**Detailed Report on Malware Detection Using Machine Learning**

**Introduction**

The rapid evolution of cyber threats, including malware and phishing attacks, presents significant challenges to cybersecurity. Traditional detection systems often struggle to provide timely and accurate detection due to the complexity and volume of these threats. This project aims to leverage machine learning (ML) classifiers to automate the detection and analysis of cyber threats, specifically focusing on malware classification.

**Dataset Overview**

The dataset used in this project consists of **100,000 entries** with **34 features**. The features include various system metrics and memory usage statistics, which are crucial for identifying malicious behavior.

Feature Breakdown

* **Numerical Features:** 33 (e.g., millisecond, state, usage\_counter, prio)
* **Categorical Features:** 1 (classification, indicating whether the instance is malware or benign)

Sample Data Structure

| **Column Name** | **Data Type** |
| --- | --- |
| millisecond | int64 |
| classification | int32 |
| state | int64 |
| usage\_counter | int64 |
| prio | int64 |
| ... | ... |

**Data Preprocessing**

Data preprocessing is a critical step in preparing the dataset for machine learning. The following steps were performed:

1. **Label Encoding**: The target variable 'classification' was encoded using LabelEncoder to convert categorical labels into numerical format.
2. **Column Removal**: The 'hash' column was dropped as it was not relevant for prediction.
3. **One-Hot Encoding**: Categorical columns were converted into dummy variables to facilitate model training.
4. **Feature Scaling**: Features were scaled using StandardScaler to normalize the data and improve model performance.

**Data Splitting**

The dataset was split into training and testing sets:

* **Training Set**: 70% of the data
* **Testing Set**: 30% of the data
* Stratified sampling was used to maintain class balance in both sets.
* A random state was set to 42 for reproducibility.

**Machine Learning Models**

Five different machine learning classifiers were trained and evaluated:

1. **Logistic Regression**
2. **Decision Tree**
3. **Random Forest**
4. **K-Nearest Neighbors (KNN)**
5. **Support Vector Machine (SVM)**

**Model Evaluation Metrics**

The models were evaluated using several metrics:

* Accuracy
* Precision
* Recall
* F1-score
* ROC curve and AUC
* Confusion matrix

**Results**

1. Logistic Regression

* **Accuracy**: 94%
* **Precision**: 0.95 (class 0), 0.93 (class 1)
* **Recall**: 0.93 (class 0), 0.95 (class 1)
* **F1-score**: 0.94 (both classes)
* **Cross-validation Accuracy**: 0.83 (+/- 0.06)

Observations:

* Good overall performance with balanced precision and recall.
* Lower cross-validation score indicates some overfitting.

2. Decision Tree

* **Accuracy**: 100%
* **Precision, Recall, F1-score**: 1.00 for both classes
* **Cross-validation Accuracy**: 0.87 (+/- 0.04)

Observations:

* Perfect performance on the test set.
* Lower cross-validation score suggests potential overfitting.

3. Random Forest

* **Accuracy**: 100%
* **Precision, Recall, F1-score**: 1.00 for both classes
* **Cross-validation Accuracy**: 0.91 (+/- 0.02)

Observations:

* Perfect performance on the test set.
* Highest cross-validation score indicates good generalization.

4. K-Nearest Neighbors (KNN)

* **Accuracy**: 100%
* **Precision, Recall, F1-score**: 1.00 for both classes
* **Cross-validation Accuracy**: 0.77 (+/- 0.07)

Observations:

* Perfect performance on the test set.
* Lowest cross-validation score suggests significant overfitting.

5. Support Vector Machine (SVM)

* **Accuracy**: 100%
* **Precision, Recall, F1-score**: 1.00 for both classes
* **Cross-validation Accuracy**: Not available

Observations:

* Perfect performance on the test set.
* Lack of cross-validation results makes it difficult to assess generalization.

**AUC - ROC Curve Analysis**

The AUC - ROC curve was plotted for each model to evaluate their performance in distinguishing between classes at various thresholds:

Key Observations:

* Logistic Regression achieved an AUC of 0.98.
* Decision Tree, Random Forest, KNN, and SVM achieved perfect AUC scores of 1.00.

These high AUC values indicate that all models are effective at distinguishing between malware and benign instances.

**Comparative Analysis**

| **Model** | **Accuracy** | **Cross-validation Accuracy** |
| --- | --- | --- |
| Logistic Regression | 94% | 83% (+/- 6%) |
| Decision Tree | 100% | 87% (+/- 4%) |
| Random Forest | 100% | 91% (+/- 2%) |
| KNN | 100% | 77% (+/- 7%) |
| SVM | 100% | Not available |

**Summary of Findings:**

Most models achieved perfect accuracy on the test set; however, Random Forest demonstrated the best balance between high accuracy and cross-validation performance, indicating better generalization capabilities compared to others.

**Conclusion**

This project successfully demonstrated the application of machine learning techniques in detecting malware using system metrics and memory usage data. While most models performed exceptionally well on the test set, Random Forest emerged as the most reliable classifier due to its superior cross-validation results.Future work will involve further validation on unseen data and exploring feature importance analysis to enhance model interpretability and performance. This report provides a comprehensive overview of your project from dataset preparation through model evaluation and results analysis, suitable for submission or presentation purposes.