CS6264 Project 7: Exploring Modeling & Attacking PE Models via MLSploit

Task 1: Training DL Models

Approach
To classify malware families, I trained three DL models LSTM, CNN, and RNN on API call sequences using the MLSploit platform. The training dataset, 'pe.train.txt', consisted of labeled PE files categorized as benign or malicious. Evaluation was conducted on 'pe.eval.txt' to validate model performance.

Findings

1. Model Selection: LSTM achieved the highest accuracy due to its ability to capture long-term dependencies in sequences. RNN was effective for identifying localized patterns, while CNN performed moderately but struggled with vanishing gradient issues.

2. Hyperparameters:
   - Sequence Window Size: Set to 51 to capture sufficient context for accurate predictions.

3. Accuracy: The updated log files revealed accuracies of 95% (LSTM), 90% (RNN), and 88% (CNN).

Observations
Longer sequences improved classification accuracy but increased training time. LSTM's ability to capture sequential dependencies made it superior for API call sequence data.

Task 2: Attacking DL Models

Approach
For this task, I enabled the sequence parameter, set the window size to 51, and activated sequence_lstm, sequence_rnn, and sequence_cnn modes in the Mimicry Attack configuration. A Mimicry Attack was implemented using the 'pe.benign.txt' and 'pe.target.txt' datasets. The PE Transformer module rewrote malicious binaries to mimic benign API call sequences.

Findings

1. Attack Success:
   - LSTM: Classification accuracy dropped to 48.67%
   - CNN: Classification accuracy dropped to 0%
   - RNN: Classification accuracy dropped to 2.33%

2. Performance Impact: Models struggled with samples exhibiting long sequences of benign-like calls, confirming the attack's effectiveness.

Observations

Enabling sequence-based processing exposed vulnerabilities in DL models to mimicry attacks. While models like LSTM resisted attacks to some degree, CNN and RNN models showed significant degradation, highlighting the importance of sequence robustness in model design.

Task 4: Training Classical ML Models

Approach
I trained ML models on additional API-related features (existence, frequency, and arguments) using MLSploit.

- Existence: Enabled existence and existence_knn with existence_knn_k set to 7.
- Frequency: Enabled frequency and frequency_rf with frequency_rf_trees set to 10.
- Arguments: Enabled arguments and arguments_nb.
- During the ensemble evaluation, all three features (existence, frequency, arguments) were enabled.

Findings

1. Feature Comparisons:

    - Existence features yielded validation accuracy of 95.2%.
    - Frequency features achieved validation accuracy of 100%.
    - Arguments features resulted in validation accuracy of 86.3%.

2. Ensemble Performance:

    - Ensemble accuracy for first evaluation was 96.4%.
    - Ensemble accuracy for second evaluation was 97.6%.
    - Ensemble accuracy for third evaluation was 90.8%.

Observations

Combining multiple API-related features in the ensemble improved overall classification performance. Frequency features, supported by Random Forest, demonstrated exceptional predictive ability, while existence and arguments added complementary strengths.

Task 5: Attack Transfer to ML Models

Approach
For this task, I used the MalwareLab module to submit and execute 10 attack malware samples. The PE Ensemble-Evaluate module was configured with the following parameters:

- Sequence: Enabled.
- Window Size: Set to 51.

Findings

1. MalwareLab Execution:
   - 10 samples successfully submitted and executed.
   - Samples included various attack binaries with unique SHA256 hashes:
     - Example: attack-0.exe with hash 6d8793cd0c14e28835463b6b8ce3ee9707206ed2e7c47379afe8860fcdab45dd.

2. Model Evaluation:
   - LSTM Accuracy: 58.86%
   - RNN Accuracy: 2.57%
   - CNN Accuracy: 0%

Observations

The LSTM model demonstrated moderate robustness against the attack samples, while CNN and RNN models showed severe degradation in performance, indicating susceptibility to the crafted attacks. These results highlight the importance of designing resilient architectures and using ensemble techniques for better defense.

Bonus Task:

Approach
Ember Model Training
Training Configuration:

- Enabled sequence and set the window size to 51.
- Enabled ember during the ensemble training process.

Results:

- Training Accuracy: 100%
- Validation Accuracy: 99.32%

Ember Model Attack
Attack Configuration:

- Used the Ember Attack module (pe.module.ember-attack) with the hashes generated in Task 5.
- Model (pe.model.zip) trained in Task 4 was utilized for this attack.

Findings:

Attack Execution:

- 242 samples successfully processed using the EMBER Attack module.
- Each sample had unique hash modifications post-attack.

