Detection of Personal Data in Structured Datasets Using a Large Language Model

Albert Agisha Ntwali
albert.agishantwali@hs-aalen.de
Aalen University of Applied Sciences
Aalen, Germnany

Luca Rück
luca.rueck@studmail.htw-aalen.de
Aalen University of Applied Sciences
Aalen, Germany

Martin Heckmann
martin.heckmann@hs-aalen.de
Aalen University of Applied Sciences
Aalen, Germany

Abstract

We propose a novel approach for detecting personal data in structured datasets, leveraging GPT-4o, a state-of-the-art Large Language Model. A key innovation of our method is the incorporation of contextual information: in addition to a feature's name and values, we utilize information from other feature names within the dataset as well as the dataset description. We compare our approach to alternative methods, including Microsoft Presidio and CASSED, evaluating them on multiple datasets: DeSSI, a large synthetic dataset, datasets we collected from Kaggle and OpenML as well as MIMIC-Demo-Ext, a real-world dataset containing patient information from critical care units.

Our findings reveal that detection performance varies significantly depending on the dataset used for evaluation. CASSED excels on DeSSI, the dataset on which it was trained. Performance on the medical dataset MIMIC-Demo-Ext is comparable across all models, with our GPT-4o-based approach clearly outperforming the others. Notably, personal data detection in the Kaggle and OpenML datasets appears to benefit from contextual information. This is evidenced by the poor performance of CASSED and Presidio (both of which do not utilize the context of the dataset) compared to the strong results of our GPT-4o-based approach.

We conclude that further progress in this field would greatly benefit from the availability of more real-world datasets containing personal information.

CCS Concepts

• **Computing methodologies** → **Machine learning**; **Information extraction**; • **Information systems** → *Data management systems.*

Keywords

Personal Data Detection, Large Language Models, GPT-4o, Data Privacy, GDPR Compliance, Contextual Analysis, Structured Data, Machine Learning, Information Retrieval, Entity Recognition, Data Management, Benchmarking

LLM-DPM '2025, Berlin, Germany

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Albert Agisha Ntwali, Luca Rück, and Martin Heckmann. 2025. Detection of Personal Data in Structured Datasets Using a Large Language Model. In *Proceedings of Next Gen Data and Process Management: Large Language Models and Beyond (LLM-DPM '2025).* ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Recent years have seen a strong increase in the amount of data created and stored in a digital form. In many cases also personal data is collected. However, this vast accumulation of data and its digital availability brings with it substantial challenges, particularly concerning compliance with data protection regulations [2, 5, 10].

At the heart of these regulations is the General Data Protection Regulation (GDPR) of the European Union, which is widely considered the gold standard for data privacy [5, 10]. The GDPR not only aims to protect individuals' fundamental rights regarding the processing of their data but also imposes strict penalties on organizations that fail to comply, with potential fines reaching up to 4% of global annual turnover [21]. The significance of the GDPR has led to its adoption as a foundational reference for personal data protection worldwide, influencing similar regulations in various countries [12, 24].

To comply with these regulations and for ethical reasons, organizations need to implement effective measures to detect and manage personal data. In light of the often large volume of data, this requires powerful automated detection tools [9, 22].

Problem Statement. Detecting personal data within structured datasets poses unique challenges. The format of the document being analyzed significantly influences the effectiveness of personal data detection efforts. The GDPR highlights the importance of context in defining personal data, indicating that an individual can often be identified indirectly through a combination of seemingly innocuous information. For example, while an individual's age alone may not be identifying, when combined with other attributes, such as their organization or job title, it can lead to their identification[10]. Existing solutions tend not to integrate information across several columns and thereby neglect important information [19]. For instance, a "Device Number" column in an IT asset database typically represents a device's serial number and may not initially appear to contain personal information. However, if the database is linked to an employee database that tracks assigned hardware, the device number could serve as an identifier for an employee, making it personal data. Without proper contextual comprehension, these solutions misclassify data and result in compliance issues and privacy violations.

Objective of the Study. We aim to develop an effective system for detecