Security & Privacy: Contingency table

Date	Authors	Description	version
14-02-2025	F.C. Martin,	Initial version	0.1

Introduction

This document is intended to assess the risk of using the Federated Contingency table algorithm. The document is modelled after the guidelines for describing risks for a federated learning algorithm as described in the vantage6 Security & Privacy document **[1]**.

The first section explains how the algorithm works and which data is shared between the parties. Note that we only discuss data that originates from the privacy sensitive database and not the data that is used by the vantage6 infrastructure. In the second part, we discuss which known federated learning vulnerabilities apply to this algorithm . Finally, we discuss how these vulnerabilities may be mitigated.

Algorithm description

A contingency table is a statistical tool used to analyse the relationship between two or more categorical variables. It displays the data in a matrix format, allowing for easy comparison of the variables' interactions. This method is particularly useful in survey research to identify patterns and correlations within the data.

There are four types of parties involved in the algorithm; (1) The aggregator, (2) the data stations, (3) the client and (4) the vantage6 server **[1]**. Figure 1 presents a flow diagram which explains the different steps of the algorithm, which are then explained in the remainder of this section. Note that the server is not displayed as it merely acts as a communication hub between data station, aggregator and researcher.

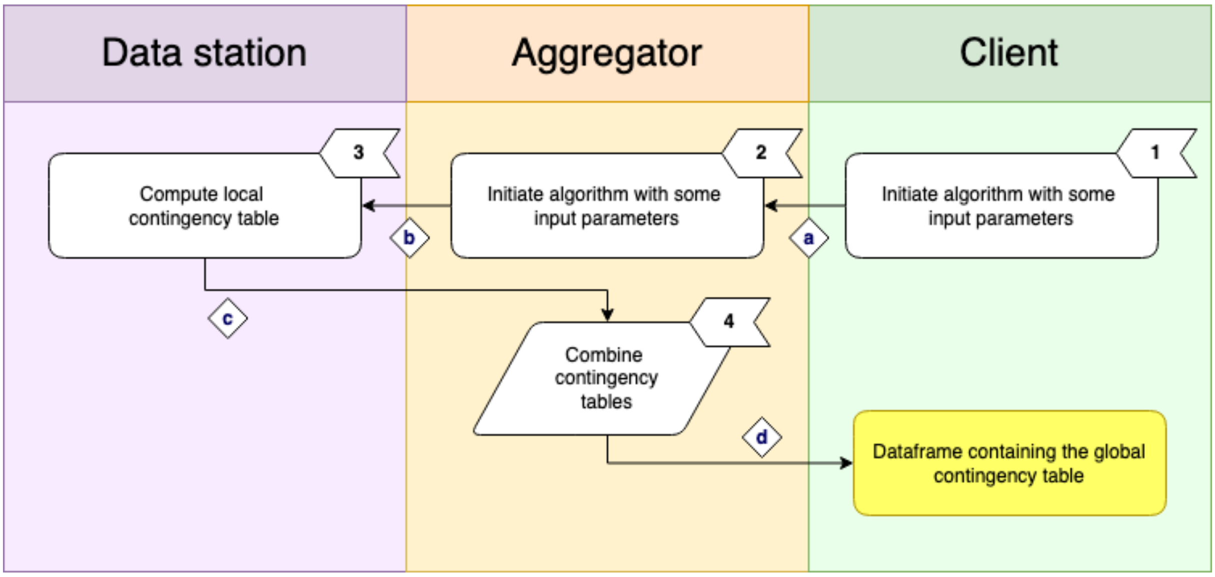
Data station
Aggregator
Client
1
Initiate algorithm with some input parameters
a
2
Initiate algorithm with some input parameters
b
3
Compute local contingency table
c
4
Combine contingency tables
d
Dataframe containing the global contingency table

Figure 1 Schematic of the contingency table algorithm. For a full description of the steps, see section 'Algorithm steps'. Note that all communication between these parties goes via the Central Server.

Algorithm steps

The different steps of the algorithm are shown in Figure 1.

1. User initiates the federated analysis.
2. The aggregator requests the local contingency tables from all the data stations.
3. Each data stations computes their contingency table in their dataset. Values that do not meet the privacy threshold will be masked.
4. The aggregator combines the local contingency tables to the global contingency table.

Data in transit

*Table 1 – A description of data transfers between vantage6 components. Note that all data transfers are mediated by the vantage6 server. The risk level comes from the original paper on Security and Privacy **[1]**.*

Description	Labels	Source	Destination	Risk
Input parameters	a	Client	Aggregator	Low
Input parameters	b	Aggregator	Data station(s)	Low
Contingency table	c	Data station(s)	Aggregator	Medium
Global contingency table	d	Aggregator	Client	Low-Medium

As is indicated in the table above, the transferred data types that are potentially most sensitive are the *contingency tables*. However, whether these are sensitive in practice depends on the privacy guard settings.

Risks

In this section we will look at the types of attack and other kind of risks that the algorithm will be vulnerable to. Not all types of attack are relevant to this algorithm. Please refer back to the Security and Privacy document **[1]** for the various types of attack definitions. It is important to note that the risk analysis is by all means not exhaustive and malicious parties will know more creative techniques.

Attack name	Risk analysis
Reconstruction	
Differencing	May be possible by making smart selection with preprocessing, or by sending multiple tasks before and after data is updated.
Deep Leakage from Gradients (DLG)	Not applicable.
Generative Adversarial Networks (GAN)	Not applicable.
Model Inversion	Not applicable.
Watermark attacks	Not applicable.

Privacy Guards / Mitigation

The aggregator will stop executing when less than three organization are selected for the analysis. Furthermore, there are several settings that can be modified to the needs of the project:

1. **Thresholding**
   The system will only share information if there are at least *n* records in the group. This is to prevent sharing information on individual records. By default, the threshold is set to 5 records. This can be set using the variable `CROSSTAB_PRIVACY_THRESHOLD`.

   Default: `1`

2. **Setting the allowed columns**
   The node administrator can set on which columns they want to allow or disallow the computation of the contingency table. This can be set using the variables: `CROSSTAB_ALLOWED_COLUMNS` and `CROSSTAB_DISALLOWED_COLUMNS`.

   Default: `ALL`

3. **Minimum number of data rows to participate**
   A node will only participate if it contains at least *n* data rows. This is to prevent nodes with very little data from participating in the computation. This can be set using the variable `CROSSTAB_MINIMUM_ROWS_TOTAL`.

   Default: `3`

4. **Not allowing zero values**
   By default, the system will share zero value counts in the contingency table. You can disable by setting `CROSSTAB_ALLOW_ZERO` to `FALSE`.

   Default: `True`

References

[1] Martin, F., van Gestel. A., van Swieten, M., Knoors D., van Beusekom, B., Geleijnse, G., 2023. 'Security and Privacy using vantage6 for Privacy Enhancing Analysis'.