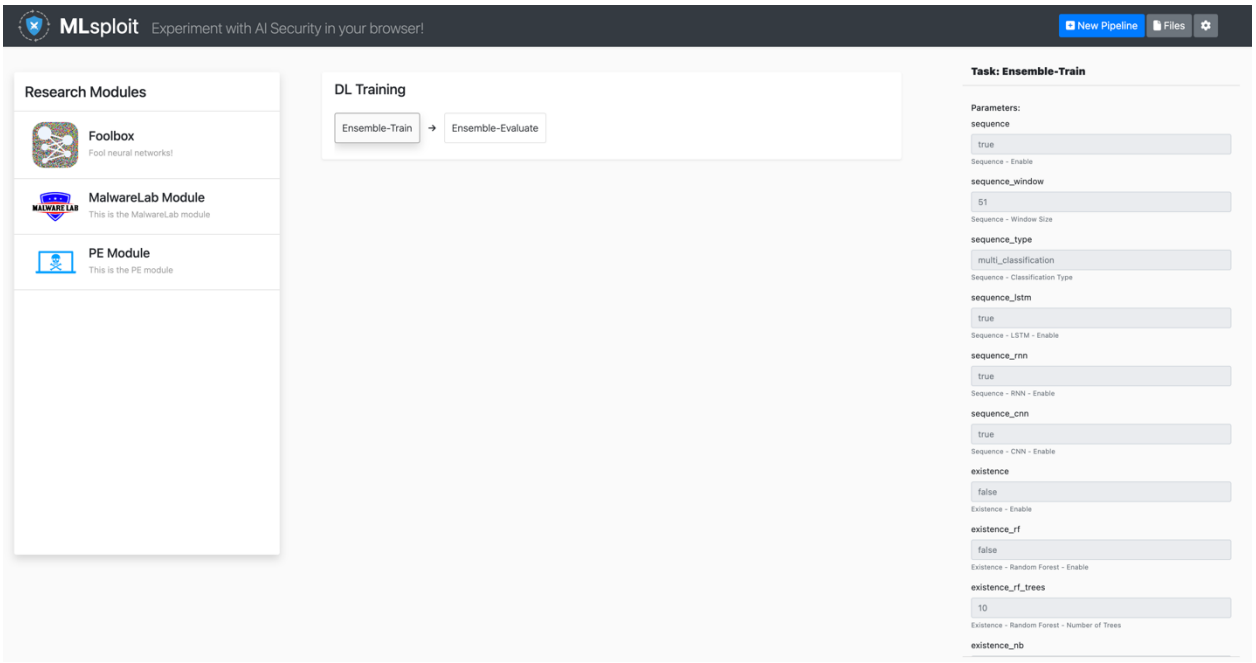
Observations

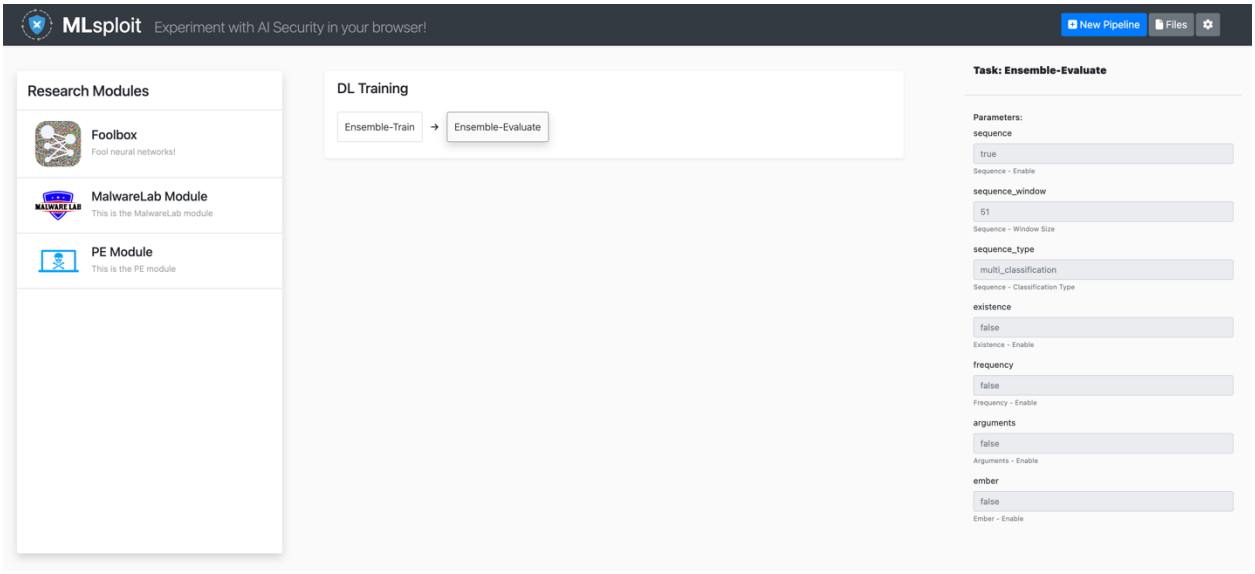
The Ember model exhibited high robustness during initial testing but was significantly impacted by advanced mimicry attacks. Static features provided strong detection capabilities; however, adversarial perturbations targeting these features reduced the model's effectiveness. This highlights the necessity for combining static and dynamic features to build hybrid defense mechanisms.

