LEMNA: Explaining Deep Learning based Security Applications

Presenter: Wenbo Guo
Dongliang Mu, Jun Xu, Purui Su, Gang Wang, Xinyu Xing
Deep Learning is Pervasive in Our Daily Life!

Health care
Self-driving car
Finance
Logistics
Insurance quote
Recruitment
Agriculture
Energy
Deep Learning in Cybersecurity (1)

Malware

Binary analysis

Intrusion detection

Fake account

Phishing

Fraud detection
Deep Learning in Cybersecurity (2)

• Deep learning becomes increasingly popular in cybersecurity
  • Binary reverse-engineering (USENIX’15; USENIX’17; CCS’17)
  • Malware classification (KDD’17; European S&P’17)

• Deep learning outperforms traditional approaches in some security applications
  • E.g., Binary function start identification (USENIX’15)

```
nop; nop; nop; sub esp, 0x4c;
90 90 90 83  ec  4c
```

![Recurrent Neural Network (RNN)]

<table>
<thead>
<tr>
<th></th>
<th>ELF x86</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
</tr>
<tr>
<td><strong>ByteWeight</strong></td>
<td>98.41%</td>
</tr>
<tr>
<td><strong>RNN</strong></td>
<td>99.56%</td>
</tr>
</tbody>
</table>

Presenter – Wenbo Guo
(http://www.personal.psu.edu/wzg13/)
The Concerns of Deep Learning

• Hard to establish a trust for the model behavior
• Difficult to diagnose the model mistakes
• Unlikely to discover the model biases
  • Racial bias of machine learning based criminal prediction system ("Machine Bias" essay)
Explanation of Deep Learning Model (1)

• Definition of explanations: A set of features that make key contributions to the model result (e.g., the important pixels attributive to a classification result)

This is the reason behind an AI decision
Explanation of Deep Learning Model (2)

• Definition of explanations: A set of features that make key contributions to the model result (e.g., the instructions / hex code attributive to an identification result)
Existing Techniques to Derive Explanations

• Proposed in the machine learning community
  • ECCV’14; ICLR’17; ICCV’17; ICML’17; KDD’16; KDD’16; NIPS’17

• Categorized into two kinds
  • Blackbox approach (without the requirement to access a target model)
  • Whitebox approach (with the requirement to access a target model)

• Mainly designed for explaining image recognition applications

• Generally claimed to work perfectly for Convolutional Neural Network (CNN)
One Example of Model Explanation (LIME, KDD’16)

Highest coefficients attaching to the 3 most important features
Limitation of Existing Explanation Techniques

• Difficult to provide an accurate explanation
  • Existing approaches assume the local linearity
  • This assumption will cause errors

• Not suitable for an RNN
  • Existing approaches assume feature independency
  • Can’t model the feature dependency
  • Derive incorrect explanations

Feature independence
Our system design!
LEMNA: **Local Explanation Method using Nonlinear Approximation**

- Explanation technique designed for security applications
  - Providing accurate explanations
    - Supporting locally non-linear decision boundaries
  - Being customized to RNN
    - Modeling the feature dependency

Presenter – Wenbo Guo
(http://www.personal.psu.edu/wzg13/)
LEMNA: **Local Explanation Method using Nonlinear Approximation**

- **Providing accurate explanations**
  - Supporting locally non-linear decision boundaries

- Being customized to RNN
  - Modeling the feature dependency
Supporting Locally Non-linear Decision Boundaries

• Approximating via a mixture regression model:
  • Mixture regression model is a combination of multiple linear regressions
  • Mixture model yields better approximation than linear regression - more accurate explanations

Best component
LEMNA: **Local Explanation Method using Nonlinear Approximation**

- Providing accurate explanations
  - Supporting locally non-linear decision boundaries
- **Being customized to RNN**
  - **Modeling the feature dependency**
Modeling the Feature Dependency

- Mixture regression model with fused lasso:
  - Fused lasso: regularization term that models the feature dependency
  - Forcing the adjacent features have equal weights - encouraging these features to be selected together

\[ \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots \]

**Feature independence**

\[ \beta_1 x_1 + \beta_2 (x_2 + x_3) + \beta_3 (x_4 + x_5) \ldots \]

**Feature dependency**
Deriving an Explanation from DNN with LEMNA

\[ y_i \sim \sum_{k=1}^{K} \pi_k N(\beta_k x_i, \sigma_k^2) \]

subject to \[ \sum_{j=2}^{M} ||\beta_{kj} - \beta_{k(j-1)}|| \leq S \]

\[ \beta_{i1} x_1 + \beta_{i2} (x_2 + x_3) + \beta_{i3} (x_4 + x_5) + \cdots \]

Highest coefficients attaching to the 3 most important features
Explanation Accuracy Evaluation

- Dataset and target models:
  - **Binary function start detection:** RNN (USENIX’15)
  - PDF malware classifier: MLP (ACSAC’17)

- Evaluation Metrics
  - Perturbing the selected features and observing the difference of DNN results
  - Feature Deduction Test: Constructing a new test sample by nullifying the selected features from the original instance

---

**Graphs:**
- **Binary O0**
- **Binary O3**
  - Red: LEMNA
  - Blue: KDD’16
  - Black: Random
Fidelity Evaluation (1)

• Feature Deduction Test
  • Constructing a new test sample by nullifying the selected features from the original instance
Fidelity Evaluation (2)

- Feature Deduction Test

![Graphs showing Feature Deduction Test results for Binary O0, Binary O3, and PDF Malware.]
Fidelity Evaluation (3)

• Feature Augmentation Test
  • Selecting a sample from opposite class and constructing a new instance by replacing the feature values of the opposite sample with those of original instance
Fidelity Evaluation (4)

- Feature Augmentation Test

![Graphs showing PCR (%) vs Nfeatures for Binary O0, Binary O3, and PDF Malware datasets. The graphs compare different methods: GMM-FL, LIME, and Random.](http://www.personal.psu.edu/wzg13/)
Fidelity Evaluation (5)

• **Synthetic Test**
  • Synthesizing a testing sample by preserving the feature values of the select while randomly assigning values for the remaining features
Fidelity Evaluation (6)

- Synthetic Test
Why LEMNA is useful...
Demonstration of LEMNA in Identifying Binary Function Start

• Building trust in the target models
  • Capturing the well-known heuristics
  • Discovering new knowledge

• Troubleshooting and patching model errors
  • Reason for false positives and false negatives
  • Targeted patching of DNN models
Building Trust in the Target Models (1)

- Capturing the well-known heuristics
  - Preparing stack and padding instructions

  ```
  push ebp; mov ebp, esp
  ```

- Discovering new heuristics that cannot be exhausted summarized by human
  - Start of utility function and preparations at the function start

  ```
  ret; sub esp, 0x1c
  ```
Cases (with O0 / O1 compilation options) successfully identified as a function start because of these instructions

<table>
<thead>
<tr>
<th>Explanation</th>
<th>Assembling code</th>
</tr>
</thead>
<tbody>
<tr>
<td>5b 5d c3 55 89 e5</td>
<td>pop ebx; pop ebp; ret;</td>
</tr>
<tr>
<td>5b 90 c3 53 83 ec 18</td>
<td>pop ebx; nop; ret;</td>
</tr>
<tr>
<td>8d b4 26 00 00 00 89 c1 8b 40 0c</td>
<td>lea esi, [esi+eiz*1+0];</td>
</tr>
<tr>
<td>90 90 90 90 56 53</td>
<td>nop; nop; nop; nop; push esi; push ebx</td>
</tr>
</tbody>
</table>

*Building Trust in the Target Models (C.W.H) (2)*
Troubleshooting and Patching Model Errors (1)

• Reason for false negatives
  • Instructions that always appear in the middle of functions that show up around function starts

  \[
  \text{jmp 0xfffff900; xor ebp, ebp}
  \]

• Reason for false positives
  • Indicators for function start appear in the middle of a function

  \[
  \text{ret; sub esp, 0xc1}
  \]
Troubleshooting and Patching Model Errors (R.F.N.) (3)

Cases failing to be recognized as a function start because these instructions typically appear in the middle of a function.
Troubleshooting and Patching Model Errors (3)

- Patching method: correcting a specific error testing sample

- Patching results

<table>
<thead>
<tr>
<th>Applications</th>
<th>Before Patching</th>
<th>After Patching</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FN</td>
<td>FP</td>
</tr>
<tr>
<td>Binary 00</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Binary 03</td>
<td>28</td>
<td>13</td>
</tr>
</tbody>
</table>

Pinpointing the important features → Synthesizing some new training samples → Retraining the target model
Troubleshooting and Patching Model Errors (4)

Patching errors with random hex

<table>
<thead>
<tr>
<th>Applications</th>
<th>Before Patching</th>
<th>After Patching</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FN</td>
<td>FP</td>
</tr>
<tr>
<td>Binary 00</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Binary 03</td>
<td>28</td>
<td>13</td>
</tr>
</tbody>
</table>

Binary O0: 3 1
Binary O3: 28 13

e9 50 fd ff ff 31 ed 5e
jmp 0xffffffff50; xor ebp, ebp; pop esi

AA B4
F3 C2
E2 64
75 6C
Conclusion

• Explanation is important, especially for security applications
• Mixture regression model and Fused lasso can help derive high accurate explanations for the applications we have studied
• There is an emerging need for model explanations in various deep learning based security applications
Thank you very much!