Formal Verification of Tree Ensembles against Real-World Composite Geometric Perturbations

Valency Oscar Colaco & Simin Nadjm-Tehrani

Linköping University, Sweden
Introduction
Machine Learning Model

Stop Sign

Rotation + Changes in Lighting

Scaling + Changes in Lighting

Translation + Changes in Lighting
Real-World Composite Geometric Perturbations are a combination of one affine transformation (e.g., rotation, scaling or translation) along with changes in lighting (darkening or brightening).
These perturbations may result in unpredictable (or unstable) system behavior!
Research Goal
Provide **evidence** to support a **safety assurance case** for **vision-based intelligent systems** that use ML in the presence of these perturbations.
Real-World Composite Geometric Perturbations

Model

Supports

Evidence for Safety Assurance Case

Tree Ensembles
Real-World Composite Geometric Perturbations \rightarrow \text{Formal Verification} \rightarrow \text{Evidence for Safety Assurance Case} \\
\text{Tree Ensembles} \rightarrow \text{Model}
Verifier of Tree Ensembles (VoTE)

- Toolsuite to formally verify that tree ensemble models comply with specified requirements
- Based on equivalence class partitioning and abstract interpretation
- Fast & scalable in terms of memory efficiency
Proposed Solution
Contributions

- A method to capture these perturbations in the interval domain through feature space modelling for vision-based systems

- Realization of the method, implemented as an extension to VoTE, and the application of the method to two case studies
Verification Overview
Abstracting the Perturbations

\[ \lambda(x) = \{1, 4, 2, 3, 5, 4, 7, 1, 2, 0, 7\} \]

* Visualisation for illustrative purposes only
Abstracting the Perturbations

$$\lambda(x) = \{1, 4, 2, 3, 5, 4, 7, 1, 2, 0, 7\}$$

$$\alpha((\lambda(x))) = (\min \lambda(x), \max \lambda(x))$$

$$\alpha((\lambda(x))) = (0, 7)$$

For lightening offset ($\zeta_H = 2$) and darkening offset ($\zeta_L = 0$)

$$\alpha((\lambda(x))) = (\min \lambda(x) - \zeta_L, \max \lambda(x) + \zeta_H)$$

$$\alpha((\lambda(x))) = (0 - 0, 7 + 2) = (0, 9)$$
Correctness & Stability

- **is_correct**: The classifier correctness is proven when it predicts the correct label for an input sample based on the ground truth.

- **is_stable**: The classifier stability is proven when it predicts the same label for all points in the defined perturbation region.
Extended Stability Metrics

- **Robustness**: is_correct ∧ is_stable
- **Fragility**: is_correct ∧ ¬ is_stable
- **Vulnerability**: ¬ is_correct ∧ is_stable
- **Breakage**: ¬ is_correct ∧ ¬ is_stable
Results & Conclusion
## Results - Composite Rotations (MNIST)

**Experimental Parameters:** Rotation Range – 0 to 10°, Lightening Offset = 3, Darkening Offset = 2

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Model Accuracy (%)</th>
<th>Robustness (in presence of perturbations)(%)</th>
<th>Time taken for Verification (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max. Depth</td>
<td>No. of Trees</td>
<td>Random Forest</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>76.59</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>10</td>
<td>80.23</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>15</td>
<td>81.95</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>20</td>
<td>83.20</td>
</tr>
</tbody>
</table>
Conclusion

- Applying formal verification to show that a system exhibits its intended behavior can help increase the trustworthiness of intelligent systems.

- Accuracy as a metric alone, is not enough for convincing safety arguments in safety-critical systems that use machine learning components.

- Apart from safety, these perturbations can violate the security properties of classifiers, and our approach can also be used to verify adversarial robustness.

- Finally, our approach can also be used with different types of verification engines and models depending on the context.
Thank you for your kind attention

Questions? Email me at valency.colaco@liu.se