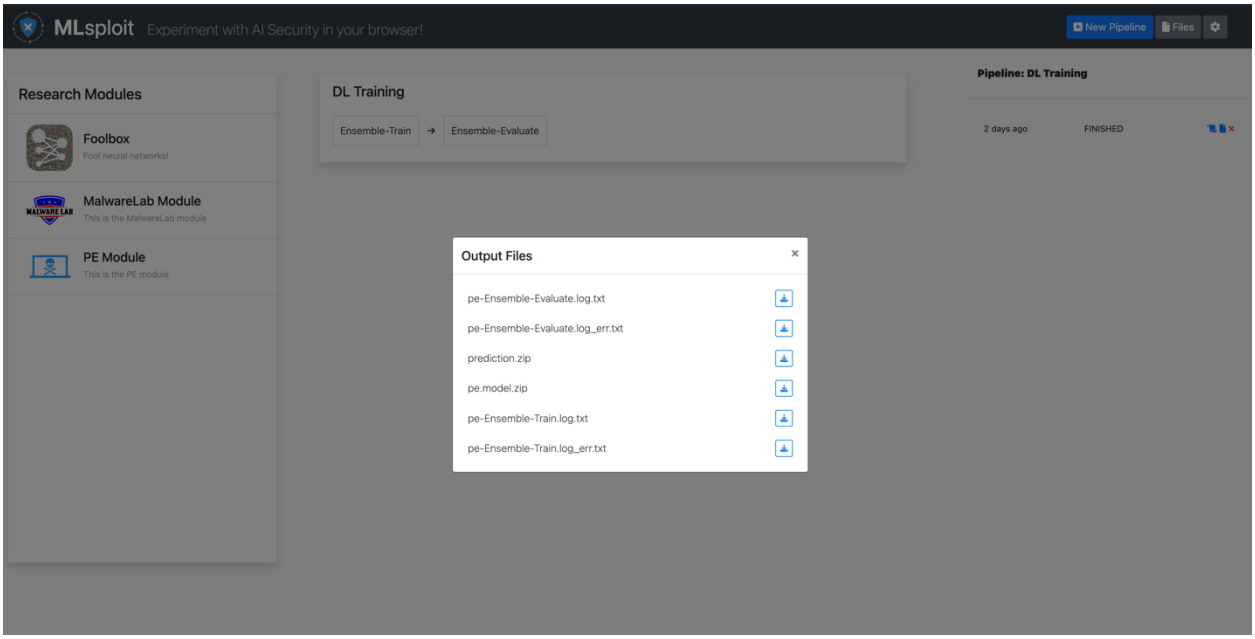
Appendix

Screenshots

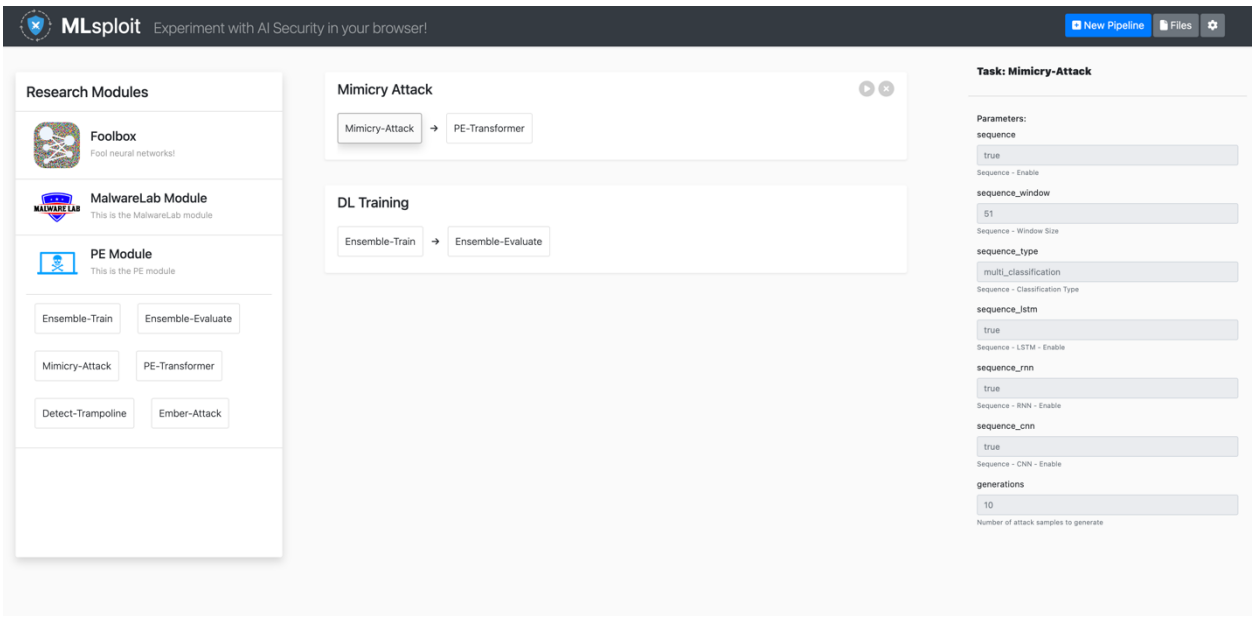
- Training DL Models

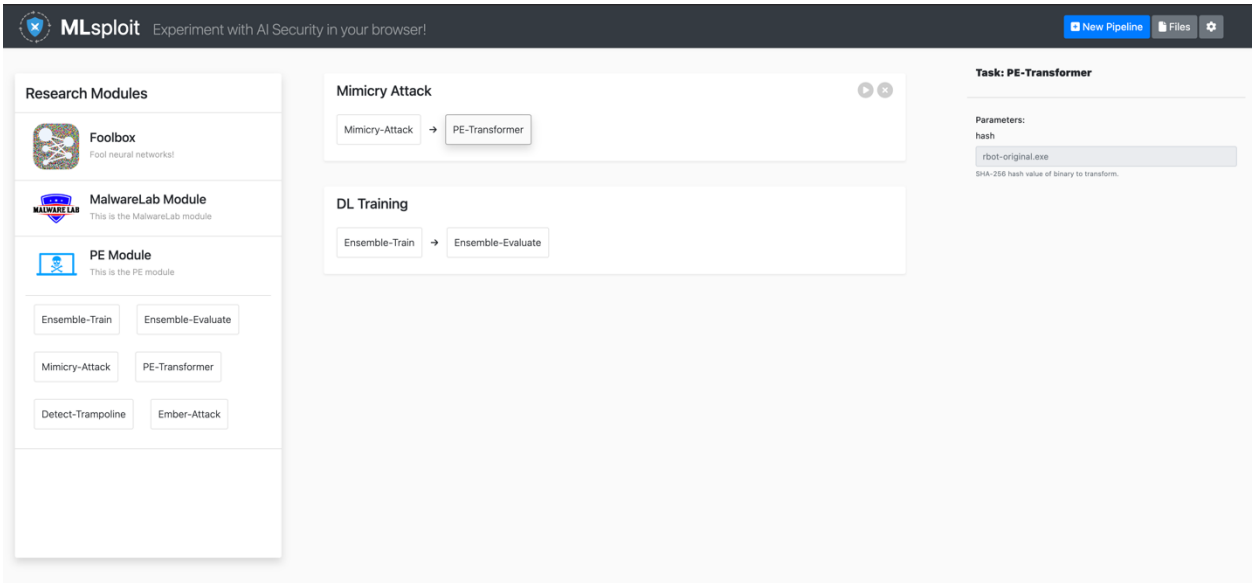


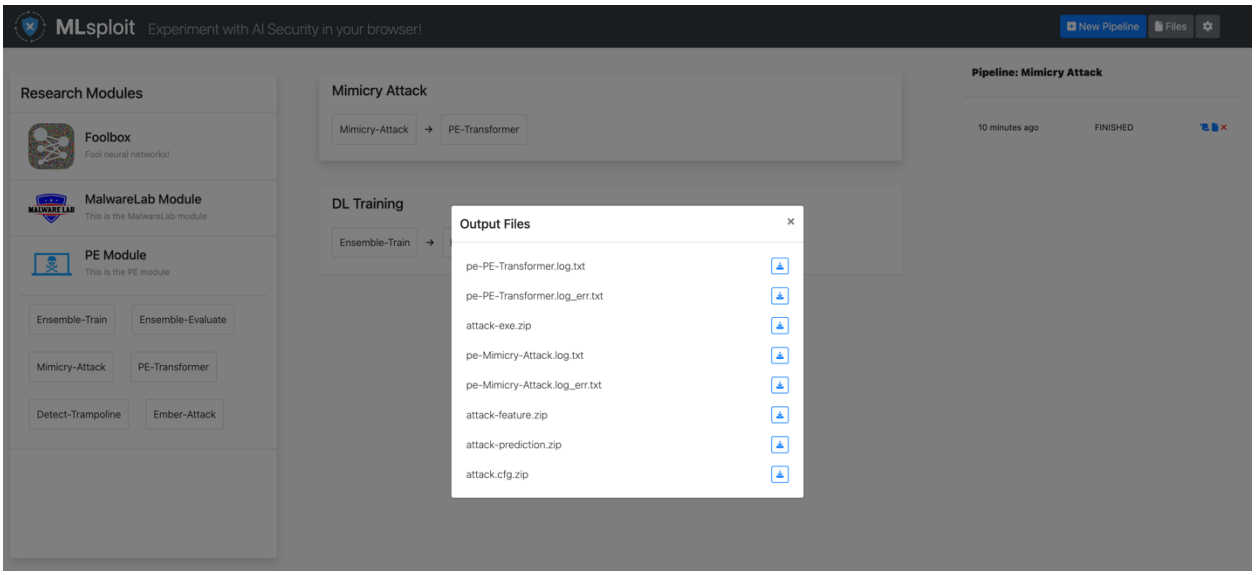

MLsploit
Experiment with AI Security in your browser!
New Pipeline
Files
Research Modules
Foolbox
Fool neural networks!
MalwareLab Module
This is the MalwareLab module
PE Module
This is the PE module
DL Training
Ensemble-Train
Ensemble-Evaluate
Task: Ensemble-Evaluate
Parameters:
sequence
true
Sequence - Enable
sequence_window
51
Sequence - Window Size
sequence_type
multi_classification
Sequence - Classification Type
existence
false
Existence - Enable
frequency
false
Frequency - Enable
arguments
false
Arguments - Enable
ember
false
Ember - Enable


MLsploit
Experiment with AI Security in your browser!
New Pipeline
Files
Research Modules
Foolbox
Fool neural networks!
MalwareLab Module
This is the MalwareLab module
PE Module
This is the PE module
DL Training
Ensemble-Train
Ensemble-Evaluate
Pipeline: DL Training
2 days ago
FINISHED
Output Files
pe-Ensemble-Evaluate.log.txt
pe-Ensemble-Evaluate.log_err.txt
prediction.zip
pe.model.zip
pe-Ensemble-Train.log.txt
pe-Ensemble-Train.log_err.txt

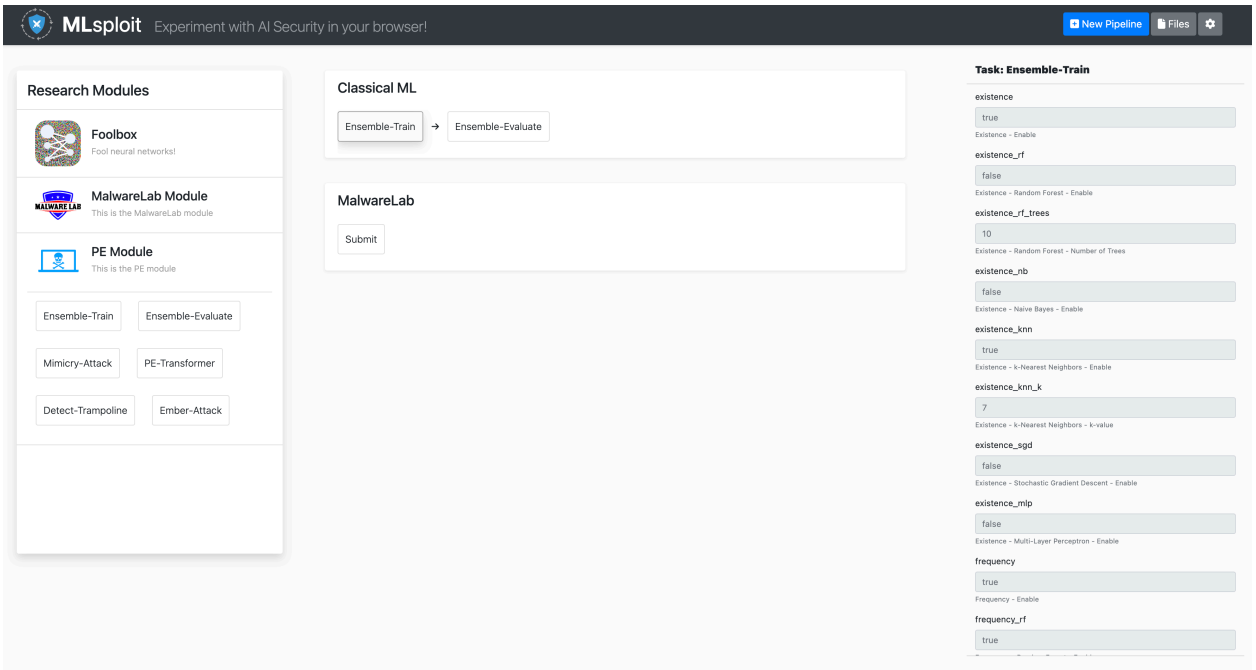
- Attacking DL Models

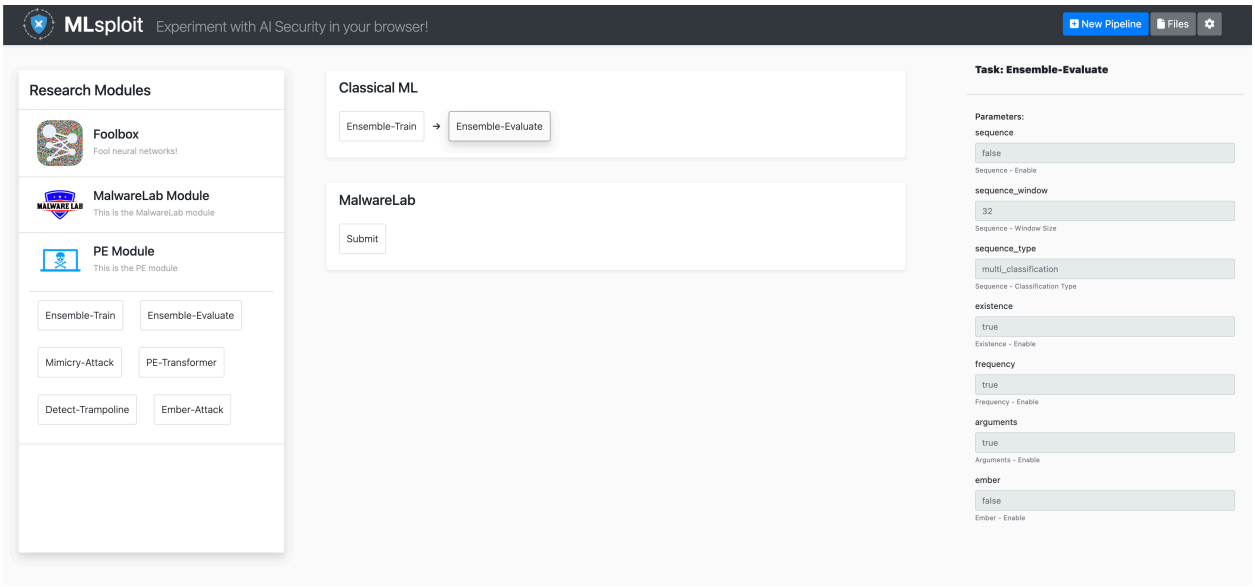


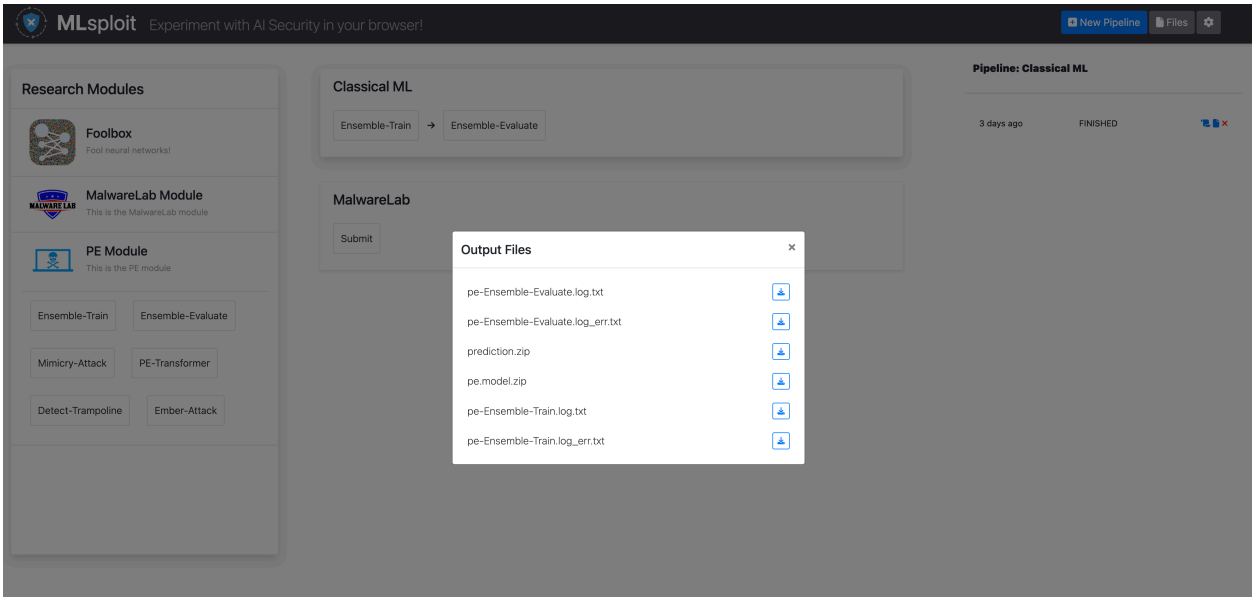




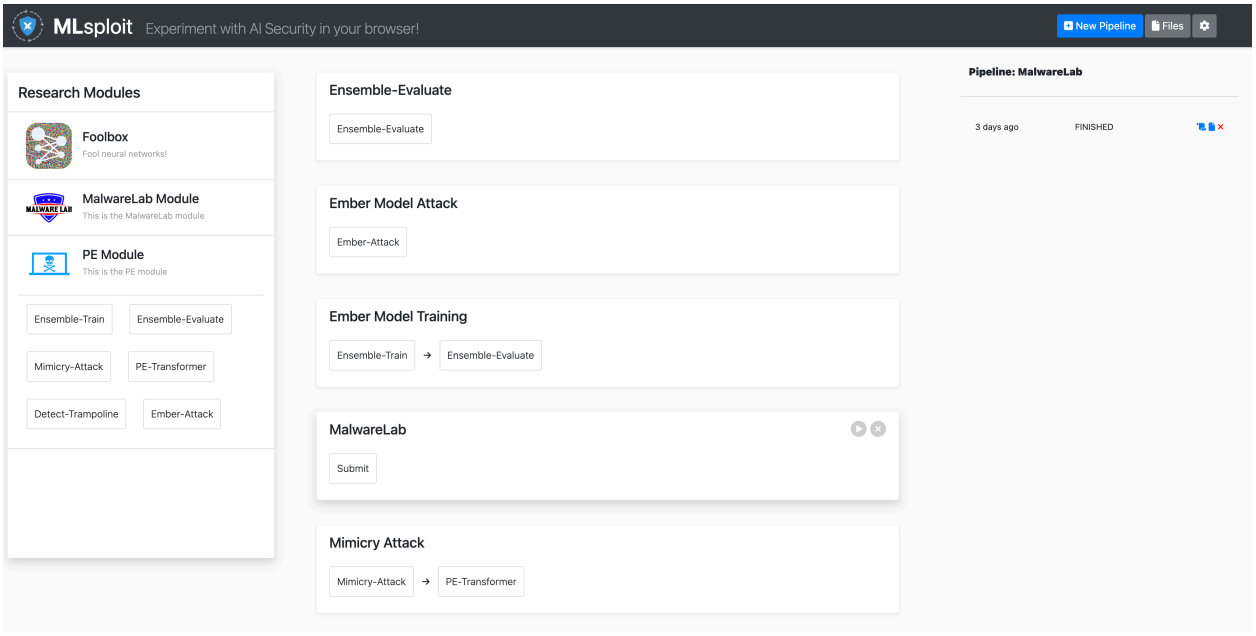
- Training Classic Models

