

# **D4.2 Report on the implementation of deep learning algorithms on distributed frameworks**

**Authors: Ohad Arnon (EMC), Andreas Trügler, Samuel Sousa (KNOW), Stefan Gindl, Abdel Aziz Taha, Michael Boch (RSA), George Margetis, Manos Adamakis (FORTH)**

June 2022

# TRUSTS Trusted Secure Data Sharing Space

## D4.2 Report on the implementation of deep learning algorithms on distributed FW

### Document Summary Information

Grant Agreement No	871481	Acronym	TRUSTS
Full Title	TRUSTS Trusted Secure Data Sharing Space		
Start Date	01/01/2020	Duration	36 months
Project URL	<a href="https://trusts-data.eu/">https://trusts-data.eu/</a>		
Deliverable	D4.2 Report on the implementation of deep learning algorithms on distributed FW		
Work Package	WP4 Privacy preserving technologies		
Contractual due date	30.06.2022	Actual submission date	30.06.2022
Nature	Report	Dissemination Level	Public
Lead Beneficiary	EMC		
Responsible Author	Ohad Arnon		
Contributions from	KNOW, RSA, FORTH, EMC		



## Revision history (including peer reviewing & quality control)

Version	Issue Date	% Complete <sup>1</sup>	Changes	Contributor(s)
V0.1	12.06.2022	75	Initial Deliverable Structure	Andreas Trügler (KNOW), Lukas Helminger (RSA), Samuel Sousa (KNOW), Abdel Aziz Taha (RSA), Ohad Arnon (EMC), George Margetis (FORTH), Manos Adamakis (FORTH), Michael Boch (RSA)
V0.2	14.06.2022	85	Peer Review	Gianna Avgousti (eBOS), Samuel Sousa (KNOW), Stefan Gindl (RSA)
V0.3	21.06.2022	90	Addressing peer review comments	Ohad Arnon, Samuel Sousa
V0.4	21.06.2022	90	Peer Review	Ioannis Routis (Nova)
V0.5	24.06.2022	95	Addressing peer review comments	Ohad Arnon, Samuel Sousa, Michael Boch, Manos Adamakis
V1.0	30.06.2022	100	Final submission version	Ohad Arnon (EMC)

## Disclaimer

The content of the publication herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the TRUSTS consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the TRUSTS Consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the TRUSTS Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

<sup>1</sup> According to TRUSTS Quality Assurance Process:

1. to be declared



## Copyright message

© TRUSTS, 2020-2022. This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both. Reproduction is authorised provided the source is acknowledged.



## Table of Contents

1	Executive Summary	10
2	Introduction	11
2.1	Mapping Projects' Outputs	11
3	Privacy Preserving Data Analytics	14
3.1	Introduction	14
3.2	Technical solutions for privacy-preserving and federated data analytics	16
3.2.1	Review of deep learning methods for privacy-preserving data processing	16
3.2.2	Cryptographic solutions for privacy-preserving analytics	19
3.2.2.1	Merging Multi-Key Homomorphic Encryption with Federated Deep Learning	20
3.2.2.2	Private Set Intersection	23
3.2.3	Further federated machine learning implementations for TRUSTS	24
3.2.3.1	Ensemble Learning/Modelling	24
3.2.3.2	Vertical Federated Learning using SHAP values	29
3.2.3.3	Superseded federated learning	33
3.3	Application	36
4	Anonymisation and De-Anonymisation	38
4.1	Introduction	38
4.1.1	Task background and motivation	38
4.1.2	Contribution	41
4.2	De-Anonymisation Risk Analysis and Anonymisation Modules	43
4.2.1	Privacy models and anonymisation strategies	43
4.2.2	Tabular Data	46
4.2.3	Location Data	46
4.2.4	Textual Data	47
4.2.5	Invoice Data	50
4.2.6	Aggregated Data	51
4.3	Application	52
4.3.1	Supported Datasets	52
4.3.2	Application Architecture	52
4.3.3	Application implementation	55
4.3.4	Application functionality	56
4.3.5	Risk Analysis results	63
4.3.6	Anonymization Results	64
5	Application to TRUSTS Platform	65
5.1	TRUSTS Platform Infrastructure	65
5.2	TRUSTS Platform and Secure Computation	66
5.3	UC2	68



6	Conclusions and Next Actions	70
7	References	71



## List of Figures

Figure 1: Literature search results adapted from Sousa and Kern (2022)	17
Figure 2: Publication years of the works selected by Sousa and Kern (2022)	18
Figure 3: Schematic depiction of federated learning using multi-key homomorphic encryption	21
Figure 4: Performance of model training with and without HE for different numbers of workers	22
Figure 5: High level architecture illustrating how to collaborate using ensemble modeling	24
Figure 6: Experiment results of Ensemble modeling on different data sets having the same feature set - results per party	26
Figure 7: Experiment results of Ensemble modeling on different data sets having the same feature set – Ensemble results	26
Figure 8: Experiment results of Ensemble modeling on different data sets having different feature set - results per party	28
Figure 9: Experiment results of Ensemble modeling on different data sets having different feature set - Ensemble	28
Figure 10: The way to collaborate with SHAP values	29
Figure 11: SHAP values collaboration – data flow chart	30
Figure 12: SHAP values collaboration – flow chart	31
Figure 13: Illustrative example - first federated model results	32
Figure 14: Illustrative example - second federated model results	32
Figure 15: Illustrative example - Common model results	33
Figure 16: common sample IDs cross datasets	34
Figure 17: Superseded Federated Learning flow	35
Figure 18: Experiment Illustration – classification with missing data	36
Figure 19: Experiment Illustration – classification with GAN data	36
Figure 20: The intersection of two semi-publicly available datasets (Sweeney, 2000)	39
Figure 21: Illustration of k-anonymity and l-diversity (Machanavajjhala et. al, 2007)	40
Figure 22: Screenshot of the k-anonymity risk analysis module on a contracts datasets for k=2	44
Figure 23: Examples of enhanced visualisations with more descriptive tooltips and legends	45
Figure 24: Screenshot of the l-diversity risk analysis module for l=2 with enhanced UI	46
Figure 25: Example hierarchies from Bampoulidis et al (2019)	46
Figure 26: Screenshot of the enhanced output for spatiotemporal data risk analysis (gowalla dataset)	47
Figure 27: Demonstration of different cluster methods (6 clusters); Different countries in Europe and Africa	47
Figure 28: Screenshot of textual data risk analysis (AOL search logs and Amazon reviews)	49
Figure 29: Screenshot of financial transactions data risk analysis with enhanced UI	50
Figure 30: Time series clustering demonstration with stock data adapted from Petitjean et al. (2011)	51
Figure 31: Screenshot of the aggregation-based data risk analysis	51



Figure 32: Application architecture diagram	53
Figure 33: Sign in Page	57
Figure 34: Datasets Page – card view	58
Figure 35: Upload Dataset Page	59
Figure 36: Dataset Info Page	59
Figure 37: Risk Analysis Page	60
Figure 38: De-anonymization Risk Analysis Page - The results of the process configured in Figure 37	61
Figure 39: Anonymization Page	62
Figure 40: Processing Queue Page	63
Figure 41: De-anonymization Risk Analysis Page	64
Figure 42: High level architecture of collaborating over private personal data using HE and Spooky shared encryption key	67
Figure 43: High level architecture of MPC protocol, to collaborate over private sensitive data, while all of the computation is done on TRUSTS servers	67
Figure 44: T4.3 application usage flow	68
Figure 45: Roles and Interaction in the Industrial Data Space	69
Figure 46: The TRUSTS Architecture	69

## List of Tables

Table 1: Adherence to TRUSTS GA Deliverable & Tasks Descriptions	11
Table 2: Collaborating over different data sets with the same feature set– Training stage	25
Table 3: Collaborating over different data sets with the same feature set – Execution stage	25
Table 4: Collaborating over different data sets with different feature set – Training stage	27
Table 5: Collaborating over different data sets with different feature set – Execution stage	27





## Glossary of terms and abbreviations used

Abbreviation / Term	Description
AI	Artificial intelligence
DL	Deep learning
DP	Differential privacy
FHE	Fully homomorphic encryption
FL	Federated learning
FW	Frameworks
GA	Grant Agreement
GAN	Generative adversarial network
GDPR	General Data Protection Regulation
GUI	Graphical user interface
HE	Homomorphic encryption
HFL	Horizontal federated learning
ML	Machine learning
MPC	Multi-party computation
NLP	Natural language processing
PET	Privacy-enhancing technology
PII	Personally identifiable information
PSI	Private set intersection
QID	Quasi-identifier
RF	Random forest
ROC	Receiver operating characteristic curve
SHAP	SHapley Additive exPlanations
TRUSTS	Trusted Secured Data Sharing Space
UC	Use case
UI	User interface
VFL	Vertical federated learning
WP	Work package



# 1 Executive Summary

This deliverable is part of the Work Package (WP) 4 “Privacy preserving technologies”, of the TRUSTS project addressing task 4.4 “Federated Deep Learning methodologies”, task 4.5 “Transformation of algorithms to privacy-preserving certified” and task 4.3 “Anonymisation and de-anonymisation”. It aims to showcase the usage of compute-intense neural networks over several nodes under the TRUSTS platform.

In this deliverable we build upon the work of D4.1 “Algorithms for Privacy-Preserving Data Analytics” and describe several solutions for privacy preserving analytics that were developed and implemented within TRUSTS, focussing especially on federated deep learning techniques. We start with a description of technical solutions for privacy-preserving and federated data analytics, where we began with a general review of available solutions. We then highlight some of our cryptographic approaches to the problem, where in the first half funding period of TRUSTS we have developed prototypes for encrypted transfer learning or efficient private set intersection (PSI) that were now refined and used in the use case (UC) trials of TRUSTS. We also discuss a way to increase the security of federated learning (FL) by merging it with homomorphic encryption (HE), so that also the model updates and gradients are protected. We developed a working prototype for encrypted FL and merged our solution with the open-source machine learning (ML) suite SystemDS. We then discuss in detail several improvements on the ML side, where we have implemented ensemble learning methods, a combination of explainability methods with FL based on SHapley Additive exPlanations (SHAP) values and did research on superseded FL. Some of these implementations are again integrated with TRUSTS and have been used at the corresponding UC trials.

In the latter part of this deliverable, we focus on anonymisation and de-anonymisation, where we have developed de-anonymisation risk analysis models and corresponding anonymisation methods. These methods for risk analysis and anonymisation were collected for 5 complex data types: tabular, location, textual, invoice and aggregated data. The identification of these methods was carried out on the basis of scientific literature. The risk of de-anonymisation is a central building block for secure evaluations and with our models we provide an extensive and easy to use framework for a general analysis of anonymized data. Also, here our solutions have been integrated with the TRUSTS platform.



## 2 Introduction

D4.2 “Report on the implementation of deep learning (DL) algorithms on distributed frameworks” aims to showcase the usage of compute-intense neural networks over several nodes under the TRUSTS platform. The deliverable is part of the WP4 “Privacy preserving technologies” where the objective is to investigate, design and improve cryptographically secure protocols that enable data analysis of privacy-sensitive data. Consequently, we will focus on practical aspects of cryptographic building blocks such as, but not limited to, HE.

FL is a rather new ML technique (it was first introduced by Google in 2017) that quickly became very popular due to several advantages. It allows joint ML evaluations among a plethora of clients and at the same time follows a privacy-by-design principle, where the ML algorithm is brought to the data and not vice versa. This allows that training and evaluation data remain at their local primitives and only model weights and gradients are transmitted to a central server where the final model is aggregated from all client input. However, the efficiency of FL also comes at a price, in the last years several attacks on FL systems have been published where for example the knowledge of the transmitted model gradients is enough to also reconstruct the original training data. Thus, one should always be aware of the efficiency-privacy trade-off and carefully consider the best method with corresponding security guarantees for each UC.

### 2.1 Mapping Projects’ Outputs

Purpose of this section is to map TRUSTS Grand Agreement commitments, both within the formal Deliverable and Task description, against the project’s respective outputs and work performed.

Table 1: Adherence to TRUSTS GA Deliverable & Tasks Descriptions

TRUSTS Task		Respective Document Chapter(s)	Justification
T4.4 Federated Deep Learning methodologies	This task constitutes a horizontal layer of the TRUSTS architecture facilitating the federated training and utilization of the envisaged Deep Learning algorithms, which will be incorporated in the platform, by distributed devices, running on the edge of the system’s cloud. A cloud based	Section 3	



	framework will be deployed enabling the distribution, training, inference, monitoring and update of existing artificial intelligence (AI) models to selected distributed clients, which will be able to utilize local isolated content repositories. To this end, each federated deployment is enabled to use private or sensitive datasets for the generation of the necessary feedback to the TRUSTS platform, without endangering their unauthorized access or exposing the data source.		
T4.5 Transformation of algorithms to privacy-preserving certified	This task will strive to convert risky algorithms that compromise privacy into safe and privacy-preserving without harming their functionality. Various algorithms ought to use external sources and run computation to execute certain functions. The development of most algorithms is driven by outcome and performance, leaving privacy and security issues on the least of requirements. The challenge is in retrofitting and enabling working algorithms to perform under the desired set of privacy regulations without the need of redevelopment.		
T4.3 Anonymization and de-anonymization	In this task, we are performed de-anonymization risk analysis on the provided datasets, in order to find out the extent to which they comply with privacy regulations. After applying de-anonymization risk analysis and	Section 4 Section 5	Section 4: Anonymization and De-Anonymization Section 5: Application to TRUSTS Platform



	getting an insight into the data, we are going to inquire about the necessary anonymization measures that need to be taken. This task will collaborate with T6.1 to ensure that data protection rules and principles are respected. Since anonymization measures distort the data, a great challenge in this task is to find the golden mean between privacy and utility of the data; the extent to which anonymization measures should be applied. During the task, tools and guidelines on de-anonymizing and anonymizing the data will be developed, as well as measures of the de-anonymizability of the data and conformity to anonymity principles.		
<b>TRUSTS Deliverable</b>			
D4.2 Report on the implementation of deep learning algorithms on distributed frameworks			
This report will showcase the usage of compute-intense neural networks over several nodes under the TRUSTS platform. This deliverable is related to T4.4			



## 3 Privacy Preserving Data Analytics

### 3.1 Introduction

Throughout the centuries cryptographic ciphers have been designed to protect stored data or, with the emergence of modern information transmission, also to protect data in transmission. These scenarios usually follow an all-or-nothing principle where e.g. two parties can access full information and outsiders nothing or where only the data owner has full information and nobody else. In reality trust relationships are often a lot more complicated and diverse of course as we have seen in the previous sections, especially when it comes to outsourcing computations or accessing pre-trained ML models.

In our last report we were focusing on different encryption algorithms that allow collaboration over private and sensitive data while preserving data privacy. We introduced CryptoTL (currently under submission but also available as preprint<sup>2</sup>) for example, where we show for the first time a cryptographic privacy-preserving transfer learning approach based on HE that is efficient and feasible for real-world UCs. In general, one of the major challenges of cryptographic privacy-preserving algorithms is the efficiency-privacy trade off - the higher the privacy guarantees the lower the computational performance. This challenge has a huge impact on adapting those algorithms in the industry, which is seeking for high performance and reliable methods to preserve privacy while collaborating over private and sensitive data.

In order to solve this challenge and to provide an efficient way to collaborate over private and sensitive data, we must use those algorithms only on critical junctions while integrating them in federated and distributed computation methods that will enable efficient collaboration over private and sensitive data while preserving data privacy.

In WP4, we focus on the underlying cryptographic primitives as well as privacy-preserving ML methods (transfer and FL) and anonymization. The specific tasks in WP4 are:

- T4.1 Privacy Preserving Data Analytics
- T4.2 Privacy Preserving Transfer Learning and Classification
- T4.3 Anonymization and de-anonymization
- **T4.4 Federated Deep Learning methodologies**
- **T4.5 Transformation of algorithms to privacy-preserving certified**

The topic of this deliverable is related to T4.4, T4.5 integrating with T4.1, T4.3 and contains a description of selected methods of FL and modeling that enable privacy preserving of private and sensitive data on collaboration.

When it comes to FL, there are two different types to consider before developing a model. The first is horizontal federated learning (HFL), which is introduced in the scenarios where data sets

---

<sup>2</sup> <https://arxiv.org/abs/2205.11935>



share the same feature space but are different in sample. This type of collaboration is very rare when it comes to different companies and different domains, and it is highly common in mobile phones UCs.

The second is vertical federated learning (VFL), which is applicable to the cases where two or more data sets share the same sample ID space but differ in feature space. This scenario is much more common in the industry. Therefore, we focus on it in the TRUSTS project.

In order to find the intersection between the collaborating parties' data sets, without revealing the data and while preserving the data privacy, a secure PSI protocol is used as the first stage of any collaboration.

Once the sample IDs are known, the common analytics can be reached via different VFL methods as will be described below.



## 3.2 Technical solutions for privacy-preserving and federated data analytics

FL is a rather new and very popular technique that has been introduced by Google (McMahan et al., 2017) and follows the principle of bringing the algorithm to the data in comparison to sending data to a remote evaluation somewhere. Thus, it is a decentralized learning protocol where private and sensitive data never have to leave their local storage location, instead only model parameters are transmitted and updated on a central server (e.g. service provider) or cloud. In a first step, local devices (mobile phones, computer nodes, etc.) download the ML model from the central server, perform a training step with local data and send back the updated weights or model parameters to the server where all contributions are merged together. An overview of the current state of the art and future trends can be found in (Kairouz et al., 2019; Yang et al., 2019; Li et al., 2020; Bonawitz et al., 2019) for example. FL for general purposes has also been integrated into libraries like Tensorflow (TensorflowFL, 2020) or Apache SystemDS (Boehm et al., 2020).

Due to different breakthroughs in recent years and the development of new ML and cryptographic approaches the often-illustrated problem that one must give up privacy if you want to do data analytics is not true anymore. Several methods allow for more privacy, ranging from transfer or federated ML to cryptographic primitives like HE or multi-party computation (MPC). The corresponding security guarantees depend on the respective method, usually one has to find the UC related trade-off between security and performance or usability.

We start this section with a review of DL methods for privacy-preserving data processing (exemplary for textual data) and continue with cryptographic improvements of classical FL systems, where we have implemented a prototype for encrypted federated model updates and included it to the open-source ML system Apache SystemDS<sup>3</sup>. We also briefly summarize the application of our library for PSI that was developed within SafeDEED and TRUSTS and allows two parties to compute the intersection of their data sets without having to disclose the data to each other. PSI is a special-purpose secure MPC protocol and differs from the idea of classical FL. However, we have decided to include a short description of our implementation here as well, as it fits into the wider context of privacy-preserving data analytics of multiple parties and our implementation is also used in the TRUSTS UC trials. We end this section with a discussion of further BSOTA ensemble learning methods, as well as vertical and superseded FL and show how our solutions and implementations are used within TRUSTS.

### 3.2.1 Review of deep learning methods for privacy-preserving data processing

DL models often process private and sensitive data, which demands protection against breaches and disclosures. Especially data from domains, such as finance, bio-medicine, social media, and image, which inherently present private or sensitive content (Sousa & Kern, 2022). The General Data Protection Regulation (GDPR) confers the right to privacy to European citizens who

---

<sup>3</sup> <https://systemds.apache.org>





generate data for DL applications and requires companies that develop and maintain these applications to comply with legal terms on data collection, storage, and processing. Furthermore, a wide variety of attacks can target deep neural networks, aiming to retrieve training data instances or pieces of private information from training datasets, such as demographic attributes, location, income, health status, home address, contact details, etc. Therefore, privacy preservation is a bottom line for DL model development.

In recent years, privacy-enhancing technologies (PETs) have gained attention in the literature on DL because of the ever-increasing number of applications that use these technologies to hinder privacy issues for personal data. Noticeable PETs frequently combined with DL include differential privacy (DP), adversarial learning, FL, among others. As a result, the literature on privacy-preserving DL presents ramifications that make it difficult to have a holistic view of the current developments in this research field. This broad literature hardens the search for baselines and UC requirements by practitioners in the industry. For this reason, part of the efforts performed towards task 4.4 and task 4.5 consisted of conducting an extensive literature review on DL methods for privacy-preserving data processing, with a special focus on data in natural language format, by Sousa and Kern (2022).

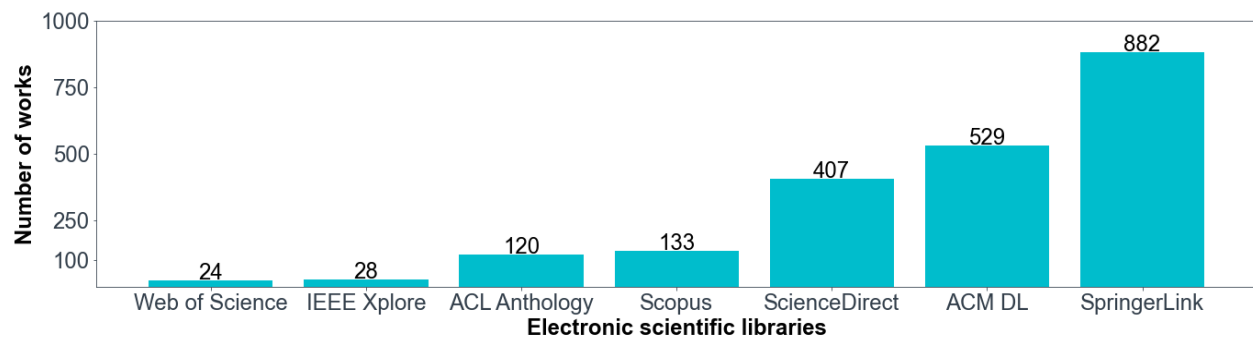


Figure 1: Literature search results adapted from Sousa and Kern (2022)

The article titled “How to keep text private? A systematic review of DL methods for privacy-preserving natural language processing”, published in Artificial Intelligence Review<sup>4</sup>, followed a systematic review structure to guide the collection, selection, and organization of scientific papers into categories. First, we constructed search strings using terms related to ‘privacy’, ‘privacy preservation’, ‘deep learning’, and ‘natural language processing’ (for the full list of terms, we recommend the reader to refer to Sousa and Kern (2022)). Second, we used the search terms we created to perform searches in the 7 electronic scientific libraries shown in Figure 1. These searches returned 2,123 papers in total. Third, we applied a series of inclusion and exclusion criteria to reduce the number of papers for the review and mitigate any selection bias that could prioritize some works over others unfairly. Finally, after applying such criteria to search results and including prominent papers from e-print archive, we had a collection of 63 works to review as Figure 2 shows. These works, published since 2016, constituted the most

<sup>4</sup>Artificial Intelligence Review, accessed on May 25, 2022, <<https://www.springer.com/journal/10462>>.



recent advances on DL-based privacy-preserving natural language processing at the time of writing this review.

