SCALABLE PRIVACY-PRESERVING DISTRIBUTED EXTREMELY RANDOMIZED TREES FOR STRUCTURED DATA WITH MULTIPLE COLLUDING PARTIES

AMIN AMINIFAR\textsuperscript{1}, FAZLE RABBI\textsuperscript{1,2}, AND YNGVE LAMO\textsuperscript{1}

\textsuperscript{1} WESTERN NORWAY UNIVERSITY OF APPLIED SCIENCES
\textsuperscript{2} UNIVERSITY OF BERGEN

IEEE ICASSP 2021
Outline

• Problem

• Distributed Extremely Randomized Trees

• Secure Multi-Party Computation for Privacy-Preserving Distributed ERT

• Efficient Handling of Large-Scale Data

• Evaluation

• Conclusion
Problem

• Learning classification models from data distributed over multiple parties
• Without sharing of the raw healthcare information, due to privacy and legal concerns
• Horizontally partitioned structured data
Distributed Extremely Randomized Trees
Secure Multi-Party Computation for Privacy-Preserving Distributed ERT
Secure Multi-Party Computation for Privacy-Preserving Distributed ERT

Each data holder party sends personal random seeds to all data holder parties.
Secure Multi-Party Computation for Privacy-Preserving Distributed ERT
Secure Multi-Party Computation for Privacy-Preserving Distributed ERT
Secure Multi-Party Computation for Privacy-Preserving Distributed ERT

Send

SSA_{P_2}^{P_1}, ..., SSA_{P_n}^{P_1}
SSA_{P_1}^{P_2}, ..., SSA_{P_n}^{P_2}
SSA_{P_3}^{P_1}, ..., SSA_{P_n}^{P_n}
SSA_{P_n}^{P_1}, ..., SSA_{P_{n-1}}^{P_n}

Receive

SSA_{P_1}^{P_2}, ..., SSA_{P_1}^{P_n}
SSA_{P_2}^{P_1}, ..., SSA_{P_2}^{P_n}
SSA_{P_1}^{P_3}, ..., SSA_{P_3}^{P_n}
SSA_{P_n}^{P_1}, ..., SSA_{P_{n-1}}^{P_n}
Secure Multi-Party Computation for Privacy-Preserving Distributed ERT

\[ \text{rnd}_\text{sum}^{P_1}_{\text{others}} + (\text{secret}_\text{val}^{P_1} - \text{rnd}_\text{sum}^{P_1}_{\text{self}}) + \]
\[ \text{rnd}_\text{sum}^{P_2}_{\text{others}} + (\text{secret}_\text{val}^{P_2} - \text{rnd}_\text{sum}^{P_2}_{\text{self}}) + \]
\[ \vdots \]
\[ \text{rnd}_\text{sum}^{P_n}_{\text{others}} + (\text{secret}_\text{val}^{P_n} - \text{rnd}_\text{sum}^{P_n}_{\text{self}}) = \text{Sum} \]
Efficient Handling of Large-Scale Data

\[
\begin{align*}
\text{True: } [0, 1] & \quad \text{False: } [1, 1] \\
\text{True: } [2, 1] & \quad \text{False: } [1, 0] \\
\text{True: } [2, 1] & \quad \text{False: } [3, 1] \\
\text{True: } [0, 4] & \quad \text{False: } [1, 1] \\
\end{align*}
\]

\[
\begin{align*}
\text{True: } [0, 2] & \quad \text{False: } [1, 0] \\
\text{True: } [0, 1] & \quad \text{False: } [3, 0] \\
\text{True: } [1, 1] & \quad \text{False: } [4, 1] \\
\text{True: } [1, 5] & \quad \text{False: } [0, 0] \\
\end{align*}
\]

\[
\begin{align*}
\text{True: } [4, 7] & \quad \text{False: } [6, 3] \\
\text{True: } [2, 9] & \quad \text{False: } [8, 1] \\
\end{align*}
\]
Efficient Handling of Large-Scale Data

\[
\begin{align*}
\text{True: } [0, 1] &+ \text{True: } [2, 1] &+ \text{True: } [2, 1] &+ \text{True: } [0, 4] \\
\text{False: } [1, 0] &+ \text{False: } [3, 1] &+ \text{False: } [1, 1] &+ \text{False: } [4, 2] \\
\end{align*}
\]

\[
\begin{align*}
\text{True: } [0, 2] &+ \text{True: } [0, 1] &+ \text{True: } [1, 1] &+ \text{True: } [1, 5] \\
\text{False: } [1, 0] &+ \text{False: } [3, 0] &+ \text{False: } [4, 1] &+ \text{False: } [0, 0] \\
\end{align*}
\]

\[
\begin{align*}
\text{True: } [2, 2] \\
\text{False: } [4, 2] \\
\text{True: } [1, 3] \\
\text{False: } [5, 1] \\
\end{align*}
\]
**Evaluation**

- Criteria of evaluation for privacy-preserving data mining approaches
  - Classification performance, overhead, and privacy

### Table 1: Scalability and privacy comparison against existing techniques

<table>
<thead>
<tr>
<th>Approach</th>
<th>Party</th>
<th>Communication ($N$ is the number of parties)</th>
<th>Min Number of Colluding Parties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Send</td>
<td>Receive</td>
</tr>
<tr>
<td>Distributed ERT</td>
<td>All</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>k-PPD-ERT</td>
<td>Data Holders Mediator</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>$N - 1$</td>
</tr>
<tr>
<td>Shamir [31]</td>
<td>k-1 Parties</td>
<td>$N$</td>
<td>$N - 1$</td>
</tr>
<tr>
<td></td>
<td>One Party</td>
<td>$N - 1$</td>
<td>$N + k - 2$</td>
</tr>
<tr>
<td></td>
<td>The Rest</td>
<td>$N - 1$</td>
<td>$N - 1$</td>
</tr>
</tbody>
</table>
Evaluation

![Graph of F1-score vs. Proportion of Participating Parties](image)

![Graph of Accuracy vs. Proportion of Participating Parties](image)

- **F1-score (%)**: The F1-score increases with the proportion of participating parties. The graph shows that the F1-score for multiple features is higher compared to the waveform.

- **Accuracy (%)**: Similarly, the accuracy also increases with the proportion of participating parties. The multiple features again outperform the waveform in terms of accuracy.

- **Legend**:
  - Red line: Multiple Features
  - Blue line: Waveform
Evaluation

- Time for the Learning Process (s)
- Number of Secure Aggregations

Graphs showing the relationship between the proportion of participating parties and time or number of secure aggregations for different features: Multiple Features and Waveform.
Conclusion

• k-PPD-ERT is an extension of ERT algorithm learning classification models when data is distributed.

• The secure multi-party computation technique for k-PPD-ERT is resilient to the collusion of up to k data holder parties.

• The secure multi-party computation technique for k-PPD-ERT is efficient with respect to the communication overhead.

• Limited participation of data holder parties at every round of the learning process decreases the overhead without any noticeable loss in the learning performance.