AI-DRIVEN VULNERABILITY AND MALWARE DETECTION IN WINDOWS BINARIES

FSU UNDERGRADUATE RESEARCH OPPORTUNITY PROGRAM
CENTER FOR UNDERGRADUATE RESEARCH & ACADEMIC ENGAGEMENT

FSU | FLORIDA STATE UNIVERSITY

Aakanksha Pathak, Kiyan Atighechi, Tyler Fink, Christian Victoria, Dr. Sharanya Jayaraman

INTRODUCTION

Background

- Windows PE files are common malware targets due to widespread system usage.
- Modern malware uses obfuscation, packing, and polymorphism to evade detection.
- Traditional signature-based tools detect known threats but fail against zero-day or modified malware.
- AI/ML models can learn structural patterns directly from binary data.
- Features such as byte distributions, entropy, API imports, and PE metadata show predictive value.

Current Limitations

- Many detection systems require disassembly or sandbox execution.
- These approaches are slow, resource-intensive, and potentially risky.
- Obfuscated malware frequently bypasses conventional defenses.
- Limited research emphasizes scalable, pure black-box analysis.

RESEARCH OBJECTIVE

- Evaluate whether AI/ML models can identify malicious Windows executables without disassembly or execution, outperforming signature-based systems.

Hypothesis:

- Pattern-learning AI models can detect concealed and previously unseen malware variants.

Approach:

- Treat executables as black boxes
- Extract structural binary features
- Train pattern-based models (e.g., CNNs, feature-based classifiers)

Technical Metrics:

- Improve speed, scalability, and reliability of automated malware detection.

METHODS

- Used the EMBER 2018 dataset, which contains labeled Windows program files (malware and safe files).
- Each file is described by 2,381 numerical features based on file structure, headers, imports, and statistical patterns.
- Treated each file as a black box — we did not open, run, or reverse engineer the programs.
- Used 100,000 samples to train the deep learning model.
- Scaled the feature values so they were consistent and split the data into training and testing sets.
- Reshaped the feature data so it could be processed step-by-step by the neural network.
- Tested two models:
  - LightGBM, used as a baseline machine learning model.
  - A 1D Convolutional Neural Network (CNN) built in TensorFlow.
- The CNN included layers that learn patterns, reduce complexity, and prevent overfitting before making a final decision.
- Trained the models for 100 epochs using small batches of data.
- Measured each model's performance using accuracy and validation results for detected malicious files using only static information.

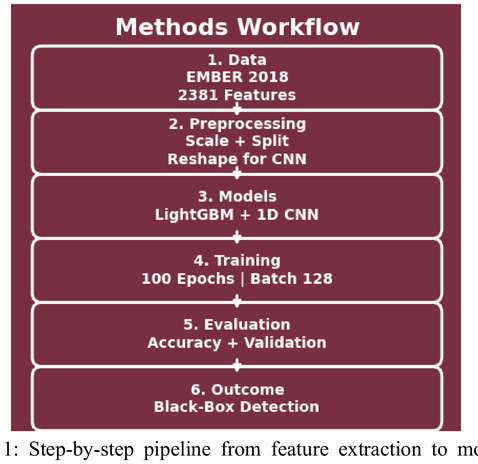

Figure 1: Step-by-step pipeline from feature extraction to model training and final malware detection results.

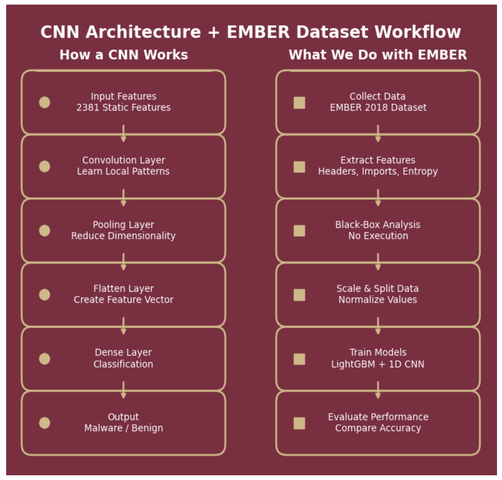

Figure 2. How the CNN processes static features and how we apply it to the EMBER dataset for malware classification.

PRELIMNARY RESULTS

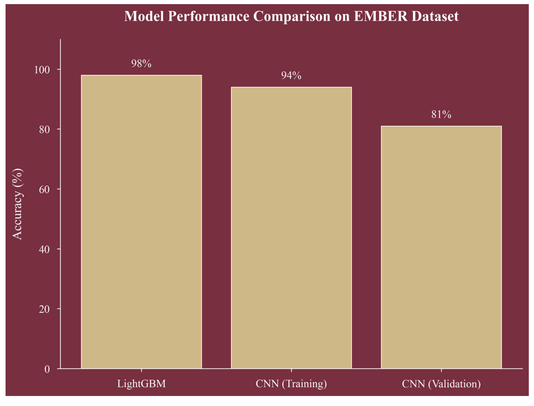

Figure 3. Model Accuracy Comparison. LightGBM achieved 98% validation accuracy, while the CNN reached 94% training and 81% validation accuracy, showing mild overfitting.

MODEL PERFORMANCE

LightGBM (Baseline Model):

- Achieved approximately 98% classification accuracy on the EMBER dataset.
- Demonstrated strong performance using structured feature vectors.

1D Convolutional Neural Network (CNN):

- Reached approximately 93–94% training accuracy.
- Achieved approximately 81% validation accuracy.
- Validation performance suggests some overfitting relative to the baseline model.

DISCUSSION

- The preliminary results show that malware can be detected accurately using only static feature data.
- LightGBM achieved the highest performance (~98% accuracy), showing that structured feature-based machine learning is highly effective.
- The CNN successfully learned patterns from the data but showed lower validation accuracy (~81%), suggesting some overfitting.
- High detection accuracy was achieved without disassembling or executing files, making this approach faster and safer than dynamic analysis.

Limitations

- CNN showed signs of overfitting.
- Experiments limited to the EMBER 2018 dataset.
- Static analysis may miss dynamic behavioral indicators.

Future Work

- Improve CNN generalization through regularization and tuning.
- Explore hybrid models (e.g., probabilistic + neural approaches).
- Evaluate performance on newer or more diverse malware datasets.
- Investigate robustness against heavily obfuscated and zero-day malware.

REFERENCES

ACKNOWLEDGEMENT

We sincerely thank Dr. Sharanya Jayaraman for her mentorship, guidance, and support throughout this research. We am also grateful to the research team for their collaboration and contributions.

We formally acknowledge the Center for Undergraduate Research and Academic Engagement (CRE) and our UROP mentors for their structured support and encouragement.