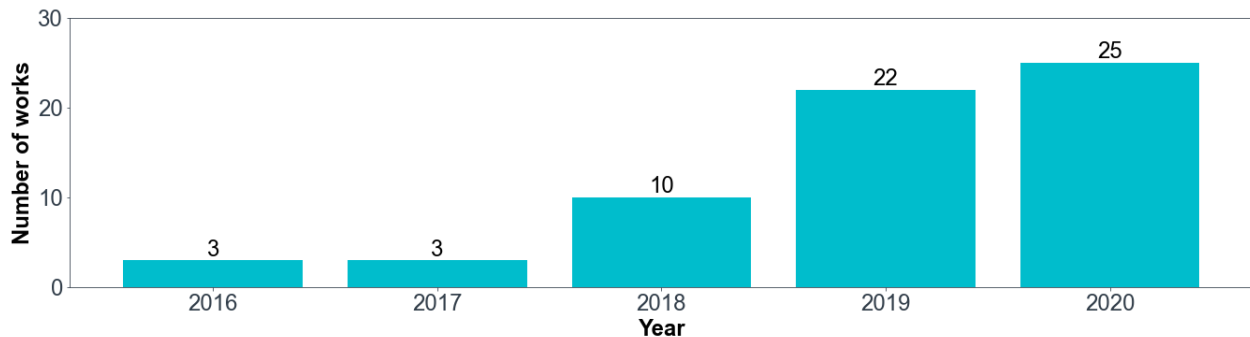


Figure 2: Publication years of the works selected by Sousa and Kern (2022)

Privacy protection for data processing is an endeavour with solution depending on many factors, such as computation scenario, utility performance, memory footprint, dataset size, data properties, natural language processing (NLP) task, and DL model. As a result, the choice of a suitable PET is not solely a problem of protecting the greatest extent of privacy as possible because privacy-utility trade-offs can turn a solution feasible for a real-world application or, otherwise, impractical. In the past few years, many works have been addressing it for DL and text data processing yet lacking categorization. Therefore, Sousa and Kern (2022) proposed a taxonomy that organizes this literature and shapes the landscape of DL methods for privacy-preserving text data processing.

When it comes to privacy-preserving text data processing approaches based on DL, Sousa and Kern (2022) have found similarities between methods considering two major factors: the target of privacy preservation and the PETs specifically. The first factor determines where privacy is assured, such as in the dataset prior to model training and inference, on model components during the learning phase, or in post-processing routines. The second factor specifies which existing PETs are appropriate for each privacy scenario. For example, encryption is often recommended when the server where the data is stored, or another computation party, is no longer trusted. Then, Sousa and Kern (2022) grouped the methods which implement encryption schemes for utility tasks of NLP into a group of encryption methods. Additionally, since encryption methods are commonly implemented alongside a DL model and remain in place during model training and inference, they inserted this group into the category of methods whose privacy focus is on the model side, namely trusted methods. This category is divided into two sub-categories according to the computation scenarios for which the trusted methods are implemented (distributed parties and cloud environments). In a similarly manner, this insight was used to construct a taxonomy composed of two levels, three categories, seven sub-categories, and sixteen groups for the surveyed DL methods and their respective PETs. We recommend the reader to refer to Sousa and Kern (2022) for further details in the full taxonomy.



Besides the categorization of works in the literature, this review has shed light on open challenges on DL for privacy-preserving data processing from five perspectives. First, an adversary attacker should not be able to trace the private version of an anonymized or de-identified document. Second, privacy preservation generally comes at the cost of computation overheads related to model run-time, memory footprint, and bandwidth consumption. Third, dataset size plays an important role in the decision for a PET in real-world DL applications since it affects model training and generalization. Further, human biases and algorithmic fairness are two critical privacy-related topics in data processing due to the ethical consequences they cause on automatic decision-making. Finally, the management of trade-offs arising from the issue of exchanging privacy preservation for performance on model tasks.

In the context of TRUSTS, this review relates to two out of three TRUSTS UCs, namely UC2: “Agile Marketing through data correlation” and UC3 “Data Acquisition to Improve Customer Support Services”. For instance, the solutions developed for these UCs receive data in text format, which may feature pieces of private information. Moreover, Sousa and Kern (2022) reviewed PETs of interest for TRUSTS, such as encryption, MPC, FL, and DP. This review can be a starting point to aid in developing privacy-preserving data processing models and guiding successive research. Open challenges on privacy preservation, like threats and computational costs related to PETs in DL, are discussed in a holistic view that includes data pre-processing, model training and inference, and post-processing routines. Therefore, Sousa and Kern (2022) bridge the gap between foundations of privacy-preserving DL methods and industry tasks for text data by approximating research directions of TRUSTS to both scientific community and the industry.

### 3.2.2 Cryptographic solutions for privacy-preserving analytics

FL has a huge efficiency advantage compared to cryptographic privacy-preserving methods, but also the corresponding security guarantees are different. Recently we have seen several attack papers that highlight the vulnerability of classical FL methods, where for example knowledge of the model weights of a DL network might be sufficient to also reconstruct the corresponding training data. Thus, a combination of cryptographic primitives and FL are a possible solution, where for example the transmitted model updates are encrypted between client and server. In TRUSTS we have implemented several software solutions with focus on such combinations of cryptography and ML, in the following we discuss especially our solution for encrypted FL and a specific example for secure MPC.

**Secure MPC.** The problem of outsourcing computations or sharing data is trusting other parties. Secure MPC protocols aim to get rid of the trust assumptions and allow several mutually distrusting parties to jointly evaluate a public function on their combined input (Rechberger & Walch, 2022). The problem of MPC has been around for several decades (Goldreich et al., 1987) and first MPC protocols were introduced by Yao (1986). In recent years, MPC has undergone a transition from protocols of mostly theoretical interest to having its first practical



implementations and instantiation. The challenges involved when designing such protocols are the complexity of the calculation (Albrecht et al., 2015), the number of rounds needed by the protocol and the necessary assumptions (Garg & Srinivasan, 2018), and the design or choice of suitable symmetric primitives (Grassi et al., 2016). MPC protocols also find applications in the statistical analysis of federated databases. Platforms such as Sharemind (Bogdanov et al., 2014) support the calculation of mean, variance, standard deviation, frequency tables and quantiles, as well as hypothesis tests, all in a privacy-preserving manner. Yet, while such platforms exist, analysis of large data sets brings them to their limits. These MPC protocols are intrinsically interactive approaches. When also considering noninteractive approaches, HE and other methods for data aggregation and computation on encrypted data become interesting.

**Homomorphic Encryption.** Fully Homomorphic Encryption (FHE) (Gentry, 2009a; Gentry, 2009b) is an encryption scheme that evaluates a class of functions or circuits on encrypted data. In this case, all data providers encrypt their data sets for a dedicated receiver and send it to a dedicated aggregator that then evaluates the function on the ciphertexts but neither learns the input data nor the result of the computation. After the aggregation the data is forwarded to the receiver. This approach however has the drawback, that it requires to encrypt the data for every possible receiver and the aggregation has to be performed independently for each set of ciphertexts. These issues can partly be mitigated by homomorphic proxy re-authenticators (Derler et al., 2017) that only require re-aggregation, but no additional computations on the sender side. This approach lacks the expressiveness in the class of functions that could be computed on the encrypted (and signed) data, though. Nevertheless, while in some cases those homomorphic schemes may lack the capability to evaluate arbitrary functions, they complement secure multiparty computation-based approaches.

### 3.2.2.1 Merging Multi-Key Homomorphic Encryption with Federated Deep Learning

There are several compelling UCs for ML that involve highly privacy-sensitive training data. Recent research suggests that the gradient updates of FL methods convey a lot of information about the training data (Yin et al., 2021). It seems that the privacy gained by employing FL is little. However, these compelling UCs for FL on private data would provide huge benefits especially regarding computational performance.

There have been several ideas on how to improve the privacy of FL. Most ideas fall into two broad categories: Either employing a secure MPC scheme to the full training function or using HE to make the aggregation part of FL oblivious to a server performing the aggregation. The former solutions all incur a massive performance and network overhead, rendering them impractical. The latter perform better, however, typically the computation overhead is still a factor of 1000. Nevertheless, since HE is only necessary for a small fraction of the calculations the overall performance is usually within a tolerable scope. HE schemes are designed to protect the data from a server where the computation is outsourced. If more than one client is involved, the handling of the cryptographic keys is not trivial anymore, since the data of the client should now also be protected among the other participants.



Recently, a multi-key HE scheme was proposed that offers a solution to this problem and protects the data from both the server and other clients (Ma et al., 2022). It is a multi-key extension of the well-known CKKS (Cheon et al., 2017) scheme. The clients use an aggregated public key to encrypt their data, where the corresponding private keys are kept secret. The server computes an encrypted sum of all client data and sends it back to the clients. They perform a so-called partial decryption, which can be combined with all other partial decryptions into the unencrypted sum without revealing the individual summands. We depict a schematic description of FL using multi-key HE in the following image.

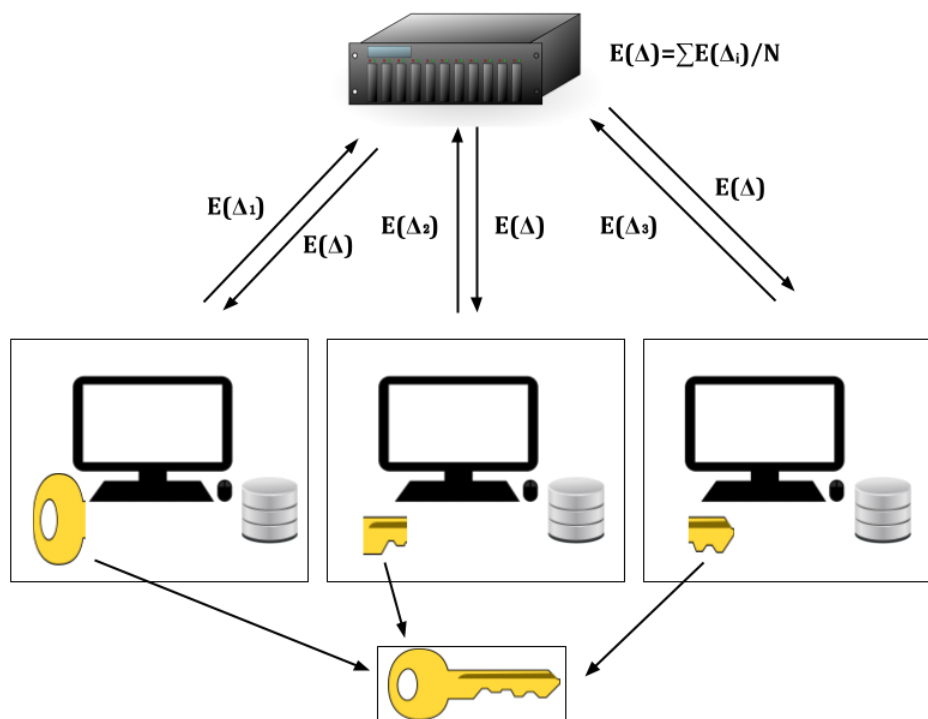


Figure 3: Schematic depiction of federated learning using multi-key homomorphic encryption

In order for this to work, the clients need to choose the shared public key in a certain way. This mandates a key exchange step before the actual training can start. The keys of the clients can be interpreted as an instance of a secret-sharing cryptographic scheme. The shared private key corresponding to the shared public key is never directly instantiated. It can be computed from different secrets held by the clients.

So, as long as at least one client does not collude with the others it is impossible to get hold of the shared private key and break privacy this way. However, if all clients but one collude, they can subtract their gradients from the sum of all gradients to get the gradients of the only honest client. This is a flaw inherent to every aggregation protocol. The protocol is thus secure if at least two clients do not collude. This is a strong privacy improvement over plain FL.

We have implemented this protocol in Apache SystemDS (Boehm et al., 2019), an industry-strength ML software suite. We have made extensive performance tests to quantize the performance cost of using HE with FL. One benchmark can be seen in Figure 4 where we compare the runtime of FL for different numbers of workers with and without using HE. Our results hint that the overall runtime performance on a cluster is less than 10% worse using HE. The effectiveness of the learning process is not reduced. Interestingly, the contrary seems to be true. The random noise introduced by the HE scheme improves the learning efficiency. This is a known effect of adding randomness to the training calculations. The network overhead is approximately a factor of three. However, our experiments show that the networking and aggregation part do not contribute a lot to the overall runtime. The vast majority of the time is spent calculating the gradient updates. This part of the training stays unchanged with the HE protocol. We have chosen the parameters of the benchmark in a way that reduces network and aggregation overhead to a minimum.

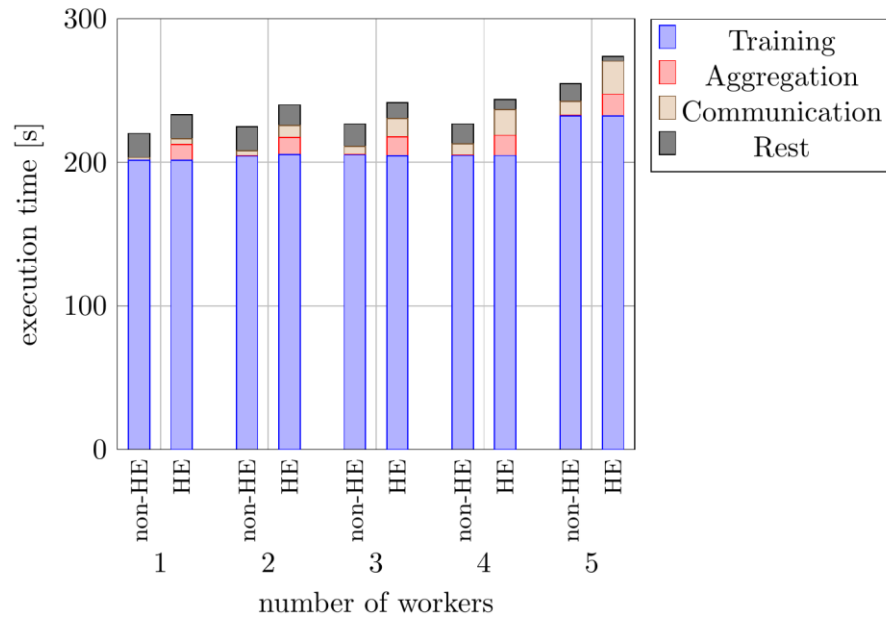


Figure 4: Performance of model training with and without HE for different numbers of workers

In summary, we show that it is possible to improve the privacy of FL massively without introducing a major performance loss. The parameters of the training process need to be chosen carefully, but it is certainly possible to arrive at good results. The only downside is higher code complexity. Our work proves that there is no technical reason to not employ FL to the most sensitive datasets and it opens the door for many new applications of ML. A publication of our findings and the corresponding source code is currently in progress.



### 3.2.2.2 Private Set Intersection

PSI is a special-purpose secure MPC. It allows two participants to compute the intersection of their data sets. Thereby, neither participant learns information from the protocol execution, except for the data entries in the intersection. For instance, PSI enables two companies to find out common customers privately - information that can subsequently be used for a joint advertising campaign. PSI is the most mature secure multi-party protocol, and computational overhead is small. Therefore, when parties engage in a PSI protocol, they do not have to expect significant performance issues.

We developed a first version of our PSI library within the H2020 project SafeDEED<sup>5</sup>. This version was using a Java PSI library, with the core of the cryptographic operations being executed by a component written in C++. While this previous implementation was sufficient for the purpose of building a small library, further integration work has shown that the performance and stability of the Java PSI library was lacking for enterprise-scale data sets. To combat these issues, a new version of the PSI functionality was developed from scratch within TRUSTS in collaboration with SafeDEED. The focus was put on performance and reliability. The new version was successfully integrated into the demonstrator of SafeDEED<sup>6</sup> and the UC Trials of TRUSTS (see corresponding report). In addition, the PSI library was added to the EUHubs4Data catalog<sup>7</sup>.

In contrast to the first library (v1), we build the second PSI library (v2) in Rust. Rust is a modern programming language with focus on performance and reliability. One of the main important features of Rust is its focus on memory safety, with its borrow checker component that ensures memory safety and removes large classes of common, often security-critical errors such as use-after-free errors and buffer overflow errors.

**Implementation details.** The previous library v1 used the PSI protocols developed for private mobile contact discovery (Kales et al., 2019). While these protocols can perform well for large set sizes (up to multiple million elements), their relative internal complexity also makes some aspects of the implementation more complex. Furthermore, if both datasets are relatively small (less than one million elements each), a simpler protocol (Jarecki and Liu, 2010), which is based on a variant of Diffie-Hellman key agreement, can perform nearly as well computationally while allowing for reduced communication overhead compared to our new protocols. In our implementation, we also apply some optimizations to the simpler protocol, namely the use of a cuckoo-filter with small false positive probability and cuckoo filter compression. We additionally protect the communication channel between the two parties using a TLS connection. For this, we use the rustls library, an implementation of TLS in the Rust programming language. Our implementation allows for both, self-signed certificates, as well as traditional public-key infrastructure. We use TLS version 1.3 per default.

---

<sup>5</sup> <https://safe-deed.eu/>

<sup>6</sup> <https://demo.safe-deed.eu/>

<sup>7</sup> <https://euhubs4data.eu/services/know-psittacus-privacy-enhancing-technology-for-data-sharing>





### 3.2.3 Further federated machine learning implementations for TRUSTS

#### 3.2.3.1 Ensemble Learning/Modelling

Ensemble methods use multiple learning algorithms to obtain better predictive performance compared to any of the constituent learning algorithms individually.

Following the assumption that the goal of any ML problem is to find a single model that best predicts our desired outcome, and since we can often not produce a model that is most accurate in all cases, ensemble methods take a myriad of models into account, and average these models to produce one final model. Thus the common approach to use ensemble learning is to train several models on the same dataset, and aggregate the results using one single ensemble model.

In addition to our other implementations, we have also followed this approach in collaboration with partners from UC2, the main idea is also related to FL. We have applied an ensemble model to aggregate distributed ML results for predicting/classifying the same problem, trained on different local datasets at servers of the involved parties.

This approach allows parties to collaborate with others in order to jointly solve a problem, without exposing their private data to each other and thus preserving the data privacy. Depending on the parties' datasets, and their description, whether they have the same feature set or different feature set, there is a UC where the parties should share their trained model between each other in order to retrain the ensemble model avoiding the need of sharing their data for that purpose. Only the final results of local evaluations are aggregated, the actual training data is not shared with others. We also want to point out that the security guarantees for methods based on data aggregation (ensemble learning, FL), are different compared to encryption methods.

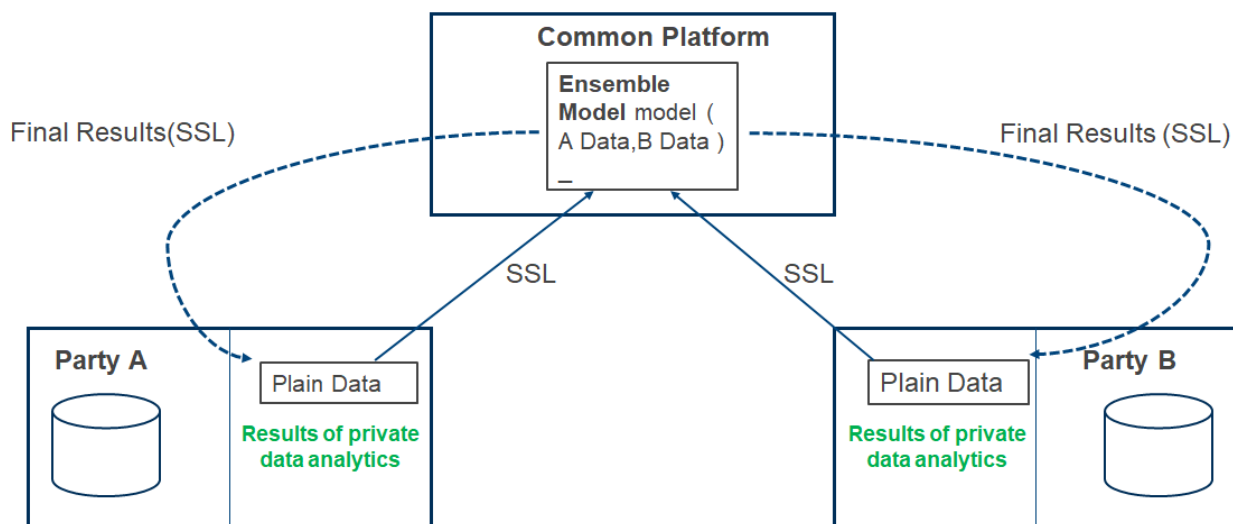


Figure 5: High level architecture illustrating how to collaborate using ensemble modeling

The collaboration between parties can be divided into two scenarios:

1. The first one is having parties collaborating over different data sets that have the same feature set.





Table 2 and Table 3 present various stages required to produce such collaboration.

