APPENDIX

Comparative Evaluation of Machine Learning Algorithms for Efficient Malware Detection

K. Vijayalakshmi [a, *], Eshaa Balamurugan Subhashini [a], Safiya Al Sabahi [a], Rahmah Al Shanawi [a], Tahani Al Harthy [a],

[a] College of Engineering, National University of Science and Technology, Muscat, Sultanate of Oman.
* Corresponding Author Email: vijayalakshmi@nu.edu.om
DOI: https://doi.org/10.54392/irjmt25616
Received: 26-03-2025; Revised: 08-11-2025; Accepted: 17-11-2025; Published: 30-11-2025

Pseudocode for RF

Input: Dataset D with features F and labels L
Output: Predicted labels for test data

1. Split dataset D into training set (80%) and testing set (20%)
2. Apply correlation-based feature selection to select essential features F_selected
3. Initialize Random Forest with n_estimators = 100, max_depth = None
4. For each tree in Random Forest:
    a. Sample subset of training data with replacement
    b. Select random subset of features from F_selected
    c. Build decision tree using Gini impurity
5. Aggregate predictions from all trees using majority voting
6. Evaluate model on testing set using Accuracy, Precision, Recall, F1-score, and ROC-AUC

Pseudocode for SVM

Input: Dataset D with features F and labels L
Output: Predicted labels for test data

1. Split dataset D into training set (80%) and testing set (20%)
2. Apply correlation-based feature selection to select essential features F_selected
3. Initialize SVM classifier with RBF kernel, C = 1.0, gamma = 'scale'
4. Train SVM on training data
5. For each test sample:
    a. Compute distance from the decision boundary
    b. Assign label based on which side of the hyperplane the sample lies
6. Evaluate model on testing set using Accuracy, Precision, Recall, F1-score, and ROC-AUC

Pseudocode for Naïve Bayes

Input: Dataset D with features F and labels L
Output: Predicted labels for test data

1. Split dataset D into training set (80%) and testing set (20%)
2. Apply correlation-based feature selection to select essential features F_selected
3. Initialize Gaussian Naïve Bayes classifier
4. Fit classifier to training data:
    a. Compute mean and standard deviation for each feature per class
    b. Calculate class prior probabilities
5. For each test sample:
    a. Compute posterior probability for each class using Bayes' theorem
    b. Assign class with highest posterior probability
6. Evaluate model on testing set using Accuracy, Precision, Recall, F1-score, and ROC-AUC

Pseudocode for Logistic regression

Input: Dataset D with features F and labels L
Output: Predicted labels for test data

1. Split dataset D into training set (80%) and testing set (20%)
2. Apply correlation-based feature selection to select essential features F_selected
3. Initialize Logistic Regression classifier
4. Fit classifier to training data:
    a. Compute coefficients for each feature by minimizing cross-entropy loss
    b. Apply gradient descent to optimize coefficients
5. For each test sample:
    a. Calculate probability of belonging to malware class using logistic function
    b. Assign class label based on probability threshold (0.5)
6. Evaluate model on testing set using Accuracy, Precision, Recall, F1-score, and ROC-AUC