RoG[§]: A Pipeline for Automated Sensitive Data Identification and Anonymisation

Sotiris Nikoletos,* Stefanos Vlachos,*† Efstathios Zaragkas,*†
Costas Vassilakis,* Christos Tryfonopoulos,* and Paraskevi Raftopoulou*
*Dept. of Informatics and Telecommunications, University of the Peloponnese, Greece
†Institute of Informatics and Telecommunications, NCSR Demokritos, Greece
Email: {dit2213cst, dit2202dsc, dit2203dsc, costas, trifon, praftop}@go.uop.gr

***Abstract*—Nowadays, the amount of data available online is constantly increasing. This data may contain sensitive or private information that can expose the person behind the data or be misused by malicious actors for identity theft, stalking, and other nefarious purposes. There is thus, a growing need to protect individuals' privacy and prevent data breaches in several application domains. Protecting data privacy though, is a complex and multifaceted issue that involves a range of legal, ethical, and technical considerations. In this paper, we discuss the challenges associated with data protection, the role of automated tools, and the effectiveness of identifying and anonymising sensitive data. We then, propose a fully-automated process for sensitive data identification and anonymisation, based on Natural Language Processing (NLP) techniques, that can be applied both in big diverse datasets and to a wide range of domains.**

***Keywords*—** sensitive/private data, automated process, pipeline, anonymisation, NLP, NER, k-anonymity

I. Introduction

In today's digital age, where users share their data through personal devices or applications they use in their everyday lives, the amount of data available online is constantly increasing. The available data can be used for research or commercial purposes, e.g, in marketing or in forecasting future trends. However this data may contain sensitive information, as an address, the user's age, a bank account, travel patterns, and other aspects that can expose the person behind the data or be misused by malicious actors for identity theft, stalking and other nefarious purposes. There is thus, a growing need to protect individuals' privacy and prevent data breaches. Yet, data privacy is a complex and multifaceted issue that involves a range of legal, ethical, and technical considerations. At its core, data privacy refers to the right of individuals to control and protect their personal information. To achieve this, robust data protection mechanisms must be considered, including *sensitive data identification and privacy preservation*.

Sensitive data identification involves identifying and categorising data based on its potential privacy implications. Additionally, privacy preservation can be achieved via *data anonymisation* that is the "process by which personal data is altered in such a way that a data subject can no longer be identified directly or indirectly, either by the data controller alone or in collaboration with any other party" [1]. Data anonymisation may include methods such as masking, generalization, and perturbation, which help to remove or obfuscate personally identifiable information. Although data anonymisation techniques seem suitable to protect individuals' privacy, manually identifying and anonymising sensitive data is a tedious and time-consuming process. This is why automated tools, such as Presidio[1] and Amnesia,[2] are gaining the attention of the community. It is crucial though, that the tools-supported anonymisation process protects personal information, while still allowing for meaningful analysis and research on the anonymised dataset.

Within ENIRISST+,[3] an infrastructure that has research and business orientation in the domains of transport, supply chain, ports, shipping, and tourism, we are taking a closer look at the transportation sector. Note that transportation is one of the most regulated industries, adopting numerous regulations that require the protection of sensitive data, such as the General Data Protection Regulation (GDPR).[4] However, protecting sensitive data in transportation is not trivial as there are two significant challenges: (1) Data Volume - A vast amount of data is generated, (2) Data Diversity - Data in various formats, such as location data, personal information, and travel patterns, is produced. It is though, difficult to (manually) identify and protect all types of sensitive data resulting from transportation.

In this work, we present a *fully-automated* process for *sensitive data identification and anonymisation*; Natural Language Processing (NLP) techniques, such as Name Entity Recognition (NER) and regular expressions (regex), are used to identify sensitive data, followed by anonymisation techniques, like k-anonymity algorithm, to replace the sensitive data with pseudonyms or generic terms. The proposed process can be applied in *big diverse datasets* and to a *wide range of domains*. The advantages of the proposed process are as follows.