Training:

Stage	Party A	Common	Party B
1- Training local models	Train model A on A's data		Train model B on B's data
2 – Creating Ensemble model		Create Ensemble model	
3 – Sharing models between parties			
4 – Running test sets	Running A's test set on Models A & B		Running B's test set on Models B & A
5 – Uploading the results + labels			
6 – Train Ensemble model		Train On A's test results and B's test results	

Table 2: Collaborating over different data sets with the same feature set– Training stage

Executing query:

Stage	Model A	Common	Model B
1- Query	Running prediction/classification		Running prediction/classification
2- sharing the results with the ensemble model			
3 – Running ensemble prediction/classification		Running query of the A's & B's results	

Table 3: Collaborating over different data sets with the same feature set – Execution stage

We have run an experiment to illustrate and prove the Ensemble modeling concept in parties' collaboration. In this experiment we used a data set that contains monitored data of a hardware system. There are more than 120 monitored parameters (features), thousands of records and a single label defining if the system is performing correctly or having a fault.

The data set of this experiment is Dell's confidential data, and it is not part of TRUSTS data sets. The first experiment illustrates the scenario where parties that have the same feature set but different data, want to collaborate in order to improve their classification whether the system is performing correctly or having a fault.

We use two different datasets with the same feature set, and follow the steps as described above for collaboration, while using a random forest (RF) classification model as the model algorithm for each part. The main goal was to illustrate that the precision, recall and the receiver operating characteristic curve (ROC) are improved from this collaboration.



Each graph in Figure 6 shows the precision, recall, and the ROC of each model based on a single dataset.

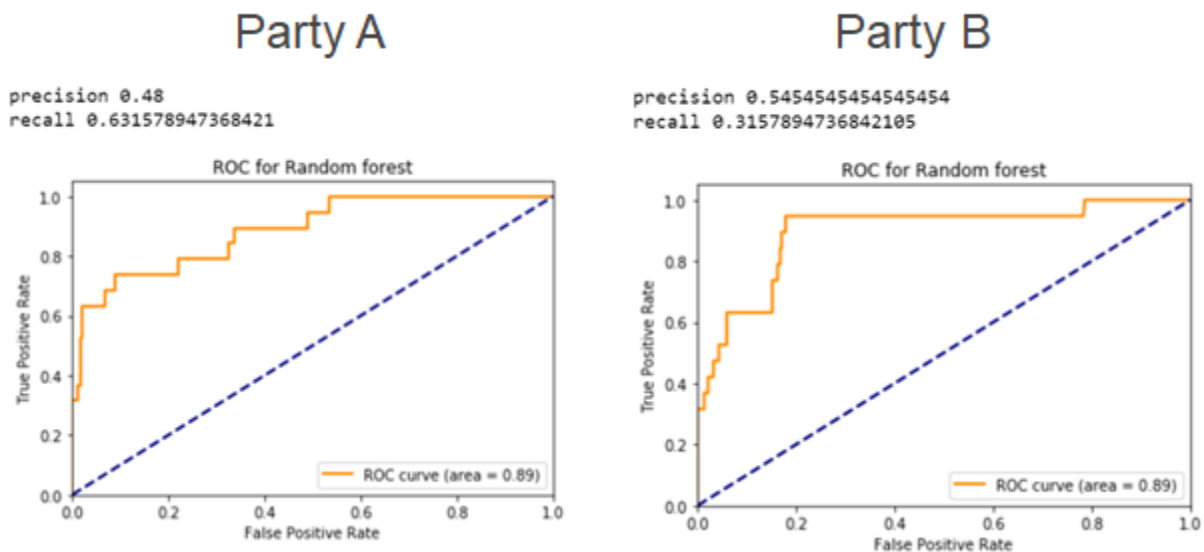


Figure 6: Experiment results of Ensemble modeling on different data sets having the same feature set - results per party

Additionally, the graph in Figure 7 shows the precision, recall, and the ROC of the ensemble model.

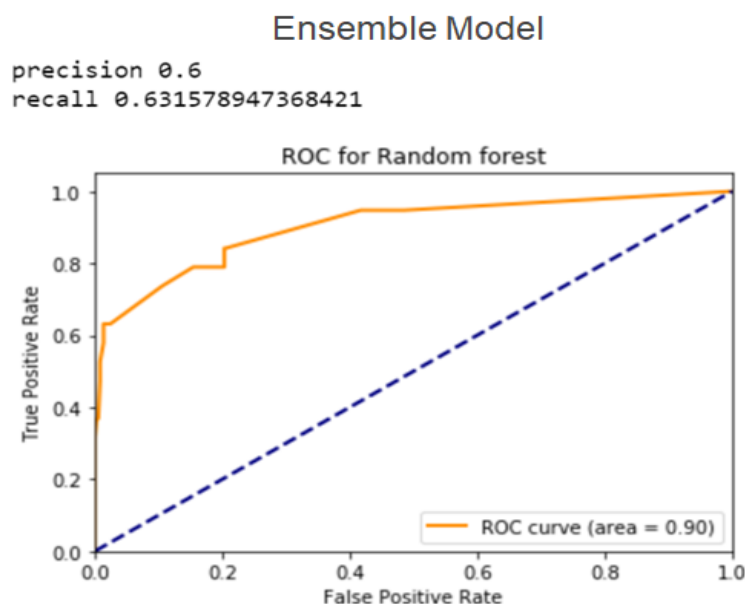


Figure 7: Experiment results of Ensemble modeling on different data sets having the same feature set – Ensemble results

As the figures suggest, the ensemble results are improved in all tested sections, namely precision, recall, and ROC.

2. The second one is having parties collaborating over different data sets that have different feature sets.

Table 4 and Table 5 present the various stages required to produce such collaboration.

Training:

Stage	Party A	Common	Party B
1- Training local models on different correlated data (common ID)	Train model A on A's data		Train model B on B's data
2 – Creating Ensemble model		Create Ensemble model	
3 – Decide on common test set			
4 – Running test sets	Running A's test set on Model A correlated to B		Running B's test set on Model B correlated to A
5 – Uploading the results + labels		Merge data on same ID's	
6 – Train Ensemble model		Train On merged test results	

Table 4: Collaborating over different data sets with different feature set – Training stage

Executing query:

Stage	Party A	Common	Party B
1- Query on different data and <b>common ID</b>	Running prediction/classification		Running prediction/classification
2- sharing the results with the ensemble model		Merge the results using common ID	
3 – Running ensemble prediction/classification		Running query of the merged results	

Table 5: Collaborating over different data sets with different feature set – Execution stage

The second experiment we have run, illustrates the scenario where parties that have different feature sets and different data referring to the same label, wants to collaborate in order to improve their classification whether the system is performing correctly or having a fault. Each dataset has different features that hold different monitored parameters of the same hardware system. In that case the label that describes faults is the same for all datasets. We follow the steps as described above for collaboration, while using a RF classification model as the model algorithm that each of the parties was using. The main goal was to illustrate that the precision, recall and the ROC are improved from this collaboration.



Each graph in Figure 8 shows the precision, recall and the ROC of each model based on a single dataset.

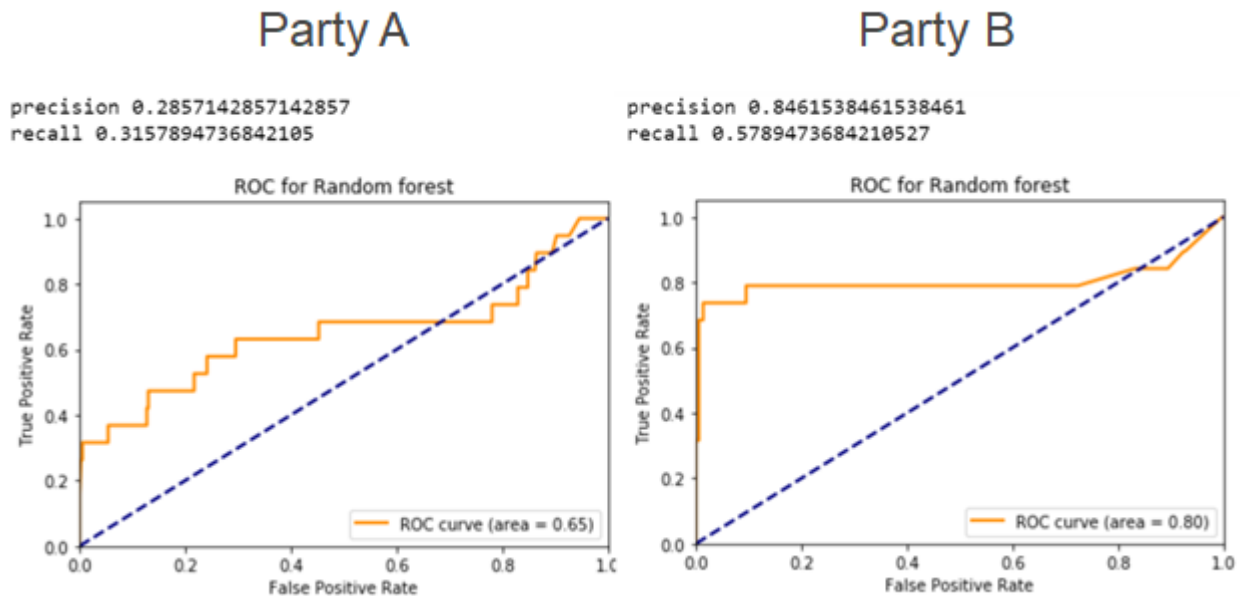


Figure 8: Experiment results of Ensemble modeling on different data sets having different feature set - results per party

And the graph in Figure 9 shows the precision, recall, and the ROC of the ensemble model.

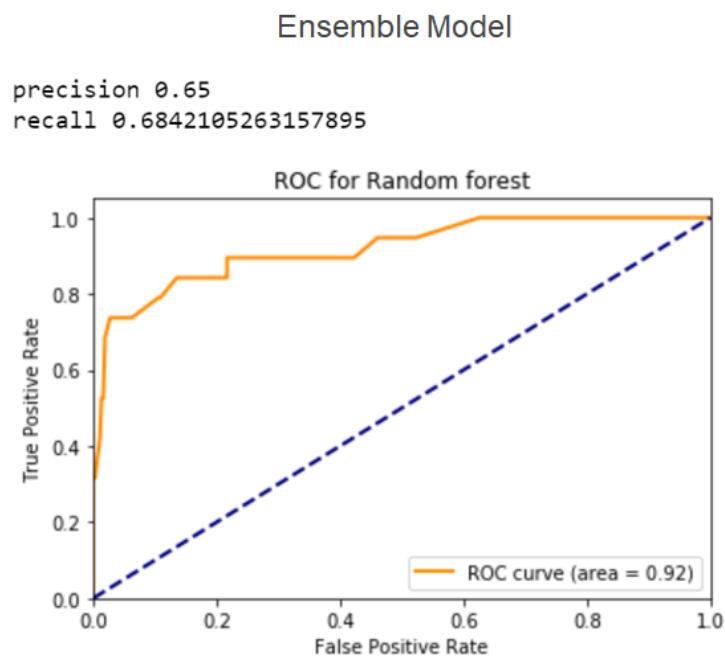


Figure 9: Experiment results of Ensemble modeling on different data sets having different feature set -Ensemble

As we can see, the ensemble results are improved in all tested sections: precision, recall, and ROC.

### 3.2.3.2 Vertical Federated Learning using SHAP values

SHAP values - SHAP values interpret the impact of having a certain value for a given feature in comparison to the prediction we would make if that feature took some baseline value.

The suggested invention provides a capability to run classification ML algorithms over more than one data set belonging to different and, at times, rival parties. Training is performed without sharing any of the raw data between the various parties, and the final model provides one single prediction while keeping data privacy and security.

The way to withhold these constraints is by running federated ML models, over each of the data sets separately, and then share only the SHAP values generated by each of the models. The SHAP values from all the federated ML are used as input to a new classification ML algorithm, which provides a single prediction based only on it. The following figures (10, 11) illustrates the solution at a high level:

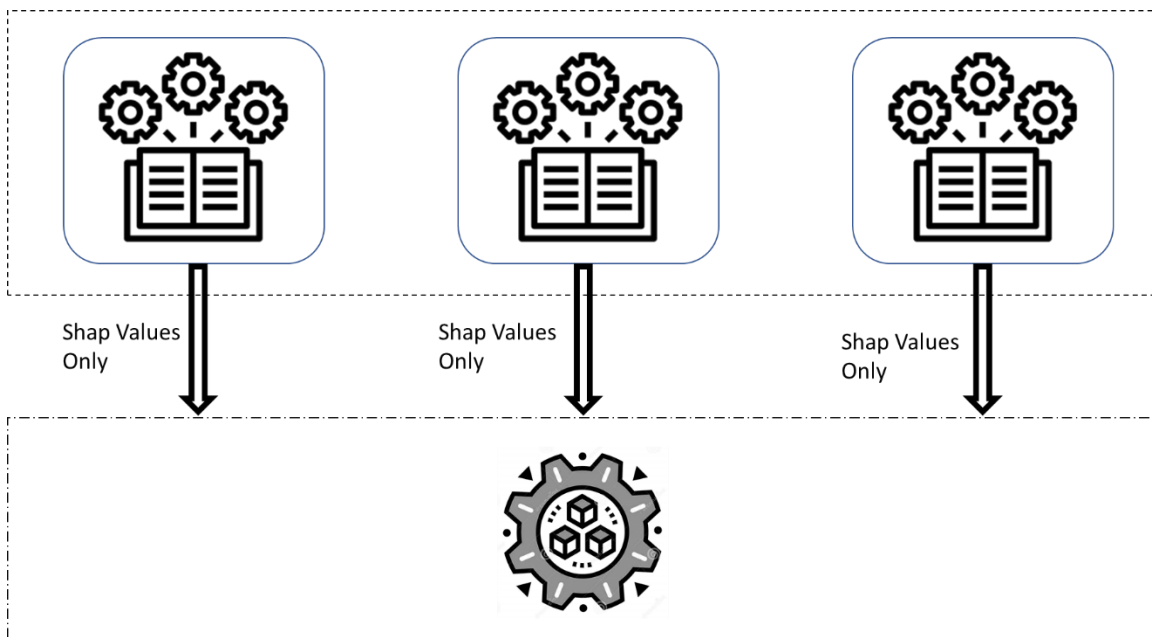


Figure 10: The way to collaborate with SHAP values

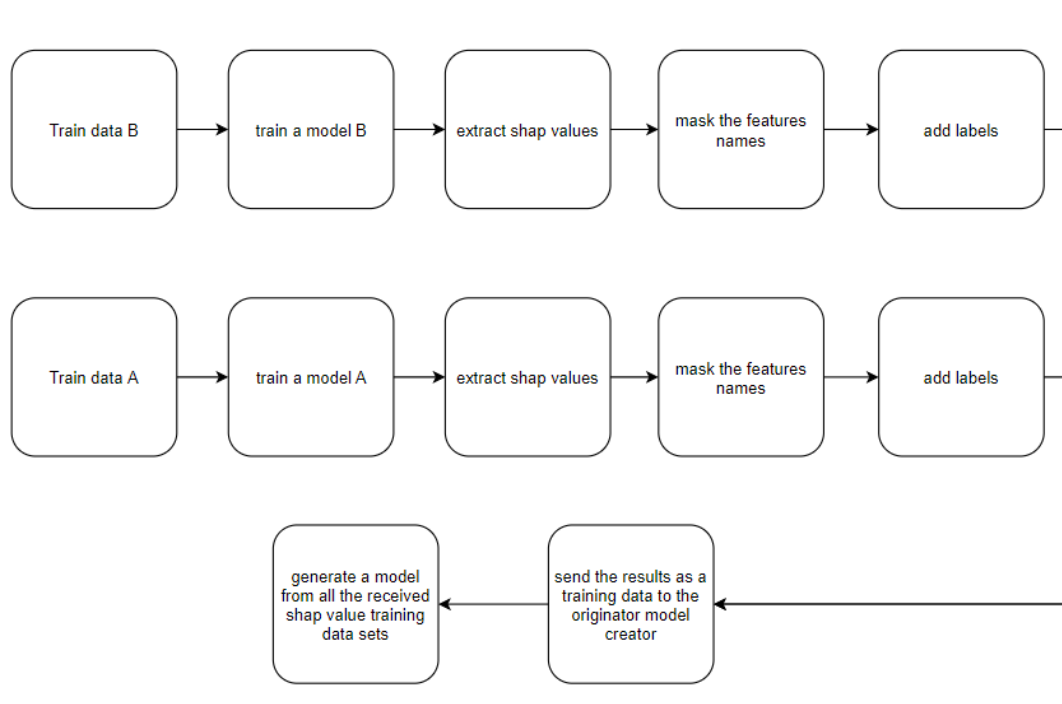


Figure 11: SHAP values collaboration – data flow chart

There are two main assumptions at the base of the invention. These are realized in many real-life scenarios:

1. The labels (which is the “ground truth” for supervised training models) are shared across the federated models using PSI or other secured protocol.
2. A unique key for joining the output from the separate datasets and models is shared between the parties.

In general, the solution consists of two main components as shown in figure 12:

1. Federated classification ML algorithm.  
For each of the data sets, a unique classification algorithm is executed on the raw data. SHAP values’ results from this model become the input data set for the second component. To protect the data, the feature's names are masked by hashing their name.
2. Common classification ML algorithm.  
This component is responsible for running a ML classification algorithm over the unified SHAP values results, coming from all of the federated algorithms.



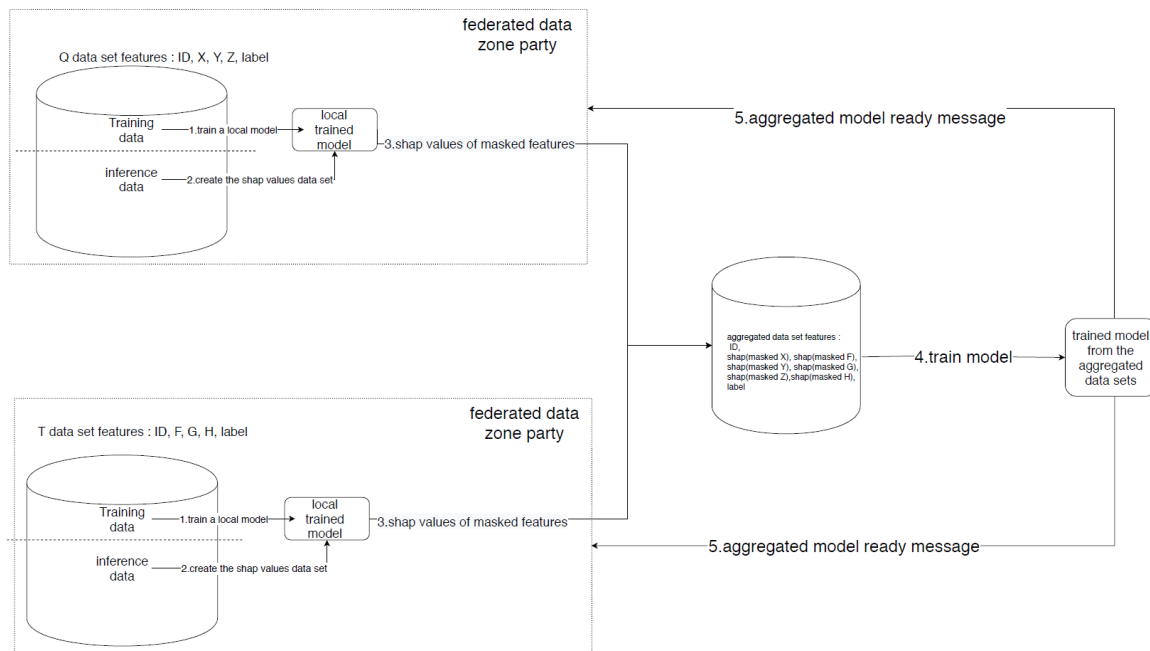


Figure 12: SHAP values collaboration – flow chart

### Illustrative example:

In this experiment we used a data set that contains monitored data of a hardware system. There are more than 120 monitored parameters (features), thousands of records and a single label defining if the system is performing correctly or having a fault.

Each data set was split into 3.

1. Training data for the federated models.
2. Test data for each model, whose prediction SHAP values are used as input for training the common model.
3. Second test data, that is used to test the common model.

A RF model was trained on each data set and was tested twice with different data. First test was used to get the SHAP values as an input to the common model. Second test was done to compare the results between each model to the common model.

The first federated model results are (trained on the original features from dataset 1) depicted by Figure 13.

```
precision 0.2857142857142857
recall 0.3157894736842105
```

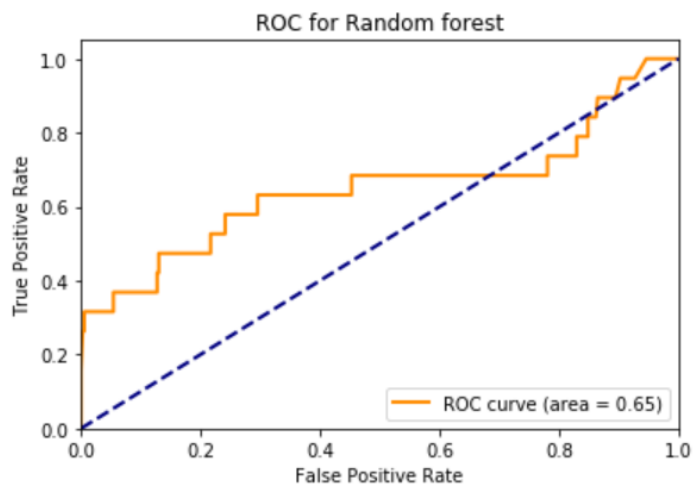


Figure 13: Illustrative example - first federated model results

The second federated model results are (trained on the original features from dataset 2) depicted by Figure 14.

```
precision 0.8461538461538461
recall 0.5789473684210527
```

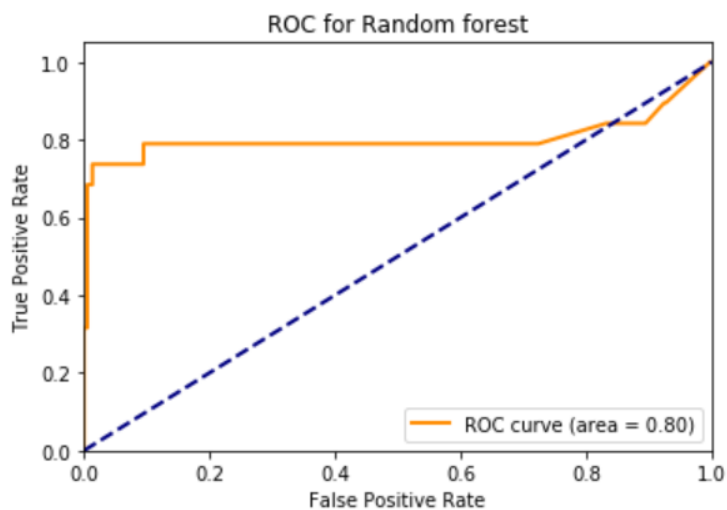


Figure 14: Illustrative example - second federated model results



While the common model (trained on SHAP values from federated models) has its results depicted by Figure 15.

