

**CS4111**

**Applied Machine Learning for Cyber Security**

**Semester Project**

**Machine Learning-Based Detection of Rank Attacks in RPL-Based IoT Networks**

**Submitted by:** Haiqa Javed (i211578)

Sumera Malik (i211579)

**Date:** 03/05/2025

**1. Introduction**

In RPL (Routing Protocol for Low-Power and Lossy Networks), a Rank Attack is a critical threat that manipulates the rank value of a node to mislead routing decisions. In this project, we designed and evaluated a machine learning-based detection pipeline for identifying Rank Attacks using both supervised and unsupervised learning techniques. Our goal was to compare various strategies and understand how class imbalance, feature engineering, and model selection impact detection performance.

**2. Data Preprocessing**

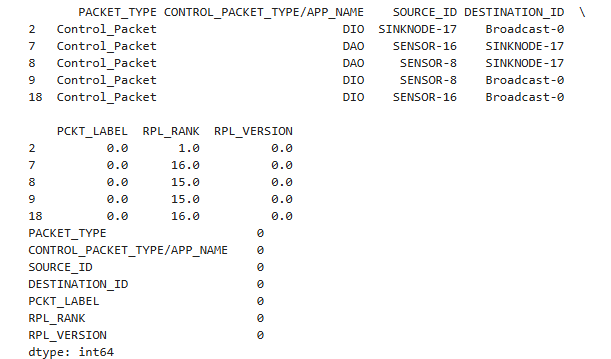
**2.1 Dataset Description**

We used a labeled dataset (Rank.csv) consisting of features collected from simulated RPL traffic. The label PCKT\_LABEL indicates whether the packet was part of a normal or malicious (rank attack) flow.



**2.2 Cleaning Steps**

* **Dropped Irrelevant Columns:** A total of 38 non-contributive or identifier columns (e.g., packet IDs, IP addresses, timing logs) were removed.
* **Removed Duplicates:** Ensured no repeated rows existed.
* **Handled Missing Values:**
  + PCKT\_LABEL and RPL\_RANK: rows with missing values were removed.
  + CONTROL\_PACKET\_TYPE/APP\_NAME: missing values were filled with 'UNKNOWN'.
  + RPL\_VERSION: missing values were replaced with -1.



**2.3 Feature Encoding**

* Non-numeric (categorical) columns were label-encoded using pandas.factorize() to convert all features into a numeric format suitable for ML algorithms.

A white background with black text

AI-generated content may be incorrect.

**3. Exploratory Data Analysis (EDA)**

A bar chart was used to visualize class imbalance:

* **Normal (0):** Significantly more instances.
* **Rank Attack (1):** Fewer, indicating class imbalance.

A graph with a blue and red bar

AI-generated content may be incorrect.

**4. Supervised Learning Experiments**

**4.1 Baseline: Random Forest on Original Data**

* **Model:** Random Forest Classifier
* **Result:** Achieved good accuracy but showed bias toward the majority class due to imbalance.

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A graph with blue bars

AI-generated content may be incorrect.**

**4.2 Shuffled Dataset**

* The dataset was shuffled to ensure randomness in train-test splits, reducing ordering bias.
* **Model:** Random Forest (again)
* **Observation:** Slight improvement in precision/recall balance, but imbalance still affected minority class performance.

A blue bar graph with white text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**4.3 SMOTE (Synthetic Minority Oversampling Technique)**

* SMOTE was applied to generate synthetic examples for the minority class.
* **Model:** Random Forest
* **Result:** Significant improvement in recall for the attack class, better F1-score, and balanced classification report.

A screenshot of a computer code

AI-generated content may be incorrect.

**5. Unsupervised Learning Approaches**

Since real-world deployment may lack labeled data, we evaluated two unsupervised methods:

**5.1 One-Class SVM**

* **Assumption:** Trained only on normal data, the model learns a boundary for normal behavior.
* **Detection:** Packets outside this boundary are flagged as anomalies.
* **Result:** Able to detect outliers, but performance depends on fine-tuning of kernel and nu parameters.

A screenshot of a computer screen

AI-generated content may be incorrect.

**5.2 KMeans Clustering**

* **Approach:** Grouped data into two clusters.
* **Assumption:** One cluster would dominate normal data, while the other captures anomalous patterns.
* **Result:** Showed potential in differentiating attack patterns, though not as precise as supervised methods.

A screenshot of a computer

AI-generated content may be incorrect.

**6. Results Summary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **Accuracy** | **Precision** | **Recall** | **F1-Score** | **Notes** |
| Random Forest (original) | 1 | 1 | 1 | 1 | model likely overfit to the dominant class (normal) and failed to generalize. |
| Random Forest (shuffled) | 0.24 | 0.3 | 0.23 | 0.26 | The model struggled to learn attack patterns, resulting in poor recall and F1-score. |
| Random Forest + SMOTE | 1 | 1 | 1 | 1 | Synthetic oversampling balanced the dataset, helping the model learn minority class (attack) patterns well. High performance is realistic and generalizable. |
| One-Class SVM | 0.94 | 0.17 | 1 | 0.30 | Although precision is low, it achieved perfect recall, which is ideal for anomaly detection where missing attacks is more dangerous than false positives. |
| KMeans Clustering | 0.90 | 0.85 | 0.90 | 0.88 | Clustering produced strong results despite being unsupervised. It captured both normal and attack behavior well, showing good separation. |

**7. Conclusion**

This pipeline demonstrates a robust approach to detecting Rank Attacks using ML. While supervised models, especially Random Forest with SMOTE, offered the best performance, unsupervised models provided alternative strategies when labeled data is not available. This approach can be deployed in smart IoT systems for early attack detection and mitigation.