- It performs *automated sensitive data identification*; there is no need for the users of the process to manually

[§]The Ring of Gyges was a magic ring mentioned by the philosopher Plato; it gives its owner the power to become invisible at will.

[1]https://www.presidio.com
[2]https://amnesia.openaire.eu
[3]https://enirisst-plus.gr
[4]https://gdpr-info.eu

investigate the dataset for potential personal or sensitive information.

- Its use is *straightforward*; only a few initial configurations are needed and no sophisticated knowledge of data science or deep learning is required.
- It is *lightweight*, due to the low computational requirements of the used/integrated tools.

Both the source code of the proposed pipeline[5] and the used dataset[6] are made *publicly available* for further use. This work can be considered as a first step towards building a fully-featured anonymisation application, with which users will interact via a friendly interface and will be able to fully automatically and efficiently anonymise any given dataset.

The rest of this paper is organised as follows. In Section II, we present the state-of-the-art research related to our work. Section III unfolds the methodology adopted, including the used tools and the development of the pipeline, while Section IV provides a use-case scenario, describing the dataset and commending on the produced results. Finally, Section V concludes the paper and gives future directions.

II. Related Work

Several researchers have focused on developing methods for identifying and protecting (through anonymisation techniques) sensitive data [2]–[6]. The work in [2] presents a comprehensive review of previous research in the domain, discusses the related challenges, and proposes NLP and Machine Learning (ML) as the most suitable tools for sensitive data identification and privacy preservation; the different NLP-based approaches are grouped under four categories. In the same context, the work in [3] takes full advantage of Artificial Intelligence (AI) and combines it with NLP to build a model for privacy preservation in healthcare and justice domains.

Regarding data anonymisation, related work includes research by [4], which develops a privacy-preserving framework for data anonymisation, through data generalisation and perturbation, in healthcare and business domains, that effectively anonymise sensitive data while preserving data utility. CHORUS [7] is a framework for building privacy mechanisms, based on the cooperation between the mechanism itself and standard SQL database engines. The work in [5] proposes a privacy-preserving technique for social network data anonymisation; the authors utilise a combination of graph theory and differential privacy to anonymise social network data, while maintaining the structural properties of the network. Moreover, the work in [6] evaluates ARX Data Anonymisation and Amnesia, two popular data anonymisation open-source tools, and concludes that although using Amnesia may have limitations and cause errors, the anonymisation process it uses is straightforward, while ARX Data Anonymisation is preferred with big datasets.

Overall, these studies highlight the importance of sensitive data identification and anonymisation in protecting individual privacy and ensuring data security. The proposed methods offer promising solutions for addressing these challenges. However, these approaches either constitute comparative studies of different tools or are limited to specific domains of interest. In addition, the main focus of the related literature is the data anonymisation. Our work, based on techniques and tools proposed in the bibliography [3], [6], proposes *a fully-automated process for sensitive data identification and anonymisation* that can be applied both in big diverse datasets and to a wide range of domains.

III. Methodology

Based on the existing literature and taking into consideration the breadth of the entity recognition and data anonymisation research fields, our interest was steered towards developing a NLP-based pipeline, which would automatically identify personal/sensitive information in a given dataset and facilitate its anonymisation. To this end, the Presidio Analyzer[7] along with the Amnesia tool[8] were identified as two suitable candidates to be used in the proposed approach. In what follows, we give a brief overview of the used tools, as well as shed light on the architecture and functionality of the assembled pipeline.

A. Presidio Analyzer

The Presidio analyser is a popular open-source tool and part of the Microsoft Presidio software, which can be used to identify and classify sensitive data, such as Personally Identifiable Information (PII), Protected Health Information (PHI), and financial information. In more detail, the Presidio analyser harnesses a range of techniques, including NER, regex, rule-based logic, blacklisting (removal of specific string terms expressed in sentence case, upper case, and abbreviations to capture all possible combinations), RFC-822 validation (standard for internet text messages), and checksum, aiming to identify *predefined and custom* PII. This tool is designed to provide accurate and efficient identification and anonymisation of sensitive data in a wide range of use cases, including healthcare, finance, and marketing. During analysis, a set of different PII identifiers are utilised, each one in charge of detecting one or more PII entities.