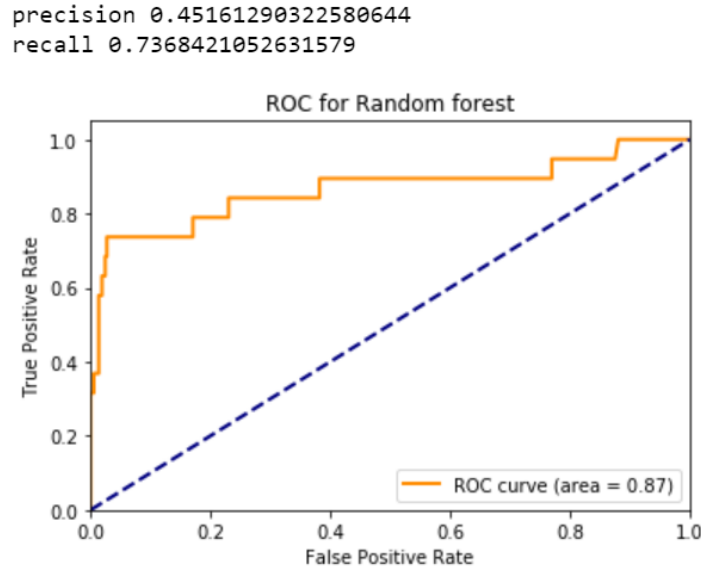


Figure 15: Illustrative example - Common model results

We conclude that for this UC an improvement of the results of the common model trained on SHAP values compared to each of the federated model trained on the partial raw data. ROC is significantly higher for the common model. While the precision of the common model is between the precision of the federated models, the recall is much higher than both of them.

### 3.2.3.3 Superseded federated learning

All of the methods mentioned above require full collaboration of the involved parties from the training phase up to the inference one. This makes the collaboration too complex, and sometimes this complexity even prevents collaborations.

In this method we want to suggest a way to perform VFL while reducing the complexity of it by limiting the collaboration only to the training phase.

For having this we should use a generative adversarial network (GAN). Given a training set, this technique learns to generate new data with the same statistics as the training set. It is a ML model in which two neural networks compete with each to generate new, synthetic instances of data that can pass as real data - to become more accurate in their predictions.

GANs typically run unsupervised and use a cooperative zero-sum game framework to learn.

The stages to perform superseded FL are:

#### The pretraining stage:



The pretraining and training stages are almost like the VFL while superseded FL add another stage in the middle as described in the flowchart below:

First the parties use PSI to identify the common sample IDs across their datasets securely and privately (as per figure 16 below).

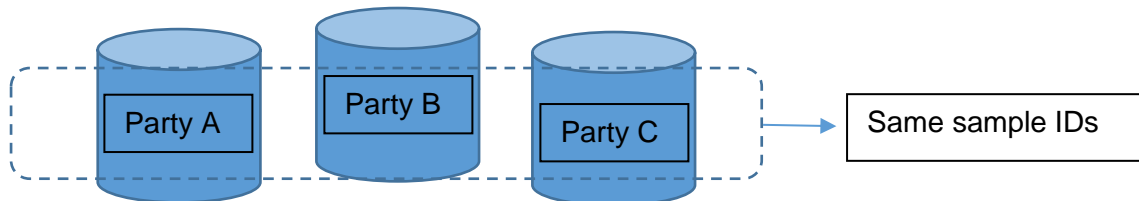


Figure 16: common sample IDs cross datasets

Once they find out there is a logical sense to collaborate, each party creates a local model relevant to the collaboration purpose.

For each model, each party creates a GAN that simulates its data depending on the inputs.

A VFL model is created for all parties.

Each party shares its own model and a GAN with all involved parties.

### **The inference stage:**

The second stage, the inference, is almost like HVL while superseded FL adds a stage prior the inference as described below:

When a party has a new record and wants to perform inference, it uses all party's GAN models (shared during training) to generate a complete data set.

The new data record plus the GAN models are used as inputs for all the models of the involved parties.

The results of each model are used as input into the VFL model to perform an inference.

Since the non overlapped data is simulated, the accuracy of the inference will increase in correlation to the number of the overlapped features (as per figure 17 below).



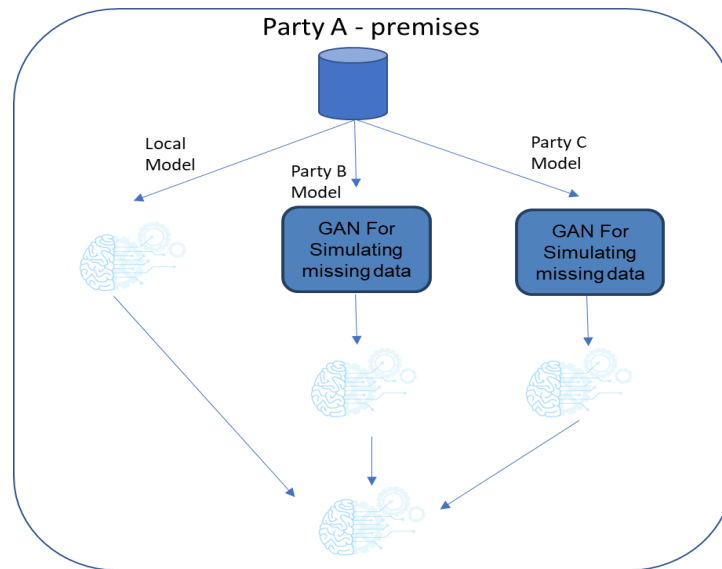


Figure 17: Superseded Federated Learning flow

**Experiment:**

A given dataset contains 100 columns and thousands of records.

We split it into a train set and a test set and used the RF classification model as the experiment model.

The test set was cloned 2 more times.

In the first clone we changed 7 different column's values to contain an approximate value – this is to illustrate the option of doing classification when we've got missing data.

In the second clone we used GAN in order to fill the missing data (the same 7 columns as in the first clone) with simulated data (which is almost similar to the actual data).

The results for the classification with missing data are, as per figure 18.



```
precision 0.10714285714285714
recall 0.7894736842105263
```

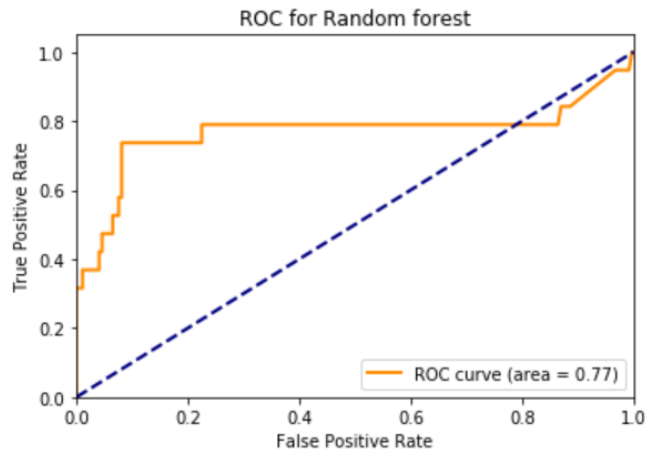


Figure 18: Experiment Illustration – classification with missing data

The results of the classification using GAN values are shown in figure 19.

```
precision 0.8
recall 0.42105263157894735
```

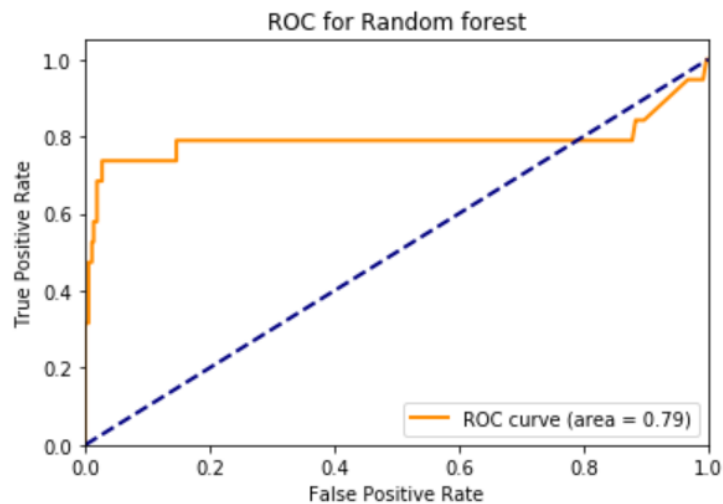


Figure 19: Experiment Illustration – classification with GAN data

As we can see, the ROC was upgraded from 0.77 to 0.79 and the precision was increased from 0.1 to 0.8.

### 3.3 Application

As already mentioned, PSI protocols are considered one of the most mature PETs that involve encryption. There exists a public PSI component (<https://github.com/Safe-DEED/PSI>) developed



in the scope of an ongoing EU-Horizon project called Safe-DEED. It can be tested through a demonstrator made publically available by Safe-DEED (<https://safe-deed.eu/>). This PSI component consists of a Java PSI library. For the core functionality, it relies on a state-of-the-art PSI C++ protocol. The protocol is secure against a malicious client - one who can deviate from the protocol - and a semi-honest server. For encryption, the protocol uses symmetric encryption schemes specially designed for PSI. This allows a significantly better throughput. More specifically, it can process thousands of customers in less than ten seconds.

The Safe-DEED PSI component is intended to be used for joint data usage between different enterprises in different domains. The Safe-DEED's PSI component only reports back the intersection (identifiers) of two (customer relation) databases. Since UC2 requires not only to find the common identifiers but also sending additional data corresponding to these identifiers, we had to adapt the protocol. It now fits the needs of UC2 and was already integrated into the UC2 workflow. The plan is to test it in the corporate environment soon.



## 4 Anonymisation and De-Anonymisation

### 4.1 Introduction

This chapter starts with a brief description of the basics of de-anonymization elaborated in D4.1 which were built upon the results of SafeDEED. Following the structure of D4.1 the new contributions are discussed in the individual subchapters and compared with the previous results.

#### 4.1.1 Task background and motivation

Individuals can be easily identified with their name, address, and social security number. This is their so-called personally identifiable information (PII). But even without any PII, it is possible to uniquely identify individuals in data sets.

Sweeney (2000a) for instance showed that 87% of the US population can be uniquely identified using only gender, date of birth and ZIP code. Figure 20 demonstrates how this information can be used to gather sensitive information. No one can be identified with one of these attributes alone, but it is possible when they are combined. These indirect attributes for identifying individuals are called quasi-identifier (QID). The task to identify individuals without their PIIs is called de-anonymisation. Unlike PIIs, the dangers of QIDs cannot be assessed so easily. For this reason, privacy models are applied here.

The challenge in working with privacy models is that they come with a cost on the dataset's utility. If too much information is removed, then the dataset becomes less useful. The more processing steps, the more it becomes distorted. Models are needed to determine the privacy-utility trade-off. So far only a limited number of corresponding models have been introduced (Li et al., 2009; Hsu et al., 2014).



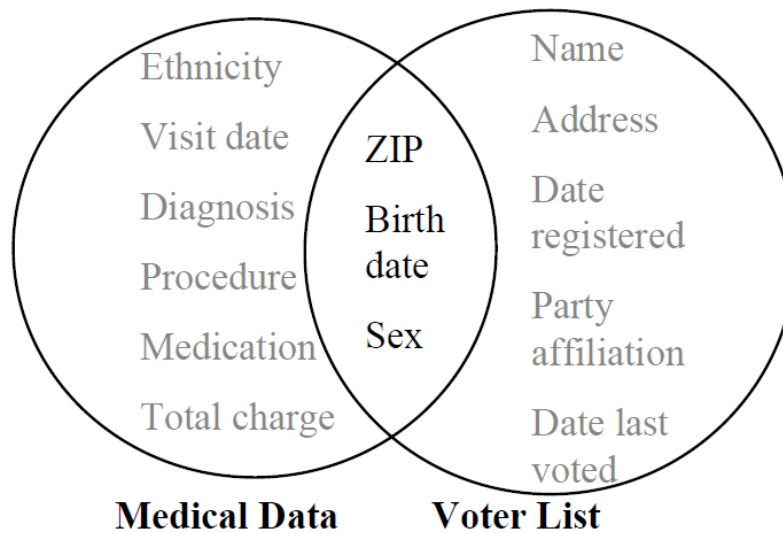


Figure 20: The intersection of two semi-publicly available datasets (Sweeney, 2000)

In TRUSTS we use k-anonymity (Sweeney, 2002a) and its extension l-diversity (Machanavajjhala et al., 2007) as privacy models. The goal of k-anonymity is to group together QIDs and have groups which are greater than k. With l-diversity as a further development, it is not enough to look at the QIDs, it is also extended to the sensitive values. This means that there must be different sensitive values greater than l. Figure 21 illustrates this and compares the results of k-anonymity with k=4 and l-diversity with l=3.



Original					4-anonymous				
	Non-Sensitive			Sensitive		Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition		Zip Code	Age	Nationality	Condition
1	13053	28	Russian	Heart Disease	1	130**	< 30	*	Heart Disease
2	13068	29	American	Heart Disease	2	130**	< 30	*	Heart Disease
3	13068	21	Japanese	Viral Infection	3	130**	< 30	*	Viral Infection
4	13053	23	American	Viral Infection	4	130**	< 30	*	Viral Infection
5	14853	50	Indian	Cancer	5	1485*	≥ 40	*	Cancer
6	14853	55	Russian	Heart Disease	6	1485*	≥ 40	*	Heart Disease
7	14850	47	American	Viral Infection	7	1485*	≥ 40	*	Viral Infection
8	14850	49	American	Viral Infection	8	1485*	≥ 40	*	Viral Infection
9	13053	31	American	Cancer	9	130**	3*	*	Cancer
10	13053	37	Indian	Cancer	10	130**	3*	*	Cancer
11	13068	36	Japanese	Cancer	11	130**	3*	*	Cancer
12	13068	35	American	Cancer	12	130**	3*	*	Cancer

3-diverse				
	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	1305*	≤ 40	*	Heart Disease
4	1305*	≤ 40	*	Viral Infection
9	1305*	≤ 40	*	Cancer
10	1305*	≤ 40	*	Cancer
5	1485*	> 40	*	Cancer
6	1485*	> 40	*	Heart Disease
7	1485*	> 40	*	Viral Infection
8	1485*	> 40	*	Viral Infection
2	1306*	≤ 40	*	Heart Disease
3	1306*	≤ 40	*	Viral Infection
11	1306*	≤ 40	*	Cancer
12	1306*	≤ 40	*	Cancer

Figure 21: Illustration of k-anonymity and l-diversity (Machanavajjhala et. al, 2007)

There are no clear statements on the levels of anonymity in the GDPR<sup>8</sup> when working with anonymized data. Only the following statement in Recital 26: “The principles of data protection should therefore not apply to anonymous information, ... data rendered anonymous in such a manner that the data subject is not or no longer identifiable”. It is therefore only addressed that the principles are not applied to anonymous data. However, it does not specify when data is anonymous.

In D4.1 the following 3 requirements for data stewards were formulated in order to show the importance of de-anonymisation risk analysis: (1) Raising awareness that datasets are not anonymous when the PII's are removed. Even when no specific privacy models are recommended a de-anonymisation risk analysis helps to gain (2) GDPR compliance by showing the data controller the risks in their datasets and helps to (3) weigh up the anonymisation measures and their scope.

<sup>8</sup> <https://eur-lex.europa.eu/eli/reg/2016/679/oj>

As described in D4.1, the literature review revealed that there are only two de-anonymisation risk analysis tools available so far: ARX (Prasser et. al, 2020) and X2R2 (Hagedoorn et. al, 2020). However, these are currently limited to tabular data and are therefore not sufficient for complex data types such as those to be supported in TRUSTS. Furthermore, there are also limitations with visualisation of de-anonymisation risks.





#### 4.1.2 Contribution

From M6-M18, as mentioned in D4.1, Research Studios Austria implemented under Task 4.3 “Anonymisation and de-anonymisation”, six risk analysis modules. As already discussed in chapter 4.1.1, there are already open source tools available for risk analysis, but only for a specific type of datasets (tabular data) and with limited visualisation options. The modules developed addressed these challenges and are supporting different types of data sets including corresponding visualisations.

In addition, the importance of risk analysis when dealing with sensitive data sets and how this can be identified was pointed out. This served as a starting point for the progress from M19-M32. Built on this, a literature review was conducted to determine which anonymisation methods are suitable for the different types of datasets built on k-anonymity and l-diversity specifications.

FORTH continued to develop the application while Research Studios Austria took care of the identification and development of suitable anonymisation methods. The existing application was expanded in order to be able to apply these anonymisation methods. Appropriate algorithms for all data types have been identified and will be implemented by the end of the project. The anonymisation process for tabular data based on generalisation hierarchies has already been integrated into the platform and can already be used (see Section 4.3).

Our contributions were the expansion of the existing application with appropriate anonymisation methods. This process was divided into the following two aspects:

- The risk analysis should serve as input for the anonymisation process. With the goal to ensure that only the most necessary changes are applied to the data, thus minimising the trade-off between privacy and utility. During the literature research, it was taken into account that the proposed methods are suitable for an interaction with privacy models like k-anonymity and l-diversity.
- A characteristic of the already implemented method for de-anonymisation risk analysis is the support of complex and multidimensional data. This was the identified limitation of the existing state-of-the-art tools. Therefore, the second aspect was that the anonymisation process must support these different types of data sets. This has already been partially implemented and the process is described in detail in section 4.3.

The TRUSTS project builds on existing parts of the SafeDEED project. In D4.1, the differences of the two projects have been pointed out. In summary, the solutions of SafeDEED were taken as a starting point and were then extended. The code was generalised and new modules were added.



The solutions developed in SafeDEED were adopted and redesigned to support complex and multidimensional types of datasets. New modules were developed for TRUSTS, such as the algorithm for spatiotemporal data and textual data. All these algorithms and methods were integrated into one application. The application, which was developed by FORTH, serves on the one hand as a data management platform and on the other hand as front end for the risk analysis methods developed by Research Studios Austria. The aim is to integrate the application as part of the TRUSTS platform.

The contributions for M19-M32 described here in D4.2 are the identification of algorithms and methods suitable for anonymisation as a supplement for the risk analysis. The objective of the literature review was to find algorithms/methods that:

- are in accordance to the TRUSTS objectives
- can be integrated in the existing TRUSTS environment
- are suitable for the data formats supported by risk analysis
- focus on improved trade-off of efficiency vs. information loss

Developments from SafeDEED were also applied here: Adapted from Bampoulidis et al. (2019), the generalisation with hierarchies is provided as a method for anonymisation. This approach is suitable for tabular data and was originally implemented as a standalone Java application. The described features from the paper were reimplemented as Python modules and integrated into the existing application. For this purpose, the Python module was containerised and added to the existing multi-container application.

As a result of the literature review, anonymisation concepts based on the risk analysis methods were identified. This includes solutions for all complex and multidimensional data types and is already processed as prototypes. Slijepčević et al. (2021) are describing in their paper four potential strategies for anonymisation of QIDs: (1) generalisation, where data is anonymised based on pre-defined hierarchies, (2) suppression which is the full removal of the data points, (3) bucketization (Xiao & Tao, 2006) where data is grouped into equal sized buckets and (4) microaggregation (Domingo-Ferrer & Mateo-Sanz, 2002) where data is grouped in combination with aggregation of the values.

Generalisation and suppression as anonymisation techniques offer a high level of protection for private data (Kabir et al., 2011). However, anonymisation always has a trade-off and working with generalisation hierarchies can also lead to a high loss of information (Bampoulidis et al., 2019; Fung et al. 2005; Iyengar, 2002; LeFevre, 2005; Sweeney, 2002b). For this reason, it is useful if several options are offered in order to be able to choose the best variant for the respective application.

Full suppression will be provided for each type of dataset but should only be used as a last resort since any information from a column will be lost which means a high cost for the utility of the dataset. For textual data, a method will be offered that is not discussed in the paper by Slijepčević (2019), but which can be understood as a targeted use of suppression. In this case,



the named entities in the texts are identified within the framework of natural language processing (NLP) with the help of a pretrained AI and selectively removed from the data set. The following concepts have been selected for the respective data sets:

- Aggregated data (Hierarchies and Microaggregation)
- Tabular data (Hierarchies)
- Invoice data (Hierarchies, Bucketization and Clustering)
- Textual data (Named entity recognition and sentiment analysis)
- Location data (Clustering)

## **4.2 De-Anonymisation Risk Analysis and Anonymisation Modules**

In this section, the de-anonymisation risk analysis module and the appropriate anonymisation methods are described. These methods have been identified through a literature review and have already been evaluated using prototypes. The structure of this section is based on the different types of datasets considered. The risk analysis method is described along with the appropriate anonymisation method. Generalisation with hierarchies has been implemented and the workflow is described in section 4.3.

