Malware Prediction - Project Proposal

Aditya Gupta
IIITD
Delhi, India
aditya22031@iiitd.ac.in

Avi Sharma
IIITD
Delhi, India
avi22119@iiitd.ac.in

Rishabh Jay
IIITD
Delhi, India
rishabh22401@iiitd.ac.in

Sahil Gupta
IIITD
Delhi, India
sahil22430@iiitd.ac.in

1. Motivation

We initiated this project to address the growing sophistication of malware, which poses significant threats to both individuals and organizations. Traditional security measures often fall short against evolving threats, so we sought to leverage machine learning to predict and prevent malware infections before they occur. By analyzing large datasets, we aim to enhance early detection capabilities and provide a more proactive solution to safeguarding digital systems.

2. Related Work

2.1. Malware Analysis and Detection Using Machine Learning Algorithms

This paper addresses polymorphic malware detection, evaluating Decision Trees (DT), Convolutional Neural Networks (CNN), and Support Vector Machines (SVM). DT achieved the highest accuracy (99%), followed by CNN (98.76%) and SVM (96.41%), with low false positive rates: DT (2.01%), CNN (3.97%), and SVM (4.63%). Link to paper

2.2. Evaluation of Machine Learning Algorithms for Malware Detection

The study focuses on dynamic malware detection using classifiers in a simulated environment. Random Forest (RF) and Gaussian Naive Bayes (NB) achieved 100% accuracy, precision, recall, and F1-score, demonstrating effective behavior-based detection. Link to paper

2.3. Microsoft Malware Prediction Using LightGBM Model

This research applies the LightGBM model for Windows malware detection, emphasizing feature engineering and evaluating performance using AUC-ROC. LightGBM achieved the highest AUC-ROC score of 0.684, outperforming Catboost and XGBoost. Link to paper

3. Timeline

Weeks 1-2 (August 28 - September 10):
Project initialization, literature review, dataset acquisition, data preprocessing, feature engineering, and exploratory data analysis (EDA).

Weeks 3-4 (September 11 - September 24):
Implementation of baseline models (Random Forest, SVM, Decision Trees, LightGBM), hyperparameter tuning, and model optimization.

Weeks 5-6 (September 25 - October 8):
Model evaluation and performance comparison focusing on AUC-ROC and accuracy. Final integration, results visualization, and project documentation.

Weeks 7-8 (October 9 - October 22):
Presentation preparation and final review. Progress report due on October 14. Mid-semester presentation on October 15. Continue adjustments based on feedback.

Weeks 9-10 (October 23 - November 5):
Finalize report draft and prepare for end-semester presentation. Making any potential changes for making more progress. Fine tuning and comparing the models .

Weeks 11-12 (November 6 - November 19):
Report submission on November 27. End-semester presentation on November 28.

4. Individual Tasks

(Subject to modification as the project evolves)

- Aditya Gupta (2022031) - Handle the implementation and hyperparameter tuning of the SVM and LightGBM models. Work on comparing these models with others and analyzing their performance metrics. Along with this , he will help in data preprocessing , etc.
- Avi Sharma (2022119) - Oversee the overall model optimization strategy, coordinating the integration of different techniques to improve performance across all models. Additionally, Avi will be responsible for the final integration of results, creating detailed visualizations, and leading the presentation of the project's outcomes and findings.
- Rishabh Jay (2022401) - Focus on data preprocessing and feature engineering, ensuring the dataset is well-prepared for model training. Collaborate on implementing and fine-tuning the Random Forest and Decision Tree models.
- Sahil Gupta (2022430) - Lead the evaluation of the models using performance metrics like AUC-ROC, accuracy, precision, and recall. Assist in refining the models using advanced techniques to enhance their performance.

5. Final Outcomes

In this project, we aim to optimize the hyperparameters of various machine learning models, including Random Forest, SVM, Decision Trees, and LightGBM, to enhance their performance in addressing the problem statement. We will also explore different techniques to improve the performance metrics of these models. Additionally, we will conduct a thorough comparison, providing detailed explanations for the observed results and highlighting the factors that contribute to their effectiveness in solving the problem.