B. Amnesia REST API

Amnesia is a tool developed by the OpenAIRE infrastructure, which enables the full or partial, if instructed, transformation of personal information to anonymous data and guarantees exceptional results in the field of privacy-preserving data publishing. The users may interact with Amnesia through a graphical interface. A Spring REST API is also provided, enabling the programmatic implementation of data anonymisation tasks via HTTP requests to and from the locally running Amnesia server. The lifecycle of a typical Amnesia data obfuscation process includes the following five basic steps:

1) Data import. Amnesia has an import wizard that guesses the type of the imported data and asks the user to confirm

[5]https://github.com/stefanos-vlachos/Presidio-Anonymizer

[6]https://www.kaggle.com/datasets/stefanosvlachos/custom-bookings-dataset

[7]https://github.com/microsoft/presidio

[8]https://github.com/dTsitsigkos/Amnesia

Fig. 1: The proposed pipeline

it. The selected dataset must be in the form of a simple text file using any type of delimiter.

2) Fields selection. The user has to define which fields will be anonymised and which will be left unprocessed.
3) Creation of generalisation hierarchies. The basic idea is to replace unique values or unique combinations of values (e.g., zip code, date of birth) with more abstract ones (e.g., city, year of birth), so that the resulting data will no longer be personally identifiable. Amnesia allows a user to create these rules in a semi-automatic way.
4) Algorithm selection. The user can select the most suitable method for each anonymisation task (e.g., k-anonymity or km-anonymity) and link the created hierarchies with the respective dataset attributes.
5) Solution selection. Amnesia produces several possible anonymisation solutions, visualises the distribution of values, and provides statistics about the safety supplied by each solution.

C. Proposed Pipeline

Taking into account the wide variety of automatically applied PII recognition methods, in contrast to the time and computational restrictions associated with training a deep learning model from scratch, Presidio analyser was considered as a promising option for the analysis stage of the proposed pipeline. In addition, Amnesia was considered as a suitable anonymisation tool since it provides sophisticated anonymisation capabilities. It is worth mentioning that we preferred Amnesia and not the built-in Presidio anonymiser module to encrypt the desired data fields, as the latter handles identified entities individually and not in the context of an attribute. The integration of Presidio analyser and Amnesia REST API in a pipeline was accomplished by using the Python programming language. The utilisation of the REST API provided us with the capability of automating the interaction with Amnesia and dynamically adapting the anonymisation process to the needs of each dataset, instead of manually using the Amnesia GUI.

Taking a deeper look into the functionality of the proposed pipeline, the workflow is divided into the following three successive phases (also shown in Figure 1):

TABLE I: Example instance of the DataFrame storing the metadata of the sensitive entities

attributeName	attributeType	mostFrequentEntityType	percentage
att1	string	PERSON	0.86
att2	string	EMAIL_ADDRESS	1.00
...	...	...	...
attN	date	DATE_TIME	1.00

1 – Analysis. This stage includes the analysis of the selected data file to *automatically* detect sensitive or personal data. The process is implemented iteratively for each property of the given dataset and is facilitated by the BatchAnalyserEngine module[9] of the Presidio analyser. Algorithm 1 provides an abstraction of this process.

Data: CSV Dataset
Result: List of sensitive attributes
Initialization;
for *each attribute in this dataset* **do**
  analyse attribute;
  count the occurrences of each identified entity;
  get the most frequent entity;
  percentange = (occurrences / num_of_lines);
  if *percentage* > 0.5 **then**
    mark attribute as sensitive;
  end
end

Algorithm 1: Pseudocode for the dataset analysis

In more detail, during each iteration, the BatchAnalyserEngine is provided with an attribute and handles each value as an individual text phrase. After the batch analysis is complete, the engine has classified a set of words as PII entities and has replaced them with a more general value (e.g., replace *Greece* with *LOCATION*). Presidio analyser supports a wide range of recognisable PII entities,[10] such as *PERSON*, *DATE_TIME*, *EMAIL_ADDRESS* or *CREDIT_CARD*.