### **4.2.1 Privacy models and anonymisation strategies**

The de-anonymisation risk analysis module is based on the two privacy models k-anonymity and l-diversity (see section 4.1.1 for a description of these two models). Figure 22 is a screenshot of the application and shows the result of a risk analysis. Here you can see that the higher the number of QIDs, the higher the probability of de-anonymisation. If it is still ~50% with 2 QIs, it rises to ~80% with 3 QIs and to 88% with 11 QIs.



## Contracts

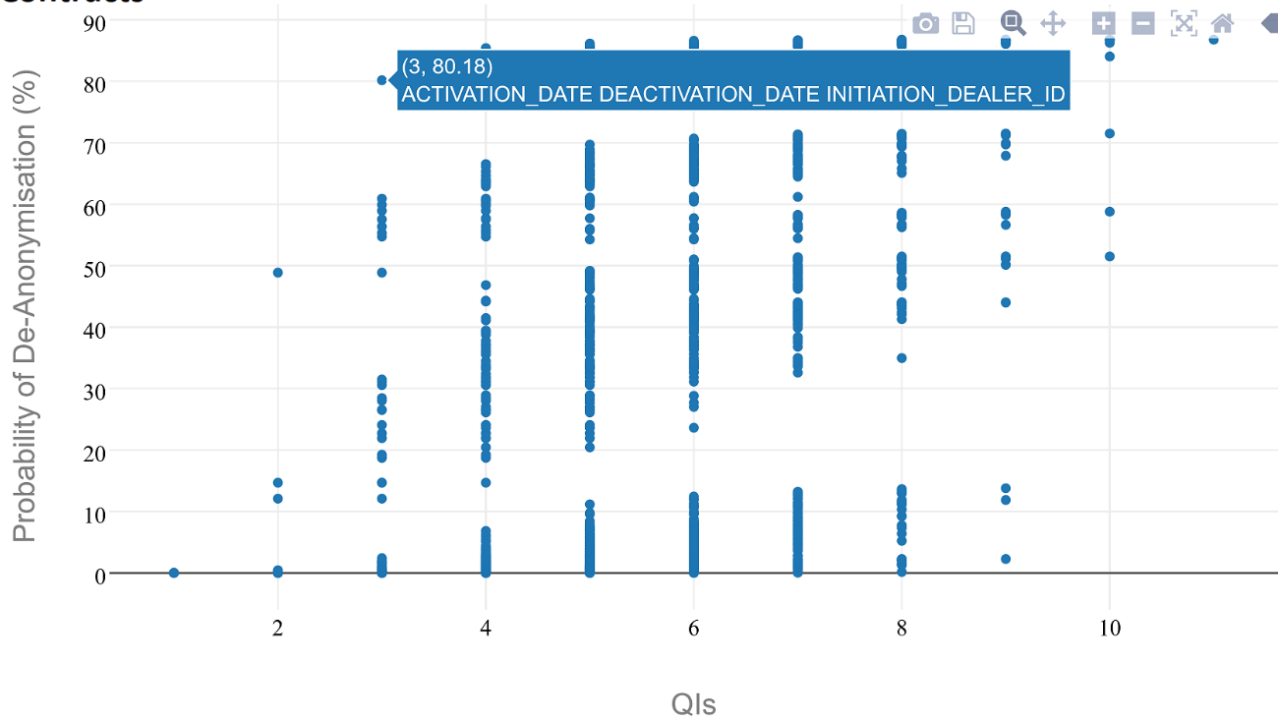


Figure 22: Screenshot of the k-anonymity risk analysis module on a contracts datasets for k=2

The risk analysis module builds on the results of the SafeDEED project (Bampoulidis, 2020a). The limitation in this first version was that k could only take the value 2. This was extended within the scope of this task and k can now have any value. Furthermore, visualisations of the results were added. These visualisation are interactive and were implemented with the two Javascript libraries plotly<sup>9</sup> and leaflet<sup>10</sup>. Figure 22 is a screenshot from D4.1. The descriptions from this earlier version of the application were not self-explanatory. Therefore, they have been adapted and improved. Figure 23 shows the more descriptive plot tooltips and legends.

<sup>9</sup> <https://plotly.com/javascript/>

<sup>10</sup> <https://leafletjs.com/>

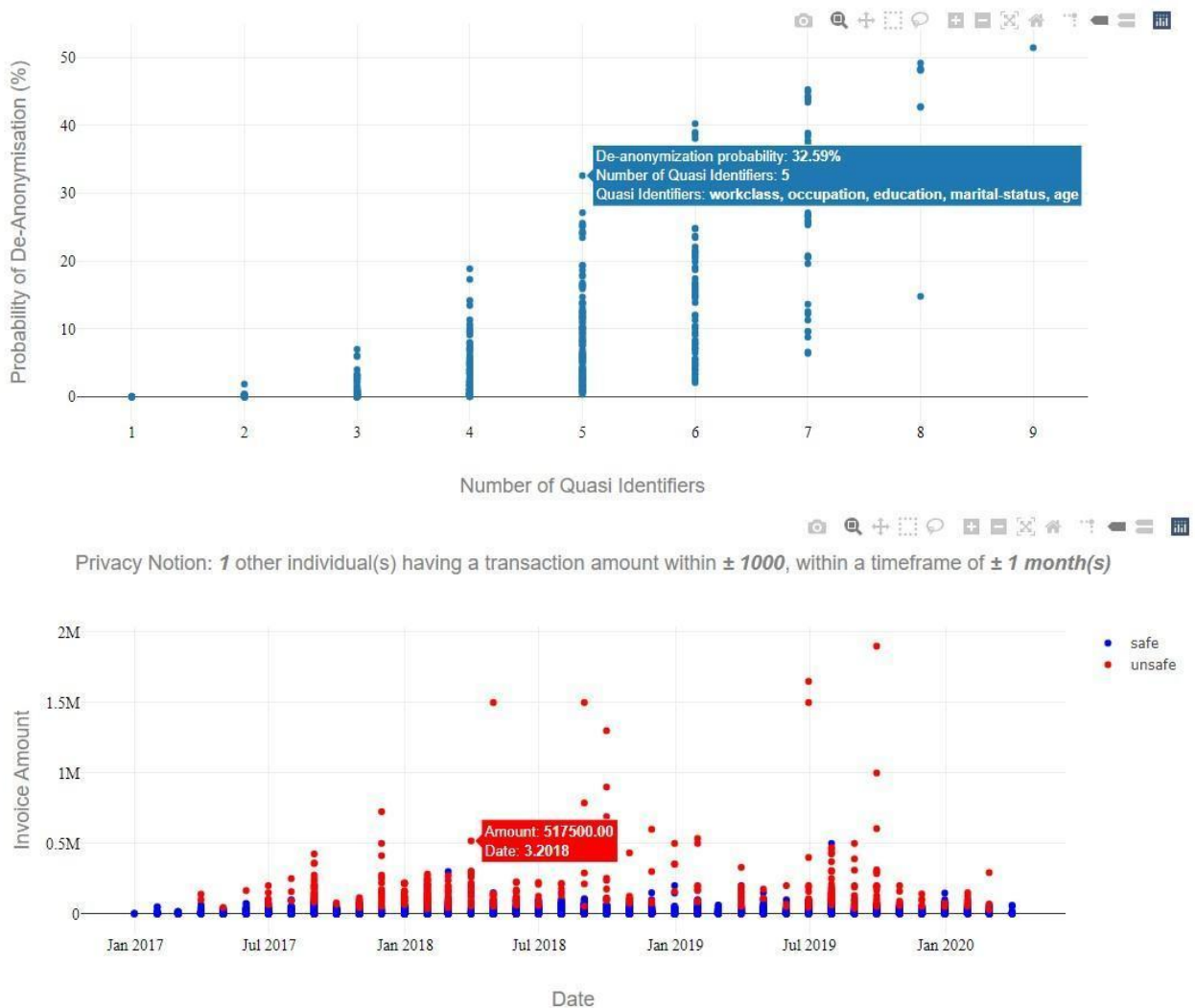


Figure 23: Examples of enhanced visualisations with more descriptive tooltips and legends

In addition to the check for k-Anonymity, tabular data is also checked for l-Diversity compliance. This privacy model builds on k-Anonymity and expands it to include the dimension of sensitive attributes. This enables a more precise analysis and reduces weaknesses of k-anonymity. A detailed description can be found in section 4.1.1. Figure 24 is a screenshot of the application and demonstrates the visualisation of the results of an analysis of the risk analysis module of  $l=2$ .

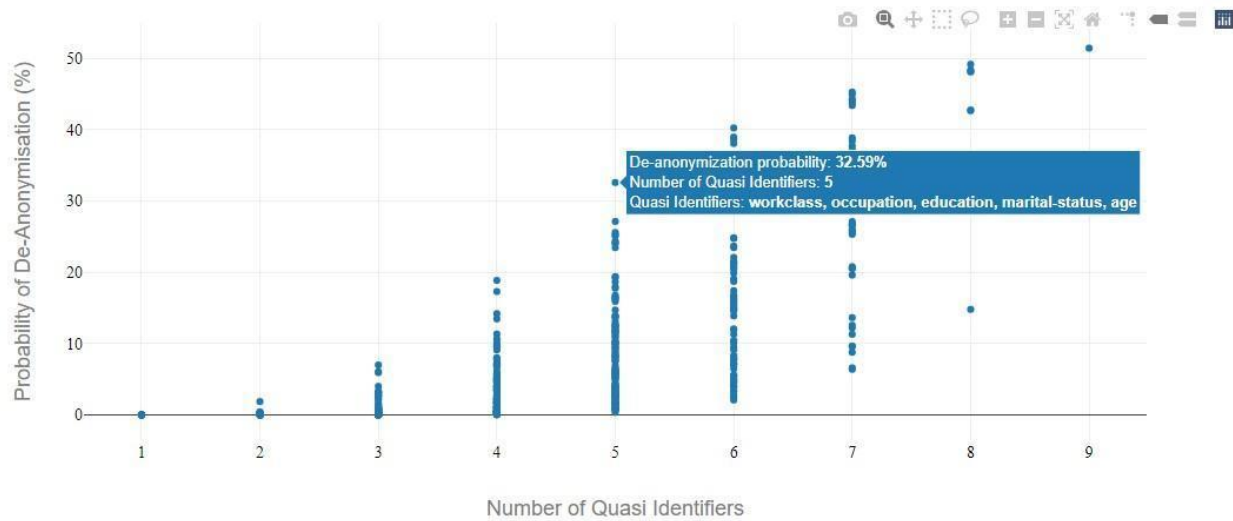


Figure 24: Screenshot of the l-diversity risk analysis module for l=2 with enhanced UI

#### 4.2.2 Tabular Data

In addition to complex and multidimensional types of data sets (see the following sections), datasets without special columns, such as time, location and aggregated values, so-called tabular data, are also supported. In the application, both k-Anonymity (see Figure 22) and l-Diversity (see Figure 24) are provided for the de-anonymisation risk analysis.

As a newly added extension, the current version of the application also offers an option for anonymisation of tabular data. This is again based on a result of the SafeDEEDs project. In their paper Bampoulidis et al (2019) PrioPrivacy, a tool developed for local recoding of tabular data is presented. They are also describing the concept of anonymisation with hierarchies (see Figure 25). The tool was programmed as a standalone solution in Java. For the TRUSTS application, concepts were taken from the paper and transferred into Python code.

QIs	Generalisation Levels			
	0 (Original Values)	1	2	3
gender	F or M	*		
Date of Birth	dd/MM/yyyy	MM/yyyy	yyyy	*
ZIP	5 digits	first 4 digits	first 3 digits	*

Figure 25: Example hierarchies from Bampoulidis et al (2019)

#### 4.2.3 Location Data

This section discusses location data, especially spatiotemporal data which contains information of individuals from two dimensions: the (1) geographical level, i.e. the location, and the (2) time level. In the first phase of the project the possibilities for risk analysis were identified and implemented. Since D4.1 the user interface (UI) has been revised. The visualisations were

improved with more descriptive plot tooltips and legends. Figure 26 shows the enhanced application's output of such an analysis. The output is a map in which all points are displayed and colour coded. The colour scale goes from green to red where green means safe and red unsafe. With this information, the data controller gets an overview of the risk and can then weigh up whether anonymisation of the data is necessary.

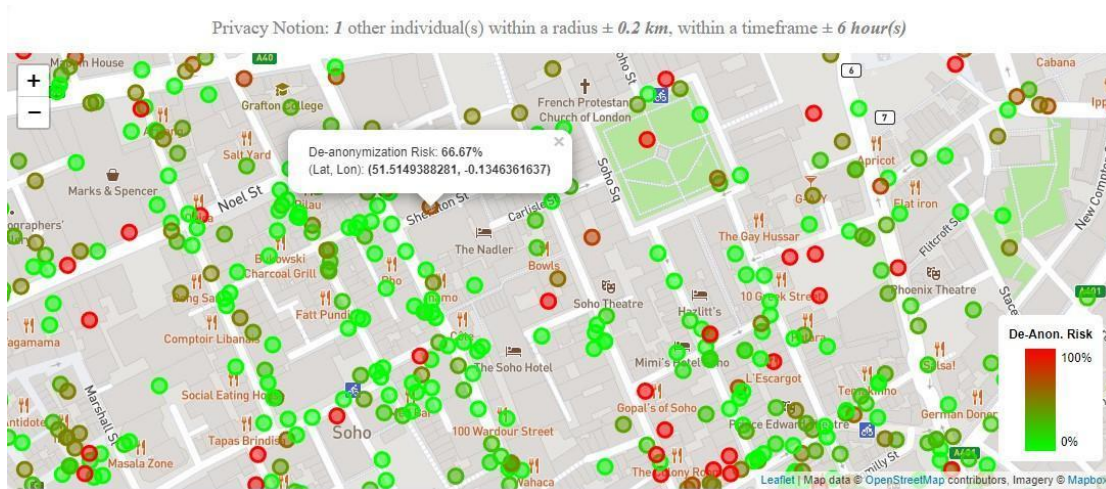


Figure 26: Screenshot of the enhanced output for spatiotemporal data risk analysis (gowalla dataset)

In the finished application, this data can be anonymised using clustering. The number of clusters can be chosen based on the desired level of  $k$  and several clustering algorithms are then applied. Figure 27 demonstrates output taken from the prototype.

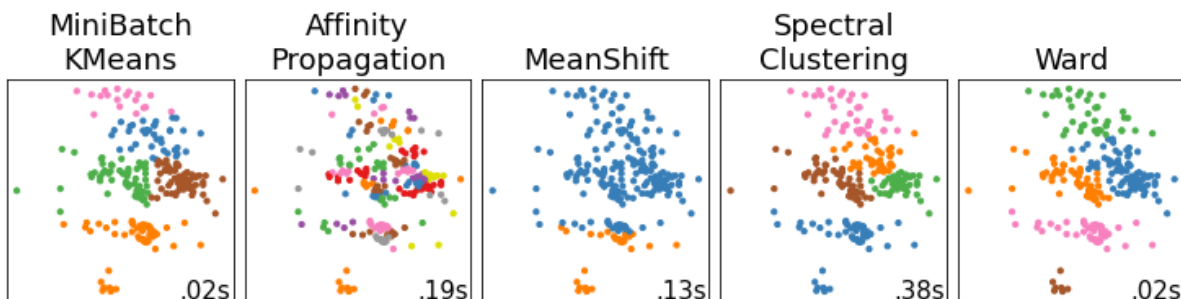


Figure 27: Demonstration of different cluster methods (6 clusters); Different countries in Europe and Africa

#### 4.2.4 Textual Data

Textual data refers to data sets with non-normalised long texts generated by individuals. These can be reviews, social media posts or the most prominent variant, search logs. In order to calculate the similarity of texts or, in the case of this application, to be able to draw conclusions about the same person we use the Jaccard Similarity.



Jaccard similarity is used to calculate the similarity of two sentences. A value between 0 and 1 is calculated, increasing values signify high text similarity. If none or only a few texts are similar it means there are only individual users in the dataset and no clusters of similar users can be formed. The data cannot be classified as safe. Figure 28 illustrates an output of the application. Here, the risk analysis was applied to two different data sets. The AOL dataset, which contains search engine search texts, and the Amazon dataset<sup>11</sup>, which contains product reviews. Red means no similarities, a value of 0 for Jaccard Similarity, and green a value of 1. These results show that there are no similar texts in the AOL dataset and that it is therefore not safe to publish. The Amazon dataset has certain similarities and it is therefore (partly) safe to publish, as clusters of similar users can be formed.

Two different natural language processing (NLP) methods are being implemented as anonymisation techniques for textual data:

1. **Sentiment analysis:** An automatic evaluation of texts with the aim of recognising the sentiment/opinion of a text as positive or negative. Instead of the whole text, only the information about positivity and negativity remains in the dataset.
2. **Automatic identifying of named entities in texts:** The data controller can determine which types of named entities are automatically recognised and replaced. For example, all locations or persons can be identified with the help of an AI and then removed.

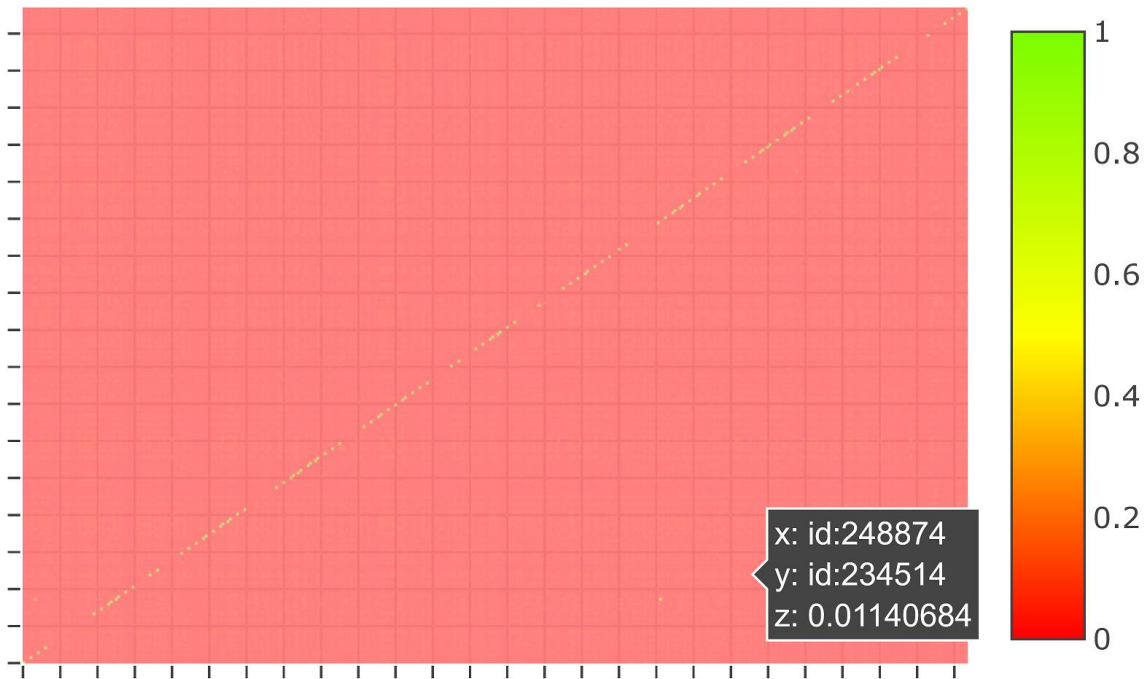
---

<sup>11</sup> <https://jmcauley.ucsd.edu/data/amazon/>





## AOL search logs



## Amazon reviews



Figure 28: Screenshot of textual data risk analysis (AOL search logs and Amazon reviews)



#### 4.2.5 Invoice Data

Invoice data, also financial transactions data is a type of dataset that contains information about transactions and payments of individuals including the time dimension. The basis for the risk analysis was a module developed within the Safe-DEED project (Bampoulidis, 2020a). Figure 29 shows the output of the risk analysis in the application. The UI was revised after D4.1 to include more descriptive tooltips and legends. The X-axis represents the number of data points and the Y-axis represents time. The individual time points are represented as points. The scale goes from green (safe) to red (unsafe) based on the specified privacy notion.

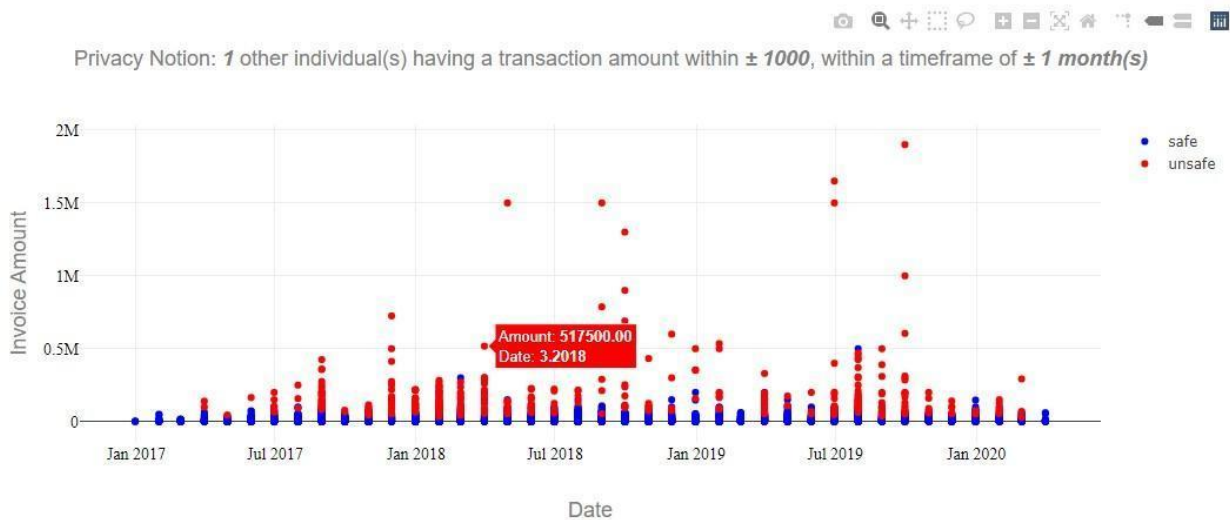


Figure 29: Screenshot of financial transactions data risk analysis with enhanced UI

The results of the risk analysis can now serve as input for the anonymisation task. The module will offer three different options.

