Privately Evaluating Decision Trees and Random Forests

David J. Wu, Tony Feng, Michael Naehrig, and Kristin Lauter

July, 2016
Machine Learning as a Service

Big Data + Machine Learning = New Applications

- patient profile and symptoms
- recommended treatment plan

NIH
Machine Learning as a Service

Big Data + Machine Learning = New Risks

adversary that compromises cloud service learns patient profile
Machine Learning as a Service

Big Data + Machine Learning = New Risks

malicious client might recover information about the model
Our Work: Decision Trees

- Nonlinear models for regression or classification
- Consists of a series of decision variables (tests on the feature vector)
- Evaluation corresponds to tree traversal

Input: feature vector \([x_1, \ldots, x_n]\)
Fully Private Decision Tree Evaluation

- Only learn $\mathcal{T}(x)$ and minimal information about $\mathcal{T}$ (e.g., bound on size of tree)
- Learns nothing about $x$

input: feature vector $x$

input: decision tree $\mathcal{T}$
Fully Private Decision Tree Evaluation

Focus on model evaluation –
assume server already has model

input: feature vector $x$

input: decision tree $T$
Comparison of Approaches

- Generic methods based on Yao 2PC \[\text{[Yao82, LP09]}\]
- Generic methods based on SWHE \[\text{[BPTG15]}\]
- Our protocol

Not drawn to scale
Comparison of Approaches

- **Generic methods based on Yao 2PC** \([Yao82, LP09]\)
- **Generic methods based on SWHE** \([BPTG15]\)
- **Our protocol**

Slightly more computation, but much smaller bandwidth

Not drawn to scale
Protocol Building Blocks: Comparisons

Require protocol to compare components of client’s feature vector with thresholds.
Comparison Protocol [DGK07, BPGT15]

client input: $x$

comparison protocol

server input: $y$

Learns $1\{x < y\}$

Learns nothing
Private Decision Tree Evaluation

Suppose client knows $b_1$, $b_2$, and the structure of the tree.

Then, client can compute the index of the outcome.
Private Decision Tree Evaluation

Suppose client knows the index of the outcome

Problem reduces to oblivious transfer: treat leaves as database, client knows index
Oblivious Transfer (OT) [Kil88, NP99, NP01]

Client input: index $i$

Server input: database $\{x_1, \ldots, x_n\}$

Oblivious transfer protocol

Learns $x_i$

Learns nothing
Private Decision Tree Evaluation

Suppose client knows the index of the outcome

Problem reduces to oblivious transfer: treat leaves as database, client knows index

leaves become OT database
Private Decision Tree Evaluation

1. Client obtains $b_1, b_2$ using comparison protocol
2. Client uses OT to retrieve classification value

**Problem:** Requires client to learn/know structure of the tree
Hiding the Structure

1. **Padding:** Insert “dummy” nodes to obtain complete tree

![Diagram showing the padding process](image-url)
Hiding the Structure

2. Randomization: Randomly flip decision variables:

$$\bar{b}_i := 1 - b_i$$
Private Decision Tree Evaluation

1. **Server:** Pad and permute the decision tree
2. **Server & Client:** Comparison protocol to compute $b_i$ in permuted tree
3. **Client:** Compute the index $j$ of the leaf node
4. **Client & Server:** Engage in OT to obtain $c_j$

**Theorem.** This protocol is secure against *semi-honest* adversaries.
Further Extensions

evaluating random forests without revealing individual classifications

Ensuring security against malicious adversaries

See paper for details!
Experiments

Implemented private decision tree + random forest protocol

Benchmarks taken between a laptop client and an EC2 server
## Decision Tree Evaluation on ECG Data

<table>
<thead>
<tr>
<th></th>
<th>Security Level</th>
<th>Computation (s)</th>
<th>Bandwidth (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Client</td>
<td>Server</td>
</tr>
<tr>
<td>[BFK+09]</td>
<td>80</td>
<td>2.609</td>
<td>6.260</td>
</tr>
<tr>
<td>[BPGT14]</td>
<td>80</td>
<td>2.297</td>
<td>1.723</td>
</tr>
<tr>
<td>Generic 2PC (Estimated)</td>
<td>128</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>This work</td>
<td>128</td>
<td>0.091</td>
<td>0.188</td>
</tr>
</tbody>
</table>

**Experimental Parameters:**
- Data Dimension: 6
- Depth of Decision Tree: 4
- Number of Comparisons: 6
**Decision Tree Evaluation on ECG Data**

<table>
<thead>
<tr>
<th>Security Level</th>
<th>Computation (s)</th>
<th>Bandwidth (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[BFK+09]</td>
<td>80</td>
<td>2.609</td>
</tr>
<tr>
<td>[BPGT14]</td>
<td>80</td>
<td>2.297</td>
</tr>
<tr>
<td>Generic2PC</td>
<td>(Estimated)</td>
<td>128</td>
</tr>
<tr>
<td>This work</td>
<td>128</td>
<td>0.091</td>
</tr>
</tbody>
</table>

10x faster than previous protocols

**Experimental Parameters:**
- Data Dimension: 6
- Depth of Decision Tree: 4
- Number of Comparisons: 6
Performance for Complete Decision Trees

![Graph showing the performance of complete decision trees with computation time and bandwidth vs. depth of the decision tree. The graph includes lines for client computation, server computation, client upload, and server upload.]
Conclusions

Simple protocols for decision tree evaluation in both semi-honest (and malicious) setting

Semi-honest (and malicious-secure) decision tree protocols provide new computation/communication tradeoffs
Thanks!

http://eprint.iacr.org/2015/386