We are measuring the existence of PII entities within an attribute to infer whether it is sensitive or not. More precisely, if a certain PII entity is identified in more than half of the entries (percentage > 0.5), the attribute is labeled as *sensitive* and the related metadata is stored in a Pandas DataFrame; the DataFrame will then be the touchstone of the anonymisation stage. An example instance of the DataFrame is shown in Table I. The selection of 0.5 as the threshold for labeling an attribute as sensitive or not is based on the following observations: (1) if at least 50% of the entries include a certain PII, this data is personal/sensitive with high probability and (2) higher threshold values, combined with the restrained capability of Presidio to identify PIIs, would (wrongly) lead to less attributes being labeled as sensitive. Note that the *attributeType* in Table I serves the process of mapping the different PII entities and Python data types to the recognised

[9]https://microsoft.github.io/presidio/samples/python/batch_processing/
[10]https://microsoft.github.io/presidio/supported_entities/

data types by Amnesia. Algorithm 2 provides the process to define the attribute type.

Data: Attribute data
Result: Attribute type
Initialization;
if *df.dtype of attribute is* object *OR* category *OR* bool **then**
  attribute type is *string*;
end
if *df.dtype of attribute is* int64 **then**
  attribute type is *int*;
end
if *df.dtype of attribute is* float64 **then**
  attribute type is *double*;
end
if *df.dtype of attribute is* datetime64 *AND date format is accepted by Amnesia* **then**
  attribute type is *date*;
else
  attribute type is *string*;
end

Algorithm 2: Pseudocode to define the attribute type

2 – Preparation for anonymisation. Before proceeding to the anonymisation of the given dataset, the automated mechanism deals with the inability of Amnesia to load datasets that contain whitespaces. Here, the given dataset is automatically cleaned off of spaces and is routed towards the anonymisation stage.
3 – Anonymisation. During this stage the preprocessed version of the given dataset is passed to the Amnesia REST API, along with the metadata of the identified sensitive columns (see Table I). An abstraction of the workflow of this process is provided by Algorithm 3.

Data: List of sensitive attributes
Result: Anonymised dataset
Initiate Amnesia service;
Get Session ID;
Load attributes data on Amnesia;
for *each attribute in this list* **do**
  create anonymisation hierarchy for attributes;
  load hierarchy to Amnesia;
  bind hierarchy to the attributes;
  anonymise the attributes;
  merge anonymised attributes with raw dataset;
end
Terminate Amnesia service;

Algorithm 3: Pseudocode for the anonymisation process

A basic prerequisite for the pipeline to be able to anonymise the data is that the user has already installed the Amnesia application locally. In the beginning of the anonymisation process, the mechanism will automatically initiate the Amnesia service, after acquiring the installation path of Amnesia from a *.properties* file. Besides the Amnesia installation path, the properties file also contains a handful of parameters that are of high significance for the proper function of Amnesia:

1) *K* – The k-anonymity factor that refers to the number of times each combination of values appears in a data set
2) *FANOUT* – It refers to the number of children for each hierarchy tree
3) *STRING_ANON_METHOD* – It refers to the way Amnesia handles and anonymises strings; the available options are *distinct* (for replacement of text values by random variables) and *mask* (for masking text values)
4) *MASK_LENGTH* – Used in cases when the masking method is utilised for anomynisation and refers to the number of the mask characters

TABLE II: Hierarchy classes for each attribute type

Attribute type	Hierarchy classes
string	Distinct, Masking
date	Distinct, Range
int	Distinct, Range
double	Distinct, Range