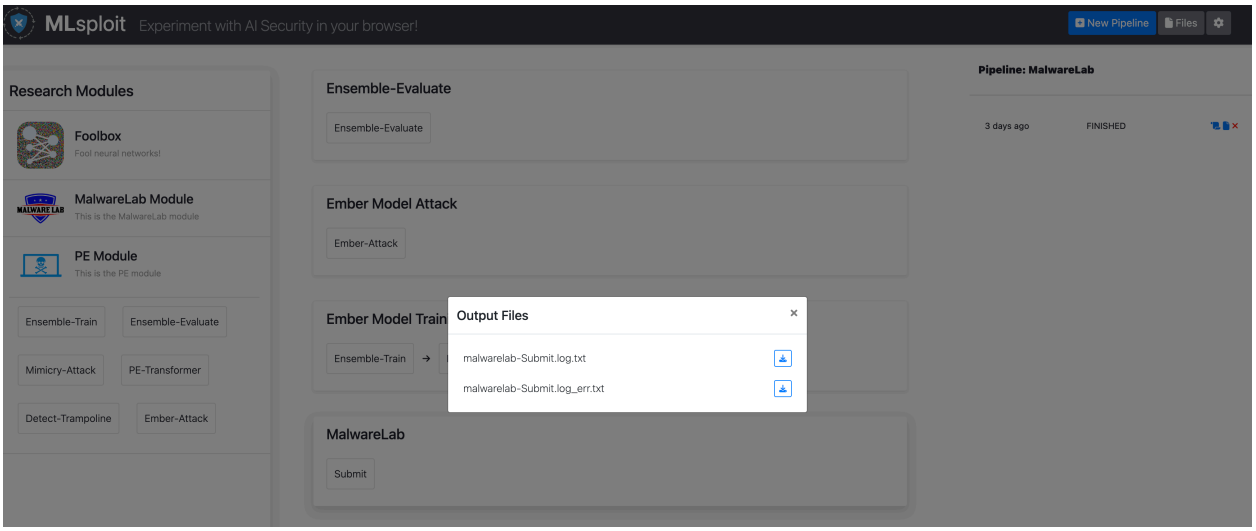


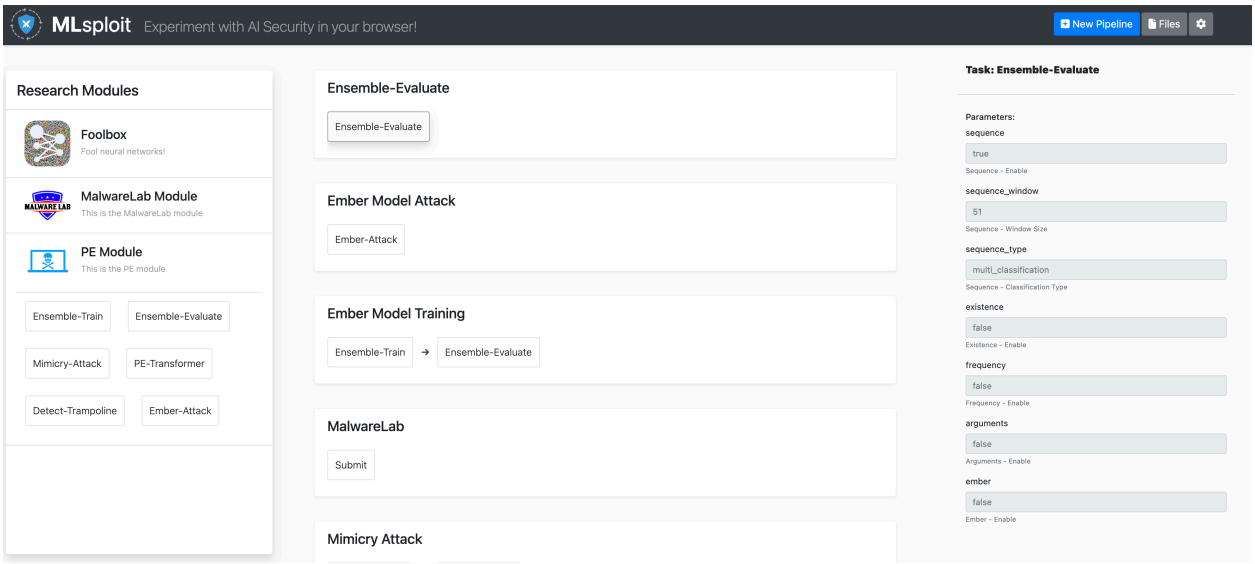

MLsploit
Experiment with AI Security in your browser!
New Pipeline
Files
Research Modules
Foolbox
Fool neural networks!
MalwareLab Module
This is the MalwareLab module
PE Module
This is the PE module
Ensemble-Train
Ensemble-Evaluate
Mimicry-Attack
PE-Transformer
Detect-Trampoline
Ember-Attack
Classical ML
Ensemble-Train
Ensemble-Evaluate
MalwareLab
Submit
Task: Ensemble-Evaluate
Parameters:
sequence
false
Sequence - Enable
sequence_window
32
Sequence - Window Size
sequence_type
multi_classification
Sequence - Classification Type
existence
true
Existence - Enable
frequency
true
Frequency - Enable