- **Generalisation**, where the data controller can manually create a hierarchy based on the results of the risk analysis.
- **Bucketisation**, where the financial values are grouped together in optimal sized bins.
- **Time series clustering**, where not the individual values are considered, but the whole time series. The data controller can determine the number of clusters depending on the  $k$  or  $l$  value and similar time series are grouped together.

Figure 30 shows an example of a clustering of time series (adapted from Petitjean et al. (2011)) of one of the developed prototypes.

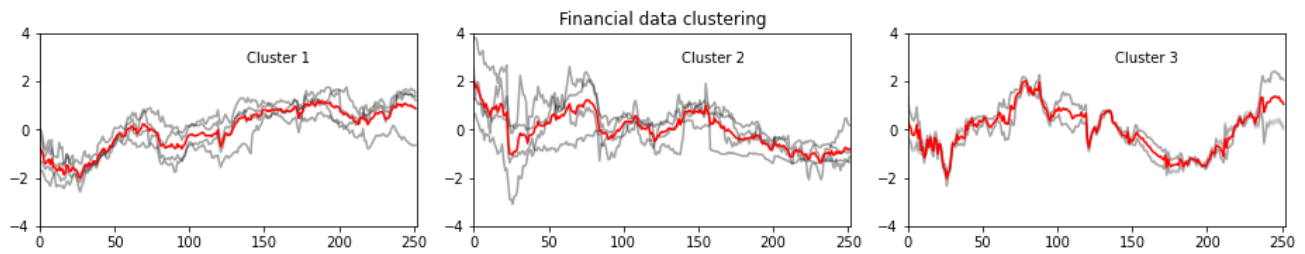


Figure 30: Time series clustering demonstration with stock data adapted from Petitjean et al. (2011)

#### 4.2.6 Aggregated Data

Aggregation-based data is data that contains aggregate values, such as sum, count, and average, about individuals. If the aggregate values are low and there is another sensitive aggregate attribute, then there could be a privacy breach. Figure 31 is an example output of this risk analysis module: the aggregate values of a dataset visualised in a bar plot with a horizontal line (k) representing the minimum acceptable value of an aggregation attribute. In this case, the records containing the attribute “shares” being below the acceptable value should be removed if there is another sensitive attribute (e.g., sum of income). The core of the risk analysis module was developed in Safe-DEED (Bampoulidis, 2020a) and modified to work with the application. The anonymisation module will provide two options:

1. Full suppression, where as soon as the data exceeds the value (k), it will be deleted
2. Microaggregation, in which the data is grouped and aggregated again.

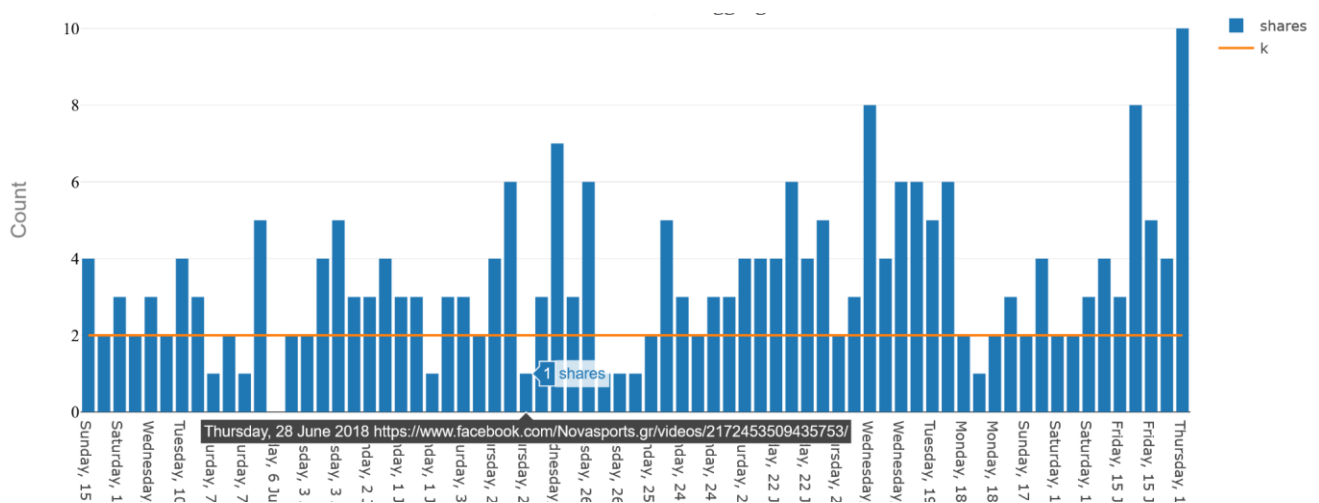


Figure 31: Screenshot of the aggregation-based data risk analysis

### 4.3 Application

The application is a toolkit designed and developed to incorporate the aforementioned mechanisms while also providing end users with an intuitive and usable UI. It offers a solution for importing datasets, configuring risk analysis or anonymization processes based on dataset type, queuing and parallelizing such processes, storing and displaying risk analysis findings, and creating anonymized datasets.

For this version of the deliverable, we provide the overall description of the application, including architecture and UI components, as it has been updated to employ the anonymization mechanisms described above. Furthermore, the description below includes the UI improvements done during the second half of the project.

#### 4.3.1 Supported Datasets

The files that should be used for the de-anonymization risk analysis and anonymization process should be formatted according to the following conventions.

- **File location and separator/delimiter:** The file has to follow the .CSV format. The delimiter type should also be one of the following (special)-characters: comma, semicolon or tab.
- **Data type:** The data types that are currently supported are: tabular, aggregated, financial transactions, textual and spatiotemporal data.
- **Title and short Description:** For each dataset that is imported into the application a title and a short description should be specified.

The data and metadata related to the datasets imported to the application are all stored locally, in the user's machine which is running the application.

#### 4.3.2 Application Architecture

The application is being developed in a loosely coupled manner, where every component is a standalone entity in a separate docker image. All the components/images needed for the application are deployed through a single docker-compose file.

The main components of the app's architecture are the following:

- The application Frontend/GUI
- The Coordinator Server, which is responsible for the necessary backend operations regarding the coordination of the backend components according to specific workflows.
- The Risk Analysis, as well as the recently introduced Anonymization Server, which undertake the ingestion of the dataset as well as the execution of the risk analysis and anonymization processes, that are described in the previous sub-sections.



- The Data Management System that connects the components (metadata) and stores the results of each risk analysis process.

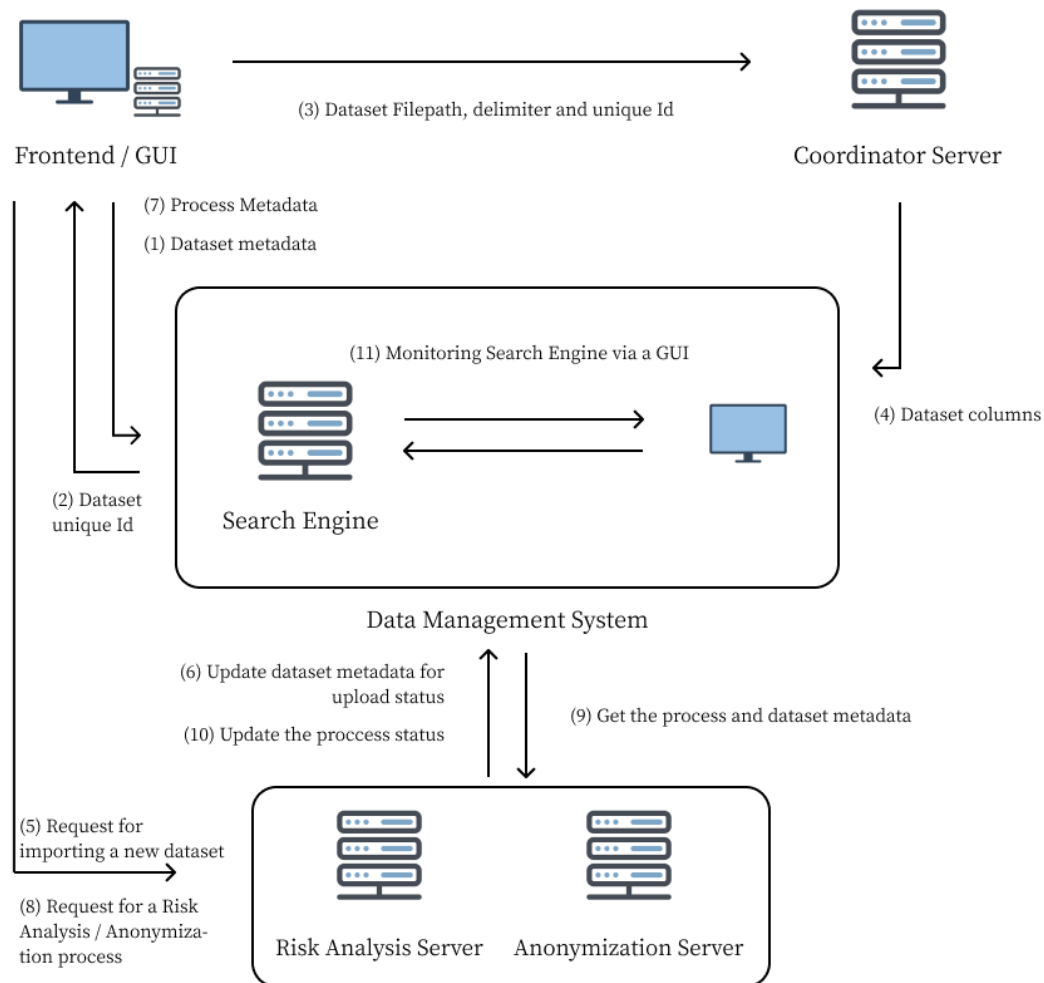


Figure 32: Application architecture diagram

The application architecture as well as the information flow between the application's components are depicted in Figure 32. The information flow is triggered as per the users' interaction with the application's frontend UI as described in the next sub-section. Specifically, the overall workflow is as follows:

- 1. Dataset Metadata:** The user chooses a dataset to be "uploaded" (imported) and sets some metadata for it (the dataset's .CSV file and delimiter, title, sort description and type of the data) by filling out the upload page's form.
- 2. Dataset unique Id:** Through a REST request the frontend/GUI sends these metadata to the Data Management System, where a new document (that contains them) is created in the index responsible for storing the metadata related to datasets.



**3. Dataset File path, delimiter and unique Id:** A REST request is also sent to the Coordinator Server with the unique Id of the newly created document (which contains the dataset's metadata), as well as the dataset's file path and delimiter.

**4. Dataset columns:** After receiving the metadata, the Coordinator Server reads the .CSV file and extracts the names of the columns as well as the total number of records contained in the file. Then, with the use of the unique Id, it updates the dataset's document in the Data Management System, thus completing the collection of all the necessary metadata for the dataset.

**5. Request for importing a new dataset:** With the dataset's metadata already collected, a REST request is sent to the Risk Analysis and Anonymization Servers APIs in order for the import of the dataset to start.

**6. Update dataset metadata for upload status:** After the process of importing the dataset is completed, the server updates the dataset's metadata. At this point, the dataset upload is complete and the user is able to initiate a new risk analysis or anonymization process.

**7. Process Metadata:** From the app's GUI, a user can initiate a process (risk analysis or anonymization) for a specific dataset and fill out the necessary parameters in the respective process page. After all the necessary metadata is set for the process, a new document is created in the index responsible for that type of process in the Data Management System.

**8. Request for a Risk Analysis / Anonymization process:** After the document containing the metadata of a process is created in the Data Management System, a REST Request is sent to the Risk Analysis or Anonymization Servers (depending on the type of the process) containing the unique Id of the process that needs to be initiated.

**9. Get the process and dataset metadata:** With the use of the process' unique Id, the server retrieves the metadata from the Data Management System and initiates the process based on the parameters specified.

**10. Update the process status:** If the process was a risk analysis, the Risk Analysis Server creates a new index in the Data Management System and stores the results once they are ready. When this operation is completed, the server changes the process's status to "Completed," and the user can review the results from the process's page. If the process was an anonymization, the Anonymization Server creates a new file in the user's datasets folder containing the newly anonymized dataset and sets the process status to "Completed" after the process finishes.

**11. Monitoring search engine via a GUI:** For the purposes of monitoring the Data Management System (both for the app's development and maintenance) a Kibana instance is deployed. With Kibana, a user (with the appropriate access rights) is able to review with relative ease the status of both the Elasticsearch node and the indices stored in it.





### 4.3.3 Application implementation

For the development of this application, a variety of modern frameworks, runtime environments and Databases were used.

For the frontend / graphical user interface (GUI) of the application, the framework used is Angular v10. Angular is a widely used, TypeScript-based, open-source web app framework capable of creating robust web applications. The Data Management System is based on the Elastic Stack<sup>12</sup> which comprises three fundamental components: Elasticsearch, Logstash, and Kibana. Elasticsearch is a distributed, free and open search and analytics engine for all types of data, built on Apache Lucene and is more than capable of meeting the storing as well as the communication requirements of the application. Logstash is a free and open server-side data processing pipeline that ingests data from a multitude of sources, transforms it, and then sends it to the Elastic stack. In the context of the application, Logstash is used to ingest to an Elasticsearch index the logs of the deployed Elasticsearch node. Lastly, Kibana is a free and open user interface that lets a user visualize Elasticsearch data and navigate the Elastic Stack and it is used for monitoring the Elasticsearch node deployed.

The Coordinator Server was developed using the Node.js<sup>13</sup> cross-platform. Node.js is a JavaScript runtime environment, built on Chrome's V8 JavaScript engine. Specifically, Node.js is used as an intermediate server (with the use of the back-end, web application framework Express.js) for the communication between the Angular framework and the Elasticsearch node.

The Risk Analysis Server is a Java component based on the Spring Boot framework<sup>14</sup>. For storing and efficient querying of the data locally, before it ingests the results to the data management system, the SQLite<sup>15</sup> database is used.

The Anonymization Server is programmed in Python with the use of Redis<sup>16</sup>, an open source, in-memory data store. It consists of three components: the (1) communication component, which uses the Python library Flask<sup>17</sup> to provide an API for communication with the search engine. The (2) message queue, in which the requests of the search engine are temporarily stored. A redis database for all requests in order to process them asynchronously. This means that long-running processes do not block the entire system and it also serves as a load balancer. Requests are no longer processed directly, therefore there is no restriction on the number of simultaneous requests. The (3) Anonymisation Toolkit itself, which applies the anonymisation methods. Chapter 4.3 gives an overview of the provided anonymisation methods.

---

<sup>12</sup> <https://www.elastic.co/what-is/elasticsearch>

<sup>13</sup> <https://nodejs.org/en/>

<sup>14</sup> <https://spring.io/projects/spring-boot>

<sup>15</sup> <https://www.sqlite.org/index.html>

<sup>16</sup> <https://redis.io/>

<sup>17</sup> <https://flask.palletsprojects.com/en/2.1.x/>



As already mentioned, the application is deployed as a Docker<sup>18</sup> container stack. Docker is an open platform for developing, shipping, and running applications that enables the separation of the application from the infrastructure. All the application's components are housed in separate docker images and deployed with a single docker-compose file. In this way, each component of the application is independent of the others yet capable of communicating with them through well-defined API's.

#### **4.3.4 Application functionality**

After the successful deployment of the application, a user can have access to the GUI by visiting the specified link on an internet browser. Upon entry, the user will be asked for their credentials in order to log in. After a successful authentication, the user is able to view all the imported datasets or to upload a new one. For each dataset, the user can initiate either a risk analysis or an anonymization process. After filling out all the required parameters, the chosen process can commence. All the previously initiated processes, depending on their type, can be found in the respective processing queues. When a process is completed, the user is capable of viewing the results of the process in the respective results page.

The next subsections detail the functionality that a user can accomplish through the GUI.

##### ***User Login***

The user can log in to the application by entering their TRUSTS credentials (see Figure 33). If the authentication process succeeds, they proceed to the Datasets Page.

---

<sup>18</sup> <https://www.docker.com/>





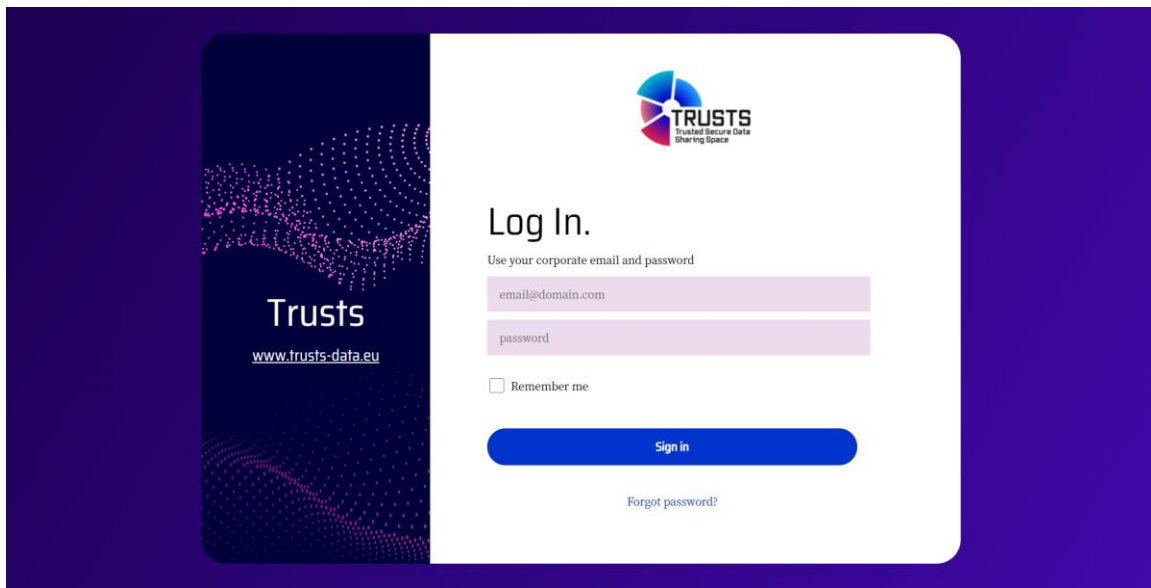


Figure 33: Sign in Page

At this stage of the application development, the authentication process is conducted through a local database. In later stages of development, the authentication will be achieved through the TRUSTS platform user authentication mechanisms.

After a successful authentication, the user is directed to the Datasets page. All the pages offered by the application follow the same design pattern, which consists of three components: the vertical navigation menu (on the left sidebar), the header and the content of the page (see Figure 34).

From the vertical menu, the user can navigate through the main functionality of the application, namely: Datasets, Risk Analysis, Anonymization and Processing Queue. At the top of each page, two buttons are located, the one for viewing the notifications of the user and the other for viewing the profile. The rest of the space, depending on the page, will display the appropriate content.

### **Datasets**

In the Datasets page, a user is able to view all the datasets imported to the application. There are two viewing options: the card view (set by default) and the list view. The user can specify the desired view with the use of the buttons located at the top-left of the page (below the user profile button).



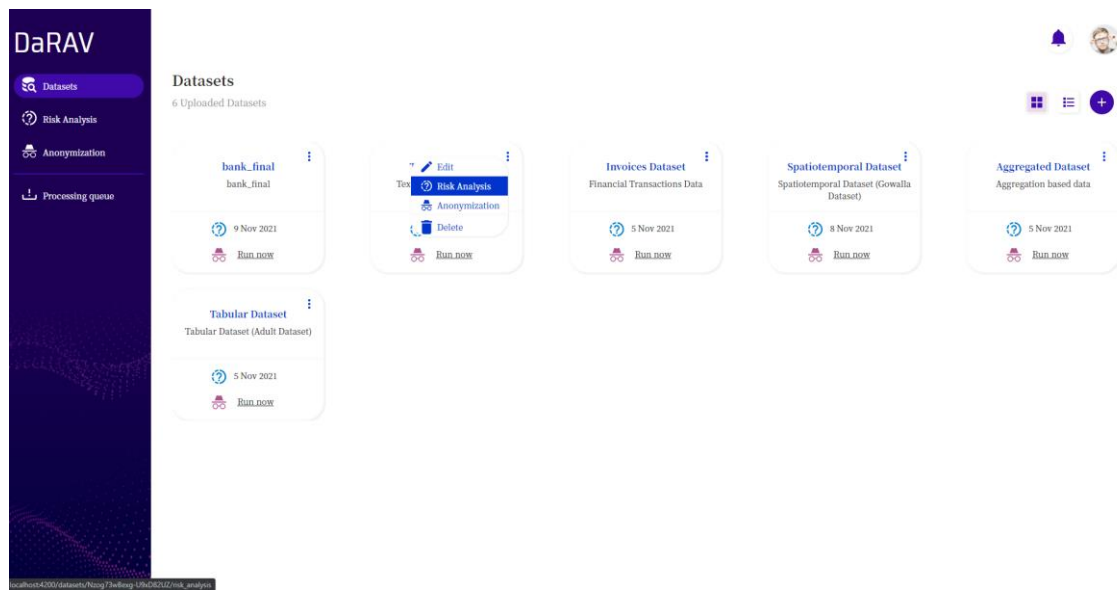
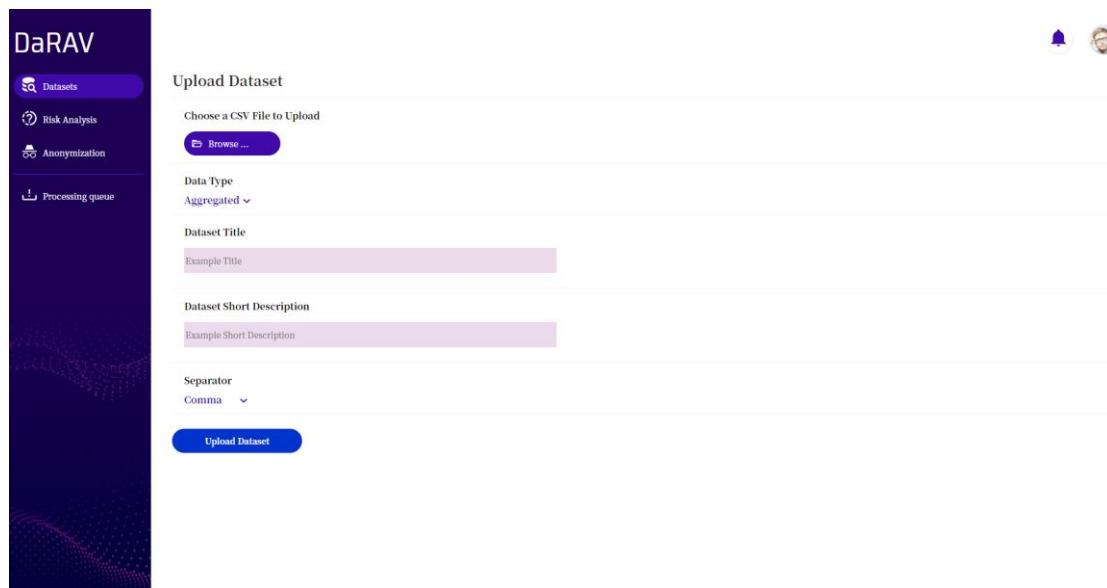


Figure 34: Datasets Page – card view

For each dataset, the title, description and last time of a Risk Analysis or Anonymization process run are displayed. From the options menu (three blue dots) the user has the ability to edit, delete or initiate a risk analysis or anonymization process for a dataset. While a dataset is being uploaded, the user is only able to edit or delete it. After the uploading process is completed, the options for a Risk Analysis or an Anonymization are enabled.

### ***Upload a Dataset***

A user can import a new dataset by selecting the “Upload Dataset” button located at the top-left of the Datasets page (below Figure 35 offers the user profile image).

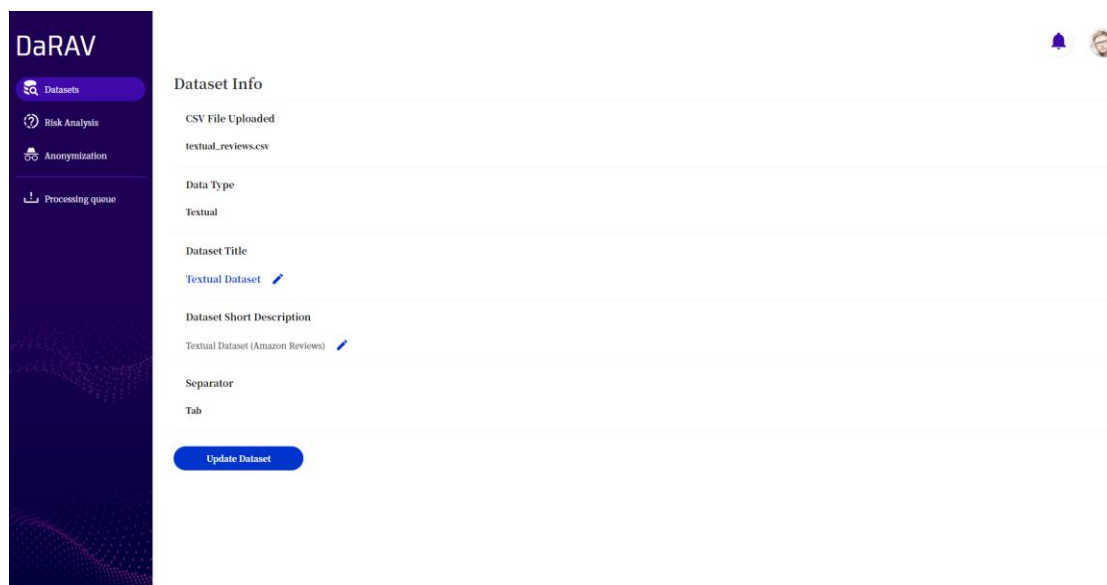


The screenshot shows the 'Upload Dataset' page in the DaRAV application. On the left is a dark blue sidebar with the DaRAV logo and navigation links: Datasets, Risk Analysis, Anonymization, and Processing queue. The main content area is white and titled 'Upload Dataset'. It contains a 'Choose a CSV File to Upload' section with a 'Browse ...' button. Below this are fields for 'Data Type' (set to 'Aggregated'), 'Dataset Title' (placeholder 'Example Title'), 'Dataset Short Description' (placeholder 'Example Short Description'), and 'Separator' (set to 'Comma'). An 'Upload Dataset' button is at the bottom.

Figure 35: Upload Dataset Page

In the Upload Dataset Page, a user can specify the .CSV file containing the dataset, the dataset's title and short description, delimiter/separator of the .CSV file as well as the data type of the dataset. After filling out this information, the importing process starts. Until this process is finished, the user is only able to edit or delete the dataset. After the completion of the process, the Risk Analysis and Anonymization options are enabled.

### ***Edit a dataset***



The screenshot shows the 'Dataset Info' page in the DaRAV application. The sidebar is identical to the previous page. The main content area is titled 'Dataset Info' and shows the details of an uploaded dataset: 'CSV File Uploaded' (textual\_reviews.csv), 'Data Type' (Textual), 'Dataset Title' (Textual Dataset with an edit icon), 'Dataset Short Description' (Textual Dataset (Amazon Reviews) with an edit icon), and 'Separator' (Tab). An 'Update Dataset' button is at the bottom.

Figure 36: Dataset Info Page

After selecting the “Edit” option of a dataset, the user is directed to the dataset’s info page (see Figure 36). In this page, all the basic information of the dataset is displayed. The user can edit the dataset’s title and short description by selecting the pen icon beside either the title or the short description.

### New Risk Analysis process

To start a new Risk Analysis process, a user must choose a dataset from the Datasets page and then select the “Run Now” button, or, from the dataset’s options menu, the “Risk Analysis” option.

**DaRAV**

**Risk Analysis**

Invoices Dataset  
12 Columns | Financial Transactions Data

Risk Analysis Method  
Invoices ▾

Name ▾	Individual Identifier	Invoice Date	Invoice Amount
PL	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Customer Code	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Order Number	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Order Entry Date	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Customer Wish Date	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Confirmed Delivery ...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Requested Order Lea...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Confirmed Order Lea...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Processing parameters  
Date format: yy.MM

Risk analysis parameters  
Number of other Individuals: 1  
Invoice Amount Within: 1000  
Within a timeframe of: 1 months ▾

**Start Risk Analysis**

Figure 37: Risk Analysis Page



Figure 38: De-anonymization Risk Analysis Page - The results of the process configured in Figure 37

Then, the user is redirected to the Risk Analysis page (see Figure 37). In the Risk Analysis page, depending on the data type of the dataset, the user will have to select an appropriate Risk Analysis method. Then, depending on the risk analysis method selected, the user will have to set the appropriate Attributes and Parameters that pertain to the method. After the required fields are filled out, the user can initiate the risk analysis process and get the results (see Figure 38).

### ***New Anonymization process***

With this version of the application a user is now able to anonymize datasets. In order to start a new Anonymization process, a user has to choose a dataset from the Datasets page and then select the “Run Now” button, or, from the dataset’s options menu, the “Anonymization” option.

**DaRAV**

**Anonymization**

tabular  
9 Columns | Tabular Dataset Short Description

Anonymization Method  
Hierarchy Anonymization

Name	Quasi Identifier	Hierarchy Level
sex	<input checked="" type="radio"/>	1
salary-class	<input checked="" type="radio"/>	2
race	<input checked="" type="radio"/>	1
workclass	<input checked="" type="radio"/>	1
marital-status	<input checked="" type="radio"/>	1
occupation	<input checked="" type="radio"/>	1
education	<input checked="" type="radio"/>	1
native-country	<input checked="" type="radio"/>	1

Processing parameters  
No Processing Parameters needed.

Anonymization Parameters and Files  
Choose a .json File containing the Attributes' Hierarchies:  
adult\_hierarchy.csv

[Browse ...](#)

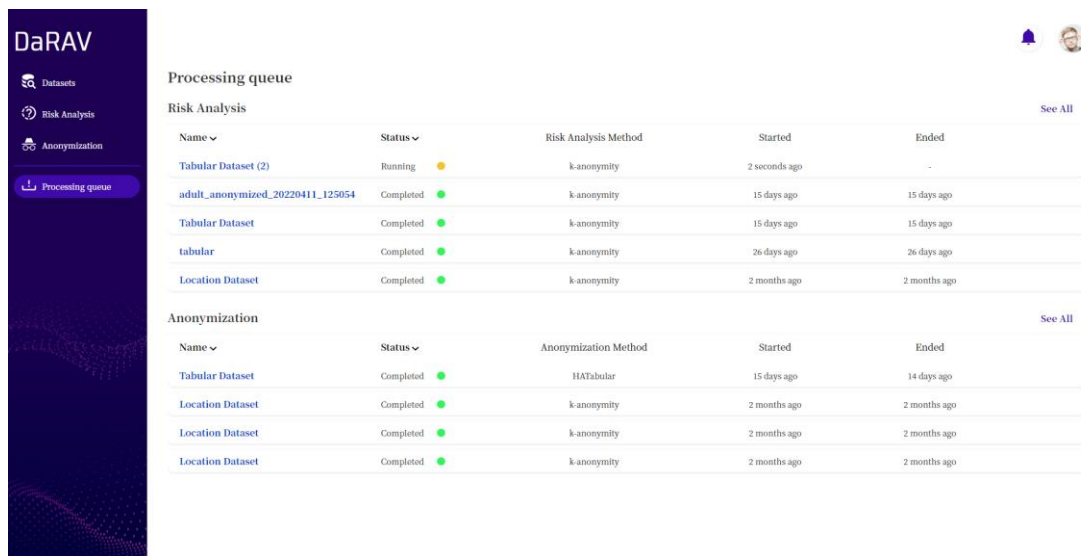
[Start Anonymization](#)

Figure 39: Anonymization Page

Then, the user is redirected to the Anonymization page (see Figure 39). In the Anonymization page, in a similar fashion to that of the risk analysis processes and depending on the type of the dataset selected, the user will have to select the appropriate attributes, parameters and files that pertain to the method. The user can choose between Hierarchy and Smart Anonymization methods. After the required fields are filled out, the user can initiate the anonymization process.

### ***Processing queue***

After the initiation of a new risk analysis or anonymization process, the user is redirected to the Processing Queue page (see Figure 40). In this page, previously initiated processes are displayed.



Processing queue				
<b>Risk Analysis</b> <a href="#">See All</a>				
Name	Status	Risk Analysis Method	Started	Ended
Tabular Dataset (2)	Running	k-anonymity	2 seconds ago	-
adult_anonymized_20220411_125054	Completed	k-anonymity	15 days ago	15 days ago
Tabular Dataset	Completed	k-anonymity	15 days ago	15 days ago
tabular	Completed	k-anonymity	26 days ago	26 days ago
Location Dataset	Completed	k-anonymity	2 months ago	2 months ago
<b>Anonymization</b> <a href="#">See All</a>				
Name	Status	Anonymization Method	Started	Ended
Tabular Dataset	Completed	HATabular	15 days ago	14 days ago
Location Dataset	Completed	k-anonymity	2 months ago	2 months ago
Location Dataset	Completed	k-anonymity	2 months ago	2 months ago
Location Dataset	Completed	k-anonymity	2 months ago	2 months ago

Figure 40: Processing Queue Page

For each process in the queue, information about the status (“Running”, “Completed” or “Canceled”), the name of the method used and the times of when the process started and when it ended are displayed. If a user wants to view all past processes, they can select the “See All” button beside each type of process (risk analysis or anonymization) and a complete listing will be displayed.

A user can view the results of a process by:

- Selecting the last process of a specific dataset from the Datasets page
- Selecting a process from either the Risk Analysis or Anonymization pages
- Selecting a process from the Processing Queue page

After choosing a process, with one of the methods above, the user is then redirected to the appropriate results page.

#### 4.3.5 Risk Analysis results

After selecting a Risk Analysis process, the user is redirected to the De-anonymization Risk Analysis page (see Figure 41).

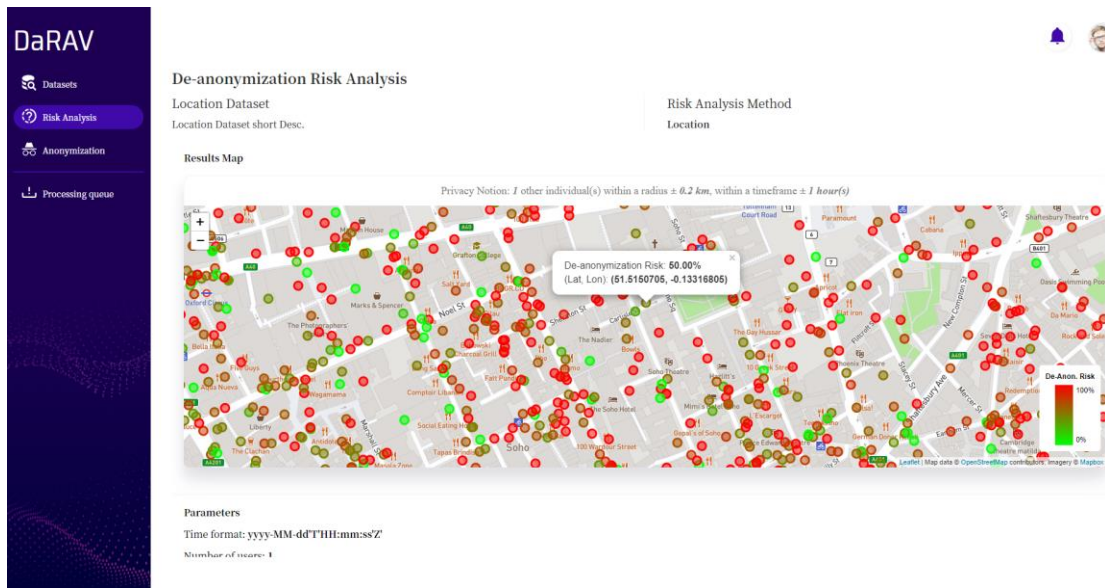


Figure 41: De-anonymization Risk Analysis Page

In this page, information regarding a specific risk analysis process is provided including metadata of the dataset under inspection. While the process' status is "Running" the user can only view the information related to the specific process and dataset. When the status changes to "Completed", the results of the process are displayed in the appropriate form (e.g. diagram or map) for the user to review them.

#### 4.3.6 Anonymization Results

After the completion of an Anonymization process, users can find the resulting, anonymized, dataset in the user datasets folder in their filesystem.

**UI Improvements:** Updated plots have been implemented for the risk analysis visualizations of the application. Specifically, in the risk analysis processes' resulting plots for datasets of financial transactions and spatiotemporal types, the privacy notion of the process is now being displayed (see Figure 37), in addition to the process results, in order for the user to better understand the plot's insights. Furthermore, all of the risk analysis plots have been revised to include more detailed tooltips, legends, and more user-friendly colour choices (see Figures 38 and 41).

**Publications:** Task 4.3 has produced one publication (Bampoulidis et. al, 2020b), where we identified the challenges of de-anonymisation and anonymisation in data sharing, which we are addressing in the current task.



## 5 Application to TRUSTS Platform

In this section, we will first give an overview of the TRUSTS platform (see deliverable D2.6 and follow-ups for more details). Then we will elaborate on some of the above-mentioned technologies that are used to enable UC2.

### 5.1 TRUSTS Platform Infrastructure

The TRUSTS platform is a set of nodes, each owned by a different organization and running on its own infrastructure. A critical tenet of the design of TRUSTS is that assets like data, services, or applications should not leave the owner's premises without a contract and without additional means to make transactions more trustworthy. This is exceptionally well suited for the case when organizations wish to make use of data owned by parties with whom they cannot share data themselves for commercial or legal reasons. Likewise, it allows organizations to make their data useful to others and thus monetizable without actually transferring the data.

From the architectural point of view, each organization runs a TRUSTS node that contains



- A platform interfaces.
- A trusted connector instance.
- Usage control systems.
- A smart contract executor.

In general, the architecture supports the following functionalities. First, the data providers keep their assets, only transferring the data after signing and settling a contract. In addition, every access to an asset depends on checking for authorization against a clearinghouse whose integrity is guaranteed by means of distributed ledger technologies. Finally, confidentiality and integrity of the communication between the systems of different organizations are done through secure communication protocols, and each communication event can be subject to the above-mentioned access control policies.

In particular, the organizations can make use of these components in the following two ways. They can purchase an application that will process their data on their premises. In this way, they can use the methods developed by others but still keep complete control of their data. Secondly, they can purchase access to a service. In this way, they can leverage the technology and data that other organizations possess but which, for whatever reasons, can not be transferred to them.

By combining these two opportunities, TRUSTS can support privacy-preserving computations. In particular, parties can install applications that access their data on their premises. Each of these applications can communicate among themselves through channels that are secured and prevent eavesdropping via the use of cryptographic protocols. Also, all transactions can be logged and checked against contractual agreements. It is further possible to transfer the applications (the images themselves), and their usage can be subject to contractual terms.

## 5.2 TRUSTS Platform and Secure Computation

When it comes to personal data, common trading practices for non-private data are prohibited, so TRUSTS becomes a data market for non-private data and services market and services provider for personal private data.

We were able to map out ways (architecture) to collaborate over personal private data and also enable running advanced analytics, developed by third party's companies on personal private data, all while complying with data protection regulations, and preserving full privacy.

For example, in the architecture below, we describe a way to collaborate over private personal data using HE and Spooky shared encryption key.



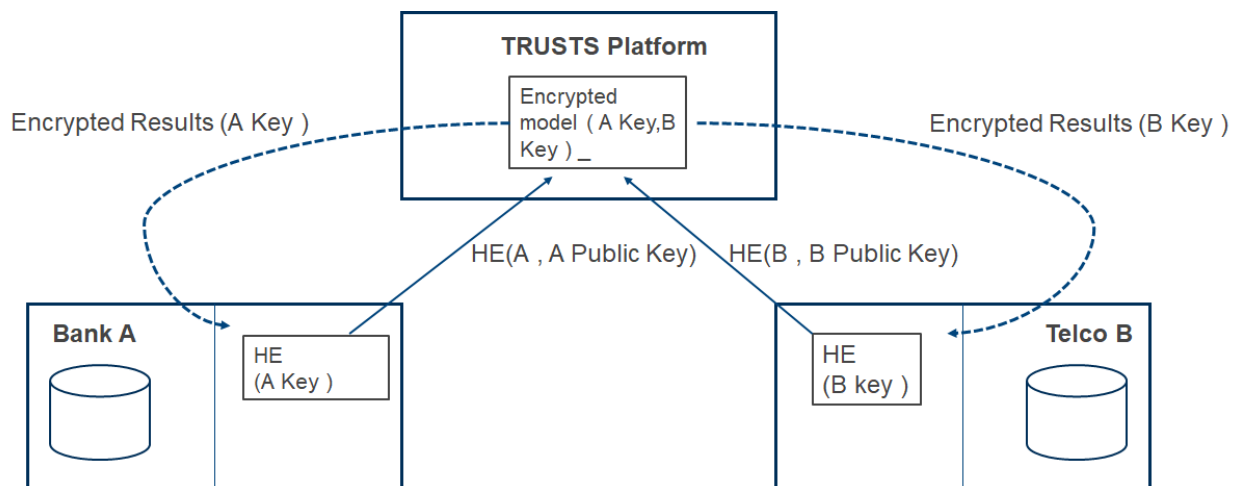


Figure 42: High level architecture of collaborating over private personal data using HE and Spooky shared encryption key

On the other hand, there is an option to use the TRUSTS cloud services as a service provider to the data owners, where all of the computations will be done on TRUSTS services. In the example below, we are using MPC protocol, to collaborate over private sensitive data, while all of the computation is done on TRUSTS servers.

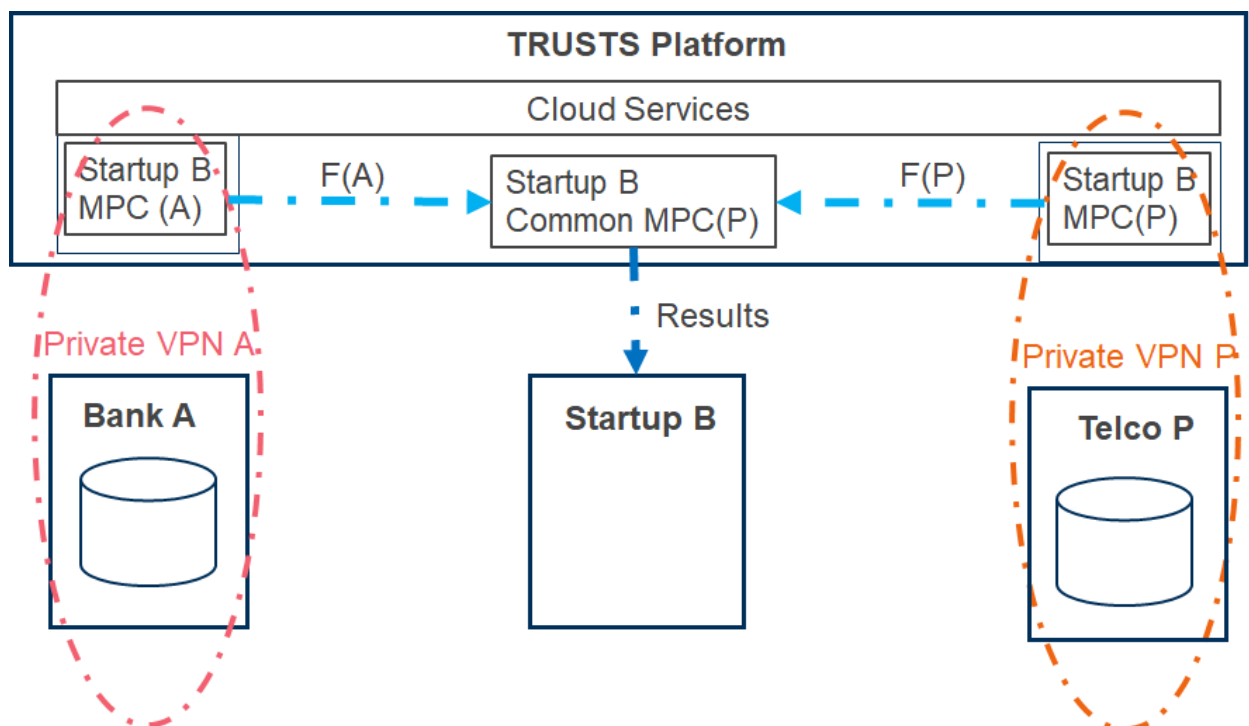


Figure 43: High level architecture of MPC protocol, to collaborate over private sensitive data, while all of the computation is done on TRUSTS servers

### 5.3 UC2

According to the TRUSTS platform architecture, the developed application will be available through the App Store of the platform, from which the users will be able to download and run locally on their premises. The figure below gives an overview of the usage of the application. Given two actors – a data seller and a data buyer – the usage flow is as follows:

1. The data seller and the data buyer download the application from the TRUSTS platform to their premises. The application needs to be executed on the TRUSTS users' premises because non-anonymised, privacy-sensitive, personal data are processed by the application, and such data should not be uploaded to the platform.
2. The data seller imports their non-anonymised, privacy-sensitive, personal data to the application.
3. The data seller uses the de-anonymisation risk analysis modules.
4. The data seller uses the anonymisation modules.
5. The data seller transfers the anonymised data to the data buyer. This will be feasible using the capabilities of the IDS Trusted Connector.

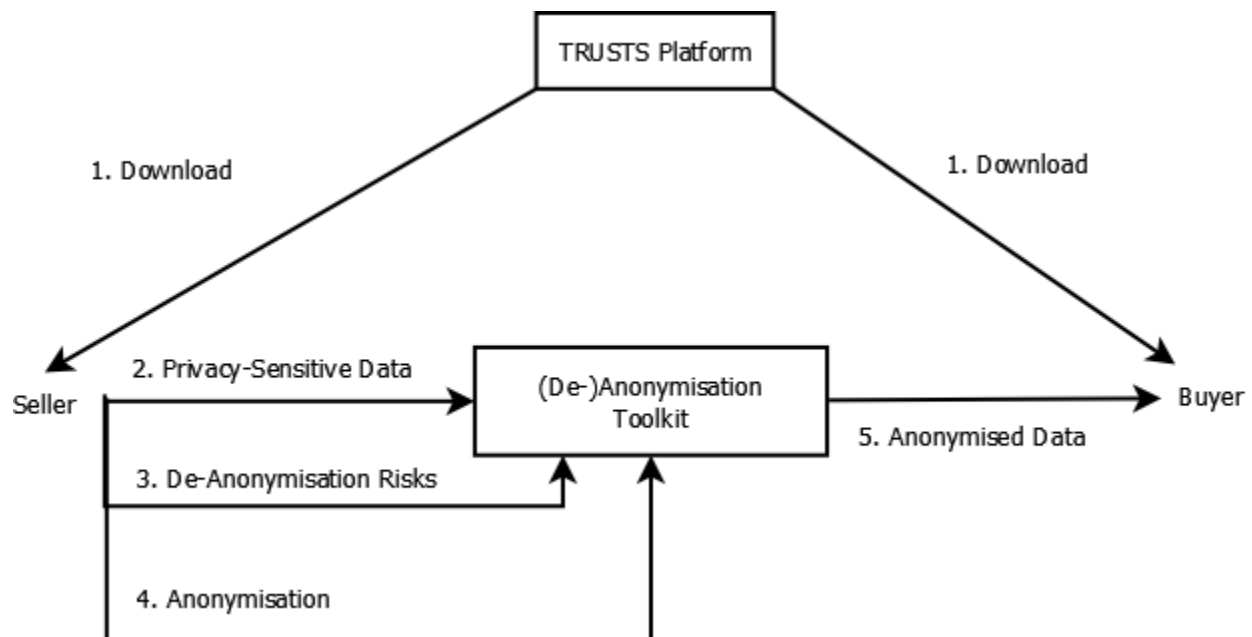


Figure 44: T4.3 application usage flow

UC2 comprises two companies (NOVA and PB) that will exchange data through a PSI protocol. The data that will be exchanged are tabular (i.e., one row corresponds to one individual), corresponding to information about customers. The UC2 partners would follow steps 1., 2., 3., and, if necessary, 4., of the usage flow described above. Since the data to be exchanged is tabular, the suitable de-anonymisation risk analysis modules are K-Anonymity and L-Diversity (if



there exists a sensitive attribute in the data). If the UC partners are content with the de-anonymisation risks in their datasets, then they may proceed to executing the PSI protocol; if not, then they may use the anonymisation modules, which will be developed in the future, prior to executing the PSI protocol.

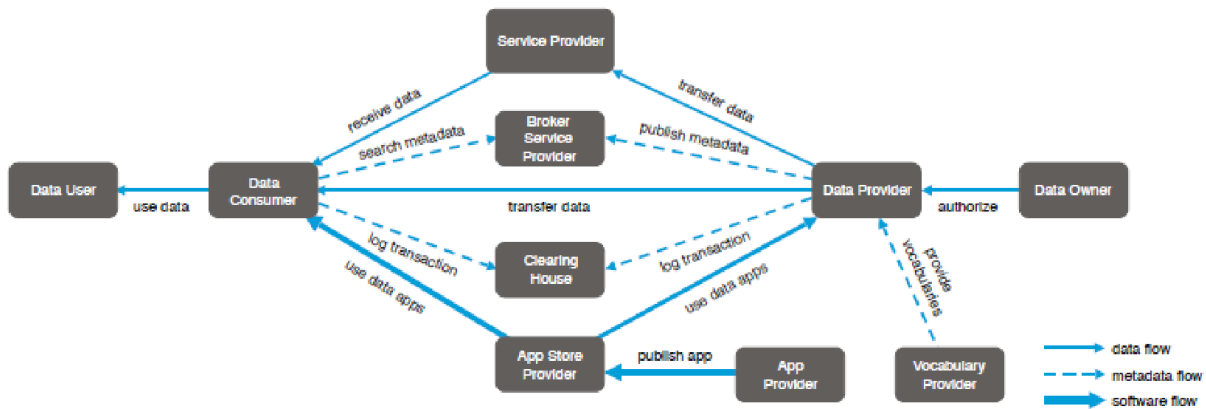


Figure 45: Roles and Interaction in the Industrial Data Space

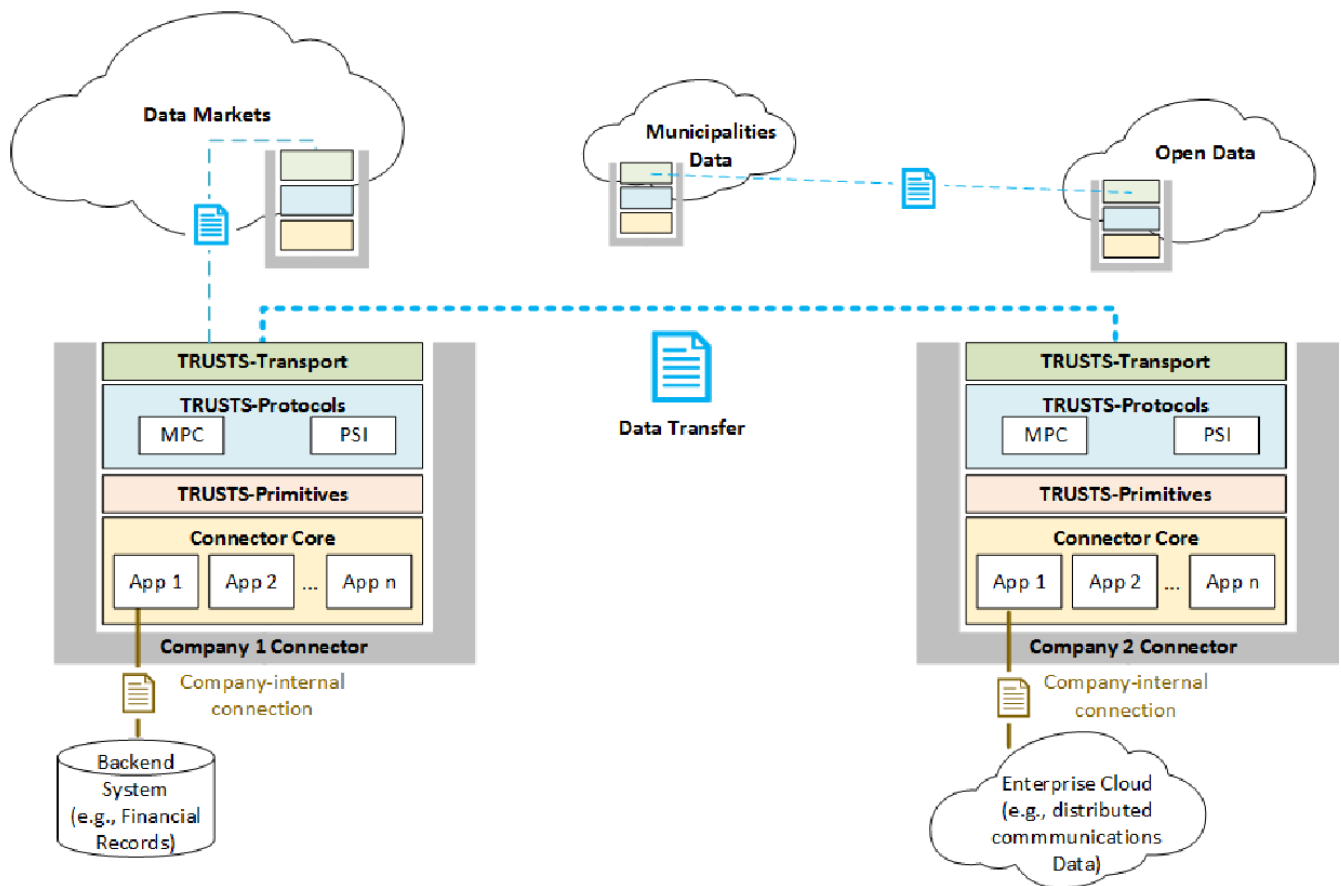


Figure 46: The TRUSTS Architecture

## 6 Conclusions and Next Actions

This report provides an overview as well as concrete recommendations regarding privacy by design. It offers an insight into the most promising privacy-enhancing techniques and algorithms based on cryptography. Furthermore, the applicability of these techniques was demonstrated by choosing concrete protocols and algorithms for UC1 and UC2. Those protocols and algorithms will be developed further to meet the UC needs.

The PSI protocol will be completely redesigned. This will allow achieving a tighter integration to the TRUSTS platform. It will also improve the technology readiness level as well as performance. As for now, it is planned to use Rust, which is a state-of-the-art programming language designed for performance and safety.

While the first part of task 4.3 focused on risk analysis, the second part now deals with suitable anonymisation methods. These methods did not yet exist in this form and are implemented with state-of-the-art concepts. This includes several applications of Natural Language Processing (NLP) and Unsupervised Learning.

As already stated in the first part of the task, the existing anonymisation tools, such as ARX (Prasser et. al, 2020) and Amnesia<sup>19</sup>, do not yet support complex, high-dimensional data. For this reason, they had to be re-implemented based on current scientific literature.

The finished application will support the following types of datasets, from risk analysis to anonymisation:

- **Tabular data**, i.e. the classic type of datasets in table form. Here, the data can be anonymised with the help of predefined hierarchies.
- **Aggregated data**, i.e. any data with aggregated values, such as sum, count and average. Based on the desired degree of anonymisation (k-anonymity level), values are suppressed as soon as they exceed k or are regrouped and aggregated (microaggregation).
- **Location data**, i.e. any data containing geographical values, can be clustered to k size and thereby anonymised.
- **Invoice data**, i.e. datasets containing information on payments and the time of the payment. Here, the data can be generalised with hierarchies, bucketisation (values grouped together) or time series clustering.
- **Textual data**, i.e. any kind of text generated by individuals which includes personal information. Two anonymisation options are offered and can be selected by the Data Controller depending on the application. The identification of named entities with transformer models and sentiment analysis, where only the polarity of the text is stored.

---

<sup>19</sup> <https://amnesia.openaire.eu/>



## 7 References

- Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K. and Zhang, L. (2016). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (pp. 308-318).
- Albrecht, M. R., Rechberger, C., Schneider, T., Tiessen, T., & Zohner, M. (2015). Ciphers for MPC and FHE. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 430-454). Springer, Berlin, Heidelberg.
- Bampoulidis A., Markopoulos I., Lupu M. (2019) PrioPrivacy: A Local Recoding K-Anonymity Tool for Prioritised Quasi-Identifiers. IEEE/WIC/ACM International Conference on Web Intelligence - Companion Volume October 2019 Pages 314–317 <https://doi.org/10.1145/3358695.3360918>
- Bampoulidis, A. (2020a). D5.10 Report on the application of re-identification techniques on use-case data v2. Safe-DEED.
- Bampoulidis, A., Bruni, A., Markopoulos, I., & Lupu, M. (2020b). Practice and Challenges of (De-) Anonymisation for Data Sharing. In International Conference on Research Challenges in Information Science (pp. 515-521). Springer, Cham.
- Barbaro, M., Zeller, T., & Hansell, S. (2006). A face is exposed for AOL searcher no. 4417749. New York Times, 9(2008), 8.
- Boehm, M., Antonov, I., Baunsgaard, S., Dokter, M., Ginhör, R., Innerebner, K., Klezin, F., Lindstaedt, S., Phani, A., Rath, B. and Reinwald, B. (2019). SystemDS: A Declarative Machine Learning System for the End-to-End Data Science Lifecycle. arXiv preprint arXiv:1909.02976.
- Bogdanov, D., Kamm, L., Laur, S., Pruulmann-Vengerfeldt, P., Talviste, R., & Willemsen, J. (2014). Privacy-preserving statistical data analysis on federated databases. In Annual Privacy Forum (pp. 30-55). Springer, Cham.
- Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, H.B. and Van Overveldt, T. (2019). Towards federated learning at scale: System design. SysML.
- Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. In International conference on the theory and application of cryptology and information security (pp. 409-437). Springer, Cham.
- De Montjoye, Y. A., Hidalgo, C. A., Verleysen, M., & Blondel, V. D. (2013). Unique in the crowd: The privacy bounds of human mobility. Scientific reports, 3(1), 1-5.



Derler, D., Ramacher, S., & Slamanig, D. (2017). Homomorphic proxy re-authenticators and applications to verifiable multi-user data aggregation. In International Conference on Financial Cryptography and Data Security (pp. 124-142). Springer, Cham.

Domingo-Ferrer, J., & Mateo-Sanz, J. M. (2002). Practical data-oriented microaggregation for statistical disclosure control. IEEE Transactions on Knowledge and data Engineering, 14(1), 189-201.

Dwork, C. (2008). Differential privacy: A survey of results. In International conference on theory and applications of models of computation (pp. 1-19). Springer, Berlin, Heidelberg.

Dua, D. and Graff, C. (2017). UCI machine learning repository. URL: <https://archive.ics.uci.edu/ml/index.php>. [Accessed 2021-05-24].

Federated, TensorFlow. "Machine Learning on Decentralized Data." TensorflowFL. (2020) URL: <https://www.tensorflow.org/federated> [accessed 2020-10-13].

Fung, B. C., Wang, K., & Yu, P. S. (2005). Top-down specialization for information and privacy preservation. In 21st international conference on data engineering (ICDE'05) (pp. 205-216). IEEE.

Garg, S., & Srinivasan, A. (2018). Two-round multiparty secure computation from minimal assumptions. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 468-499). Springer, Cham.

Gentry C. (2009a) Computing on encrypted data. In International Conference on Cryptology and Network Security (pp. 477-477). Springer, Berlin, Heidelberg.

Gentry C. (2009b) Fully homomorphic encryption using ideal lattices. In Proceedings of the forty-first annual ACM symposium on Theory of computing (pp. 169-178).

Ghinita, G., Tao, Y., & Kalnis, P. (2008). On the anonymization of sparse high-dimensional data. In 2008 IEEE 24th International Conference on Data Engineering (pp. 715-724). IEEE.

Goldreich, O., Micali, S., & Wigderson, A. (2019). How to play any mental game, or a completeness theorem for protocols with honest majority. In Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali (pp. 307-328).

Grassi, L., Rechberger, C., Rotaru, D., Scholl, P., & Smart, N. P. (2016). MPC-friendly symmetric key primitives. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 430-443).

Greene, W.H. (2003). Econometric analysis. Pearson Education India.

Hagedoorn, T. R., Kumar, R., & Bonchi, F. (2020). X2R2: a tool for explainable and explorative reidentification risk analysis. Proceedings of the VLDB Endowment, 13(12), 2929-2932.





Hsu, J., Gaboardi, M., Haeberlen, A., Khanna, S., Narayan, A., Pierce, B. C., & Roth, A. (2014). Differential privacy: An economic method for choosing epsilon. In 2014 IEEE 27th Computer Security Foundations Symposium (pp. 398-410). IEEE.

Iyengar, V. S. (2002). Transforming data to satisfy privacy constraints. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 279-288).

Jarecki, S., & Liu, X. (2010). Fast secure computation of set intersection. In International Conference on Security and Cryptography for Networks (pp. 418-435). Springer, Berlin, Heidelberg.

Kabir, M.E., Wang, H. & Bertino, E. Efficient systematic clustering method for k-anonymization. (2011) *Acta Informatica* 48, 51–66 .

Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K.A., Charles, Z., Cormode, G., Cummings, R. and D'Oliveira, R.G. (2019). Advances and Open Problems in Federated Learning.

Kales, D., Rechberger, C., Schneider, T., Senker, M., & Weinert, C. (2019). Mobile private contact discovery at scale. In 28th USENIX Security Symposium (USENIX Security 19) (pp. 1447-1464).

LeFevre, K., DeWitt, D. J., & Ramakrishnan, R. (2005). Incognito: Efficient full-domain k-anonymity. In Proceedings of the 2005 ACM SIGMOD international conference on Management of data (pp. 49-60).

Li, T., & Li, N. (2009). On the tradeoff between privacy and utility in data publishing. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 517-526).

Li, T., Sahu, A.K., Talwalkar, A. and Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3), pp.50-60.

Li, W., Ding, S., Chen, Y., Wang, H. and Yang, S. (2019). Transfer learning-based default prediction model for consumer credit in China. *The Journal of Supercomputing*, 75(2), pp.862-884.

Ma, J., Naas, S. A., Sigg, S., & Lyu, X. (2022). Privacy-preserving federated learning based on multi-key homomorphic encryption. *International Journal of Intelligent Systems*.

Machanavajjhala, A., Kifer, D., Gehrke, J., & Venkitasubramaniam, M. (2007). l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1), 3-es.



McMahan, B., Moore, E., Ramage, D., Hampson, S. and y Arcas, B.A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics* (pp. 1273-1282). PMLR.

Ozbayoglu, A.M., Gudelek, M.U. and Sezer, O.B. (2020). Deep learning for financial applications: A survey. *Applied Soft Computing*, p.106384.

Pan, S.J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), pp.1345-1359.

Petitjean, F., Ketterlin, A., & Gançarski, P. (2011). A global averaging method for dynamic time warping, with applications to clustering. *Pattern recognition*, 44(3), 678-693.

Prasser, F., Eicher, J., Spengler, H., Bild, R., & Kuhn, K. A. (2020). Flexible data anonymization using ARX—Current status and challenges ahead. *Software: Practice and Experience*, 50(7), 1277-1304.

Rechberger, C., & Walch, R. (2022). Privacy-preserving machine learning using cryptography. In *Security and Artificial Intelligence* (pp. 109-129). Springer, Cham.

Song, Y., Dahlmeier, D., & Bressan, S. (2014). Not so unique in the crowd: a simple and effective algorithm for anonymizing location data. In *PIR@ SIGIR*.

Slijepčević, D., Henzl, M., Klausner, L. D., Dam, T., Kieseberg, P., & Zeppelzauer, M. (2021). k-Anonymity in practice: How generalisation and suppression affect machine learning classifiers. *Computers & Security*, 111, 102488.

Sousa, S., Kern, R. (2022). How to keep text private? A systematic review of deep learning methods for privacy-preserving natural language processing. *Artificial Intelligence Review*. <https://doi.org/10.1007/s10462-022-10204-6>.

Sweeney, L. (2000). Simple demographics often identify people uniquely. *Health (San Francisco)*, 671(2000), 1-34.

Sweeney, L. (2002a). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05), 557-570.

Sweeney, L. (2002b). Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05), 571-588.



Suryanto, H., Guan, C., Voumard, A. and Beydoun, G. (2019). Transfer Learning in Credit Risk. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 483-498). Springer, Cham.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C. and Liu, C. (2018). A survey on deep transfer learning. In International conference on artificial neural networks (pp. 270-279). Springer, Cham.

Xiao, X., & Tao, Y. (2006). Anatomy: Simple and effective privacy preservation. In Proceedings of the 32nd international conference on Very large data bases (pp. 139-150).

Yang, Q., Liu, Y., Chen, T. and Tong, Y. (2019). Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), pp.1-19.

Yao, A. C. C. (1986). How to generate and exchange secrets. In 27th Annual Symposium on Foundations of Computer Science (sfcs 1986) (pp. 162-167). IEEE.

Yeh, I.C. and Lien, C.H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2), pp.2473-2480.

Yin, H., Mallya, A., Vahdat, A., Alvarez, J. M., Kautz, J., & Molchanov, P. (2021). See through gradients: Image batch recovery via gradinversion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 16337-16346).

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H. and He, Q. (2020). A comprehensive survey on transfer learning. Proceedings of the IEEE, 109(1), pp.43-76.