The foundation of data anonymisation in Amnesia is *generalisation hierarchies*. Generalization hierarchies are a set of rules that define how specific values should be substituted by more general ones in the anonymisation process. For example, if there is a single person in a dataset that resides in Greece, then Greece (and the rest of EU countries) will be replaced by EU to create more than k identical values in the "Country of Residence" attribute. The generalisation hierarchy that defines how these replacements take place is provided by the user as input to the algorithm. Amnesia will use the hierarchy to replace specific values with more general ones until the privacy guarantee is reached. Within the proposed approach, it was decided to generate a separate hierarchy for each sensitive attribute; based on the attribute type, different hierarchy classes are available (see Table II).

After producing each hierarchy, it is time to anonymise the corresponding attribute. Despite the fact that the capability of jointly binding hierarchies and anonymising data is provided, we decided to handle each attribute individually, due to the processing overhead of the first approach. At this point, a plethora of anonymisation solutions that guarantee k-anonymity are produced; each solution comes with a safety metric that indicates the efficiency of anonymisation applied on the data. Solutions can be presented graphically as a lattice, where each node represents a different solution (an example is shown in Figure 2). Amnesia uses heuristics to quickly identify a good, but possibly not the overall best, solution. However, in the context of a programmatic approach, such as the present work, the desired solution has to be manually selected. To this end, the selection of the least strict anonymisation solution among the solutions labeled as *safe* was considered the most suitable, so that the interpretability of the dataset is preserved as much as possible. Finally, the anononymised attributes are combined with the unprocessed attributes in a common dataset and is made available to the user. Table III provides examples

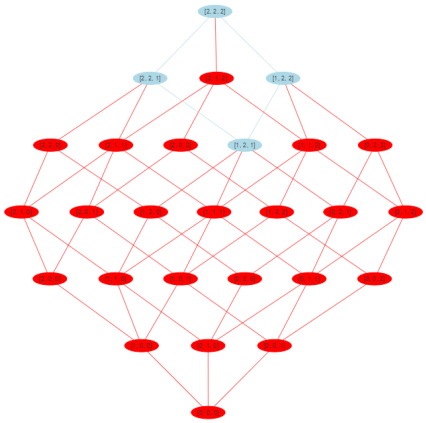

Fig. 2: A solutions tree example: blue nodes satisfy the k-anonymity requirements, while red nodes do not

TABLE III: An anonymised dataset example

Age	Zip Code	Nationality
<30	130**	*
≥40	1485*	*
3*	130**	*

of data for which k-anonymity has been ensured.

It is worth mentioning that each of the above processes is carried out via sending HTTP requests. The automation of the anonymisation process is accomplished by dynamically adjusting the HTTP request payloads based on the needs of each anonymisation task and the properties that have been defined by the user.

IV. Use Case Scenario

In order to obtain a better understanding of the anonymisation pipeline, we provide a use case scenario of the proposed mechanism, shedding light on the characteristics of the utilised dataset and showcasing the produced results.

A. The Dataset

Considering the lack of publicly available datasets that contain personal/sensitive data and for testing the functionality of the proposed pipeline, it was decided to design a *synthetic dataset* that would mimic records of maritime bookings.

We initially, employed custom built web scrapers to retrieve publicly available data concerning Greek harbours, shipping companies and ship names. Additionally, open-source Python libraries, including Faker,[11] were used in order to randomly generate data related to travelers and their bookings, such as gender, name, surname, e-mail address, nationality, age, bank account number, and date of transportation. The produced dataset consists of 24 attributes and includes 100 entries; a sample is presented in Figure 3.

