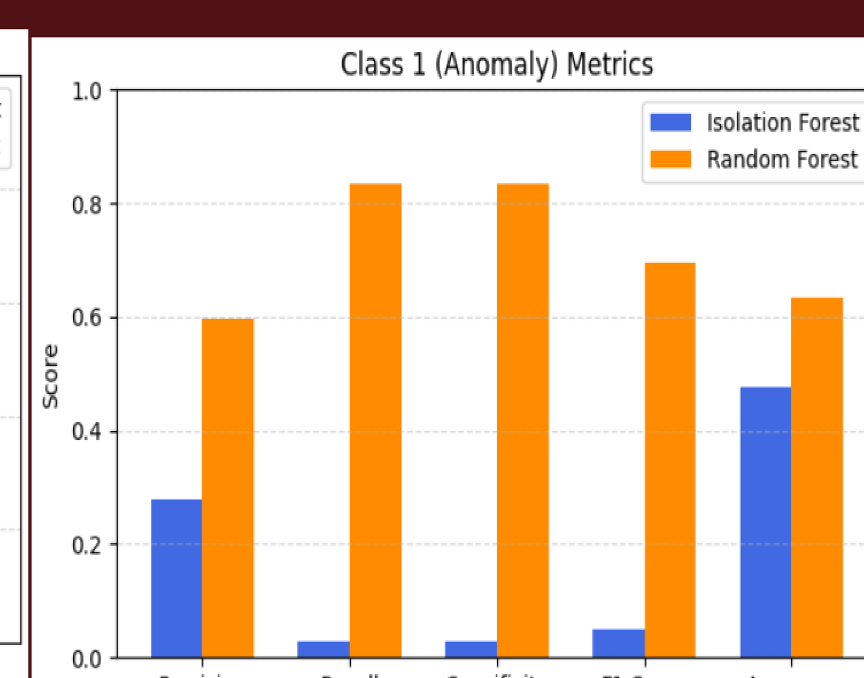
- Synthetic data may not reflect all real-world variability.
- PCA visualizations can oversimplify high-dimensional relationships.
- Unsupervised methods require careful threshold tuning.











Results and Findings

Our analysis tested both unsupervised and supervised ML models to detect anomalies in network traffic.

🏆 **Best Performing Model:** Random Forest (with SMOTE)

- ✅ Effectively captured non-linear feature interactions.
- ✅ Performed well on imbalanced data after applying SMOTE.
- ✅ Offered high interpretability through feature importance analysis.
- ✅ Achieved high accuracy and recall for rare anomalies.
- Accuracy: 63% | Precision: 60% | Recall: 83% | F1-Score: 70%

Other Model: Isolation Forest (Unsupervised)

- ✅ Advantage: Fast and does not require labeled data.
- ❌ Limitation: Prone to false positives due to lack of contextual learning.

Final Verdict: Random Forest + SMOTE is the most robust, accurate, and interpretable model for real-time anomaly detection in network traffic.

References & Data Cite

Academic & Technical References

1. Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). *Isolation Forest*. IEEE ICDM.
2. Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying Density-Based Local Outliers. SIGMOD.
3. Scikit-learn Documentation: https://scikit-learn.org
4. SMOTE Technique: Chawla, N. V., et al. (2002). SMOTE: Synthetic Minority Over-sampling Technique. JAIR.
5. Apache Kafka Documentation: https://kafka.apache.org/documentation/
6. Apache Spark Documentation: https://spark.apache.org/docs/latest/

Synthetic Network Traffic Dataset from Kaggle:

1. e.g., CICIDS2017, NSL-KDD, or similar dataset
2. https://www.kaggle.com

Conclusion & Future Work

This project demonstrates a proof-of-concept for real-time anomaly detection using machine learning. While the model shows strong performance on regular traffic, improving the detection of anomalies is key.

To Improve:

- Explore deep learning models (e.g., Autoencoders, LSTMs) or hybrid ensembles.
- Integrate real-time packet sniffing tools (e.g., Wireshark, tcpdump) for live traffic.
- Deploy an end-to-end pipeline with Kafka + Spark ML for production.

Pipeline Optimization:

- Build a full Kafka → Spark → ML Model → Alert System production pipeline and Integrate Spark MLlib for seamless streaming model deployment.