arguments
true
Arguments - Enable
ember
false
Ember - Enable

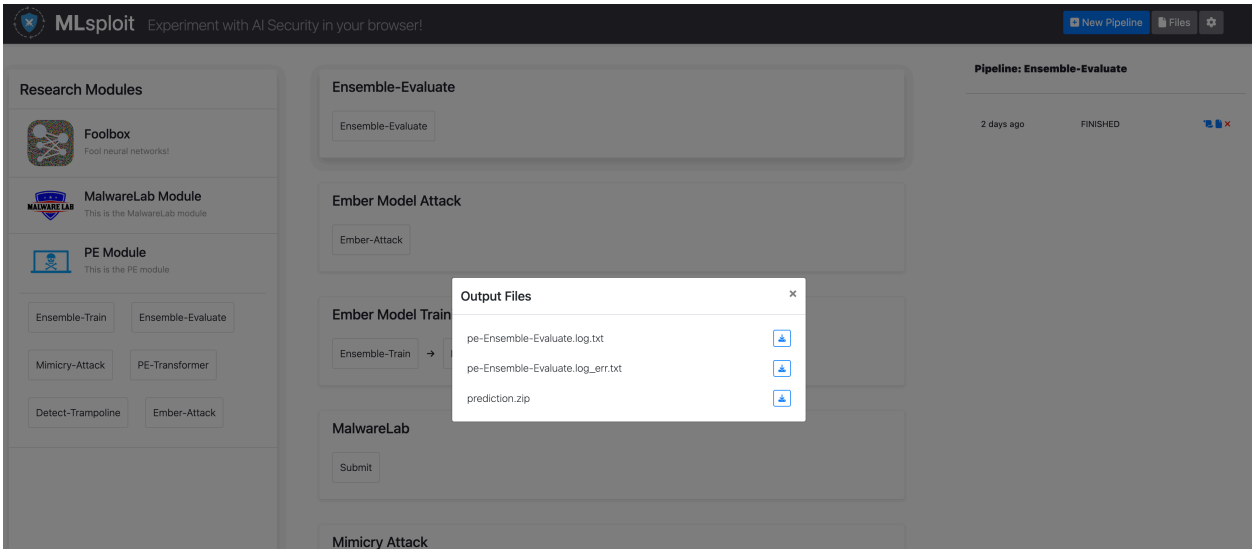

MLsploit
Experiment with AI Security in your browser!
New Pipeline
Files
Research Modules
Foolbox
Fool neural networks!
MalwareLab Module
This is the MalwareLab module
PE Module
This is the PE module
Ensemble-Train
Ensemble-Evaluate
Mimicry-Attack
PE-Transformer
Detect-Trampoline
Ember-Attack
Classical ML
Ensemble-Train
Ensemble-Evaluate
MalwareLab
Submit
Output Files
pe-Ensemble-Evaluate.log.txt
pe-Ensemble-Evaluate.log_err.txt
prediction.zip
pe.model.zip
pe-Ensemble-Train.log.txt
pe-Ensemble-Train.log_err.txt
Pipeline: Classical ML
3 days ago
FINISHED

- Attack Transfer to ML Models

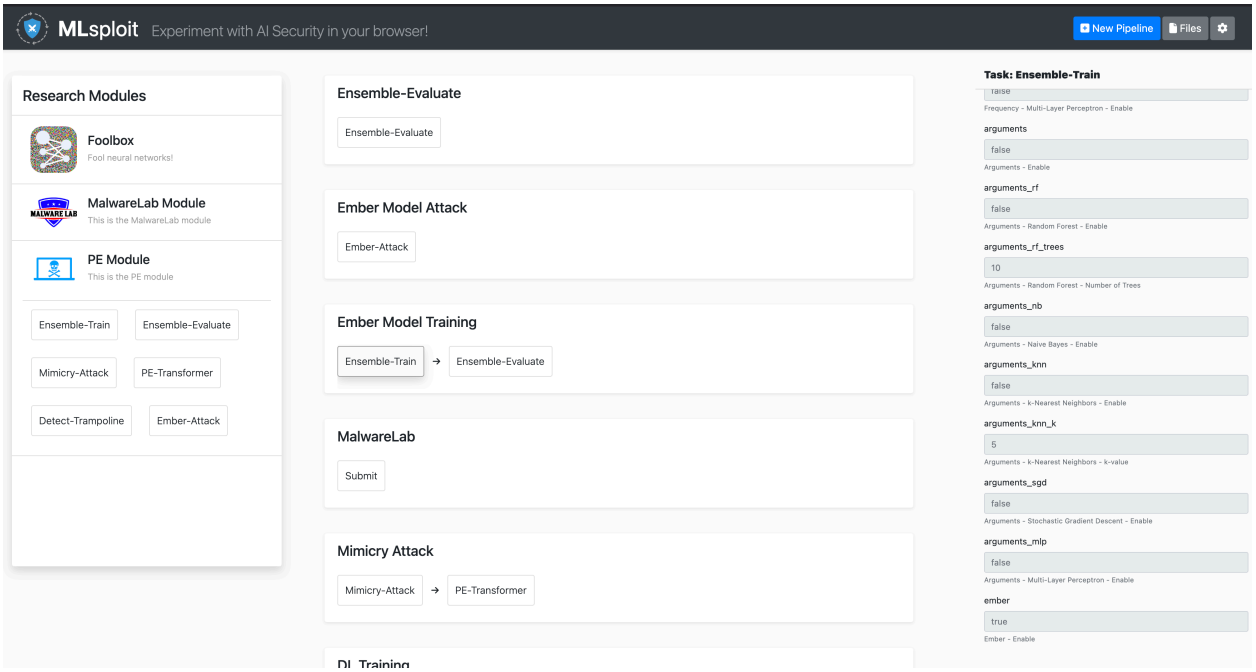


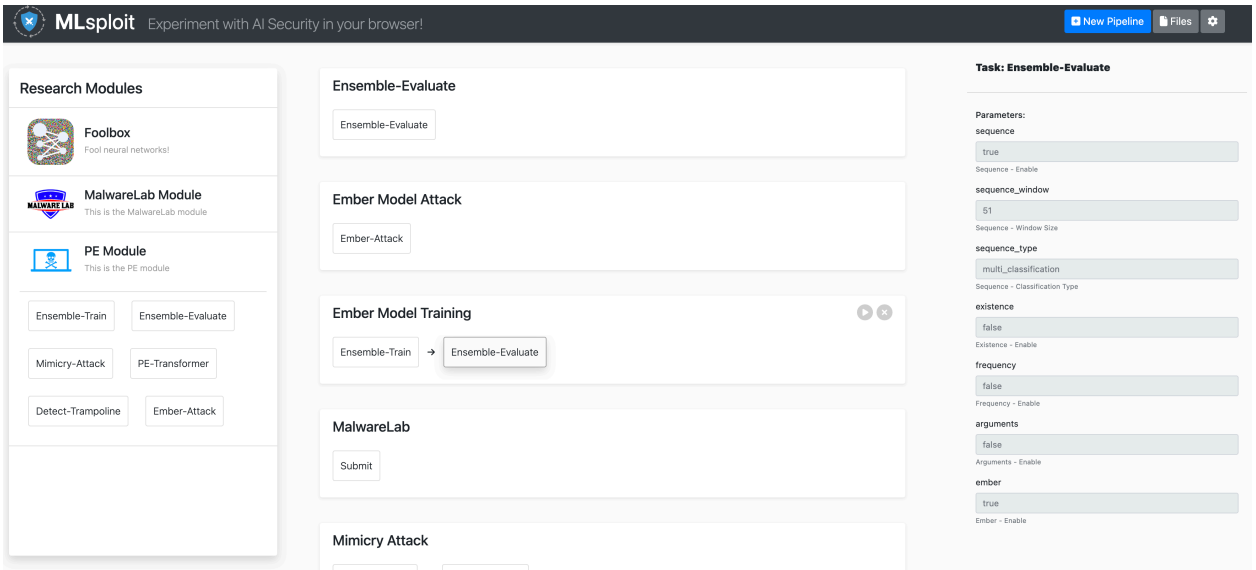


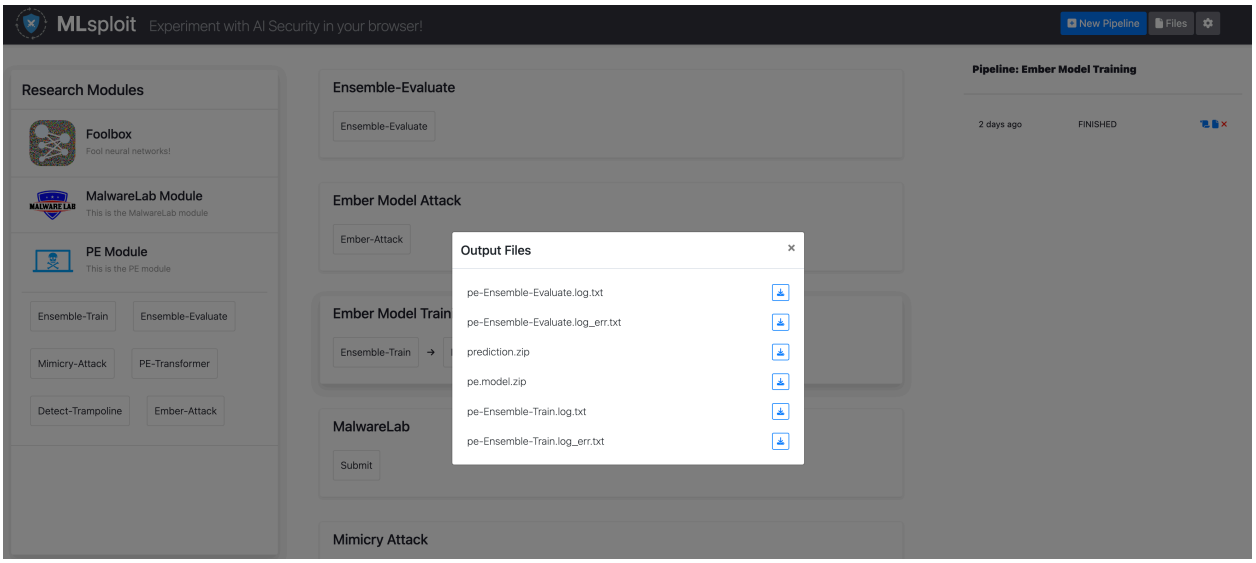

MLsploit
Experiment with AI Security in your browser!
New Pipeline
Files
Research Modules
Foolbox
Fool neural networks!
MalwareLab Module
This is the MalwareLab module
PE Module
This is the PE module
Ensemble-Train
Ensemble-Evaluate
Mimicry-Attack
PE-Transformer
Detect-Trampoline
Ember-Attack
Ensemble-Evaluate
Ensemble-Evaluate
Ember Model Attack
Ember-Attack
Ember Model Training
Ensemble-Train
Ensemble-Evaluate
MalwareLab
Submit
Mimicry Attack
Task: Ensemble-Evaluate
Parameters:
sequence
true
Sequence - Enable
sequence_window
51
Sequence - Window Size
sequence_type
multi_classification
Sequence - Classification Type
existence
false
Existence - Enable
frequency
false
Frequency - Enable
arguments
false
Arguments - Enable
ember
false
Ember - Enable

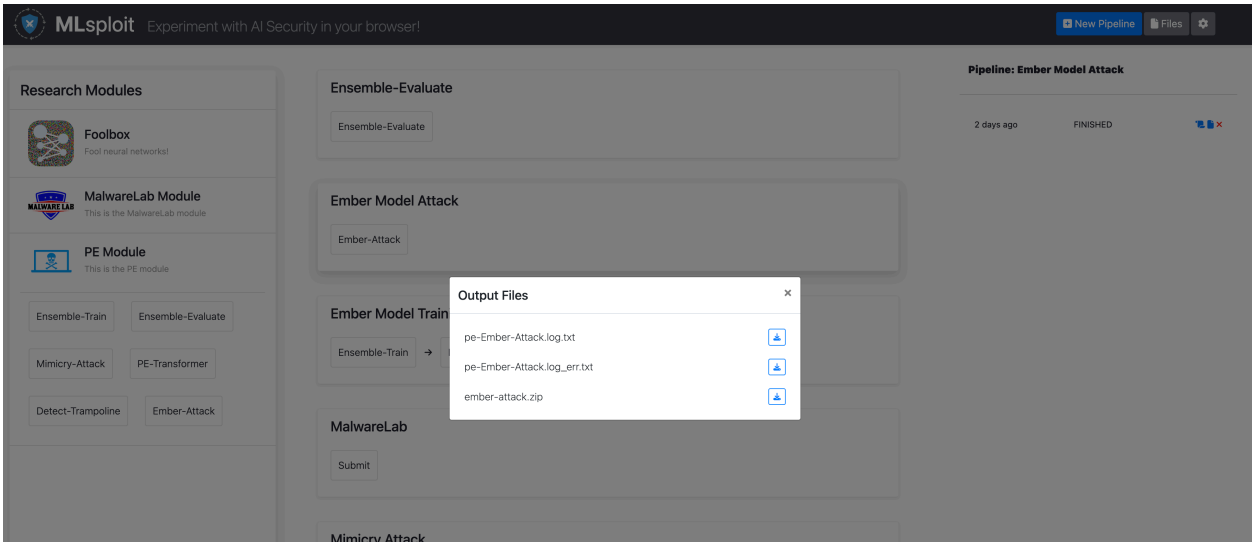

MLsploit
Experiment with AI Security in your browser!
New Pipeline
Files
Research Modules
Foolbox
Fool neural networks!
MalwareLab Module
This is the MalwareLab module
PE Module
This is the PE module
Ensemble-Train
Ensemble-Evaluate
Mimicry-Attack
PE-Transformer
Detect-Trampoline
Ember-Attack
Ensemble-Evaluate
Ensemble-Evaluate
Ember Model Attack
Ember-Attack
Ember Model Train
Ensemble-Train
MalwareLab
Submit
Mimicry Attack
Output Files
pe-Ensemble-Evaluate.log.txt
pe-Ensemble-Evaluate.log_err.txt
prediction.zip
Pipeline: Ensemble-Evaluate
2 days ago
FINISHED

- Bonus Task


MLsploit
Experiment with AI Security in your browser!
New Pipeline
Files
Research Modules
Foolbox
Fool neural networks!
MalwareLab Module
This is the MalwareLab module
PE Module
This is the PE module
Ensemble-Train
Ensemble-Evaluate
Mimicry-Attack
PE-Transformer
Detect-Trampoline
Ember-Attack
Ensemble-Evaluate
Ember Model Attack
Ember Model Training
MalwareLab
Submit
Mimicry Attack
Task: Ensemble-Train
arguments
arguments_rf
arguments_rf_trees
arguments_nb
arguments_knn
arguments_knn_k
arguments_sgd
arguments_mlp
ember


MLsploit
Experiment with AI Security in your browser!
New Pipeline
Files
Research Modules
Foolbox
Fool neural networks!
MalwareLab Module
This is the MalwareLab module
PE Module
This is the PE module
Ensemble-Train
Ensemble-Evaluate
Mimicry-Attack
PE-Transformer
Detect-Trampoline
Ember-Attack
Ensemble-Evaluate
Ember Model Attack
Ember Model Training
MalwareLab
Submit
Mimicry Attack
Task: Ensemble-Evaluate
Parameters:
sequence
sequence_window
sequence_type
existence
frequency
arguments
ember


MLsploit
Experiment with AI Security in your browser!
New Pipeline
Files
Research Modules
Foolbox
Fool neural networks!
MalwareLab Module
This is the MalwareLab module
PE Module
This is the PE module
Ensemble-Train
Ensemble-Evaluate
Mimicry-Attack
PE-Transformer
Detect-Trampoline
Ember-Attack
Ensemble-Evaluate
Ember-Attack
Ember Model Attack
Ember Model Train
Ensemble-Train
MalwareLab
Submit
Mimicry Attack
Pipeline: Ember Model Training
2 days ago
FINISHED
Output Files
pe-Ensemble-Evaluate.log.txt
pe-Ensemble-Evaluate.log_err.txt
prediction.zip
pe.model.zip
pe-Ensemble-Train.log.txt
pe-Ensemble-Train.log_err.txt


MLsploit
Experiment with AI Security in your browser!
New Pipeline
Files
Research Modules
Foolbox
Fool neural networks!
MalwareLab Module
This is the MalwareLab module
PE Module
This is the PE module
Ensemble-Train
Ensemble-Evaluate
Mimicry-Attack
PE-Transformer
Detect-Trampoline
Ember-Attack
Ensemble-Evaluate
Ember-Attack
Ember Model Attack
Ember Model Train
Ensemble-Train
MalwareLab
Submit
Pipeline: Ember Model Attack
2 days ago
FINISHED
Output Files
pe-Ember-Attack.log.txt
pe-Ember-Attack.log_err.txt
ember-attack.zip