[11]https://faker.readthedocs.io/en/master/

fromHarbor	toHarbor	departureDate	departureTime	arrivalTime
Volos, Greece (VOL)	Pachi, Greece (PME)	2023-02-24	22:48:01	1900-01-02 07:33:01
Kalymnos, Greece (KMI)	DERBENI KORINTHIAS, Greece (730)	2022-08-19	03:22:45	1900-01-01 16:21:45
Kiato, Greece (KIO)	Lefkada (Lefkas), Greece (LEV)	2022-04-04	21:03:58	1900-01-02 04:46:58
Monemvasia, Greece (MON)	Skaloma, Greece (SKM)	2021-09-17	09:40:21	1900-01-01 10:53:21
Assos, Greece (SOS)	Keramoti, Greece (KER)	2020-07-28	17:28:21	1900-01-02 08:52:21

Fig. 3: A sample of the synthetic dataset

B. The Use Case

Initially, the configuration parameters in the *pipeline.config* file are specified. In this scenario, the settings hold as follows:

- *AMNESIA_PATH="..."* – The full path to the installation folder of Amnesia
- *K=3* – The k-anonymity factor is set to 3
- *FANOUT=3* – The number of children for each hierarchy tree is set to 3
- *STRING_ANON_METHOD=mask* – The masking anonymisation technique will be applied on string values
- *MASK_LENGTH=3* – The size of the mask will be 3 characters

After setting up the configuration file, the anonymisation mechanism is initiated by executing the following command:

```
$ python batch_analyzer.py
                    -i /path/to/dataset
```

Afterwards, the synthetic transportation dataset is loaded into Presidio. The Presidio analyser scans the provided dataset to identify sensitive entities, as described in Section III-A. In the present use case, the anonymisation pipeline produced the analysis results showed in Figure 4.

In the results, we observe that the proposed process returned a subset of attributes identified as personal/sensitive, along with (i) the corresponding Amnesia data type, (ii) the dominating Presidio PII in each attribute, and (iii) the percentage of values in which this PII is present. Notice that in the case of the *email* attribute all entries have been identified as sensitive, while the *LOCATION* PII was detected in only 64% of the entries concerning the *fromHarbor* attribute. This stems from the restrained capability of Presidio to identify PIIs among synthetic fields; tackling with this issue is further discussed in Section V. It is also worth noting that not all attributes labeled as *DATE_TIME* are handled as "date" types. This stems from the fact that only a few date formats are acceptable by Amnesia. Consequently, dates that do not conform to the accepted standards are handled as string values. Applying suitable transformations to the dataset at the initial stages of the pipeline could alleviate this issue.

After the analysis of the dataset is complete, the results are passed to the anonymisation phase. In this stage: (1)

columnName	columnType	mostFrequentEntity	percentage
fromHarbor	string	LOCATION	0.64
toHarbor	string	LOCATION	0.68
price	double	US_BANK_NUMBER	0.86
departureDate	date	DATE_TIME	1.00
departureTime	string	DATE_TIME	0.68
arrivalTime	string	DATE_TIME	1.00
birth_date	date	DATE_TIME	1.00
nationality	string	LOCATION	0.92
phone_number	string	PHONE_NUMBER	0.67
firstName	string	PERSON	0.90
lastName	string	PERSON	0.62
email	string	EMAIL_ADDRESS	1.00
cardNumber	string	IBAN_CODE	1.00
creditName	string	PERSON	0.90
creditSurame	string	PERSON	0.62
creditExpiration	string	DATE_TIME	1.00

Fig. 4: Analysis results

fromHarbor	toHarbor	departureDate	departureTime	arrivalTime	ship
***	***	1970-1974	0**	19*	FLYINGCAT 5
***	***	1990-1994	1**	19*	SUPER STAR
***	***	1995-1999	1**	19*	H/S/C SANTORINI PALACE
***	***	2015-2023	1**	19*	FLYINGCAT 6
***	***	2010-2014	2**	19*	HIGH SPEED JET

Fig. 5: A sample of the anonymised dataset

The preprocessed dataset is loaded, (2) For each sensitive attribute a hierarchy is created, (3) Each hierarchy is bound to the corresponding attribute and the available anonymisation solutions are saved into a JSON file, and (4) Each sensitive column is anonymised.

Finally, the fully anonymised dataset is constructed and made available to the user. A sample of the anonymised dataset is provided in Figure 5. The files produced during the anonymisation phase of the given dataset, along with the preprocessed and anonymised versions of the dataset, are stored under the automatically generated *./build* directory in the root folder of the mechanism.

V. Conclusions and Future Work

In this paper, we proposed an automated process for sensitive data identification and anonymisation. Although the proposed pipeline can be applied in big diverse datasets and to a wide range of application domains, it allows for enrichment and improvements. To start with, we intend to evaluate our work using existing tools like SECRETA [8] or comparing it against other anonymisation approaches, e.g. [5], [7].

One potential area of improvement would be the incorporation of more advanced NLP techniques to identify more complex and varied entities and improve the accuracy. The ability to train custom models for entity detection would further enhance the effectiveness of the process, making it more flexible and customisable. Creating also, a centralised datasets repository for entity detection would simplify and streamline the model training process, making it easier for users to develop custom models that are tailored to their specific needs. To achieve the above and also to enhance the accessibility of the process, especially for non-technical users, a intuitive user interface (UI) could be developed; such an interface would simplify the process of selecting pre-built models, uploading datasets, and viewing the results of sensitive data identification and anonymisation.

Additionally, the implementation of distributed processing, for example across multiple machines or cloud instances, could further reduce the time required for sensitive data identification and anonymisation in large datasets, improving the efficiency and effectiveness of the proposed approach. This feature could be particularly beneficial for organisations dealing with large volumes of data or strict time constraints for data processing.

Acknowledgments

This work was supported in part by project ENIRISST+ under grant agreement No. MIS 5047041 from the General Secretary for ERDF & CF, under Operational Programme Competitiveness, Entrepreneurship and Innovation 2014-2020 (EPAnEK) of the Greek Ministry of Economy and Development (co-financed by Greece and the EU through the European Regional Development Fund).

References

[1] (2017) ISO 25237:2017 Health informatics – Pseudonymization. Last reviewed and confirmed in 2023. [Online]. Available: https://www.iso.org/standard/63553.html

[2] D. Mahendran, C. Luo, and B. McInnes, "Review: Privacy-preservation in the context of natural language processing," *IEEE Access*, 2021.

[3] F. Martinelli, F. Marulli, F. Mercaldo, S.Marrone, and A. Santone, "Enhanced privacy and data protection using natural language processing and artificial intelligence," in *IJCNN*, 2020.

[4] J. Mohan and V. Torra, "On the role of data anonymization in machine learning privacy," in *IEEE TrustCom*, 2017.

[5] B. Tripathy, M. Sishodia, S. Jain, and A. Mitra, "Privacy and anonymization in social networks," *Social Networking*, 2014.

[6] J. Tomas, D. Rasteiro, and J. Bernardino, "Data anonymization: An experimental evaluation using open-source tools," *Future Internet*, 2022.

[7] N. Johnson, J. P. Near, J. M. Hellerstein, and D. Song, "Chorus: a programming framework for building scalable differential privacy mechanisms," in *IEEE EuroS&P*, 2020.

[8] G. Poulis, A. Gkoulalas-Divanis, G. Loukides, S. Skiadopoulos, and C. Tryfonopoulos, "Secreta: A system for evaluating and comparing relational and transaction anonymization algorithms," in *EDBT*, 2014.

[9] N. Senavirathne and M. Rao, "Preserving the privacy of sensitive data using data anonymization," *IJAER*, 2020.

[10] S. Sampaio, P. Sousa, C. Martins, A. Fereira, L. Antunes, and R. Correia, "Collecting, processing and secondary using personal and (pseudo)anonymized data in smart cities," *Applied Sciences*, 2023.

[11] (2018) ICO, Guide to Data Protection. [Online]. Available: https://ico.org.uk/for-organisations/guide-to-data-protection/

[12] (2014) EC Article 29 Data Protection Working Party, Opinion 05/2014 on Anonymisation Techniques. [Online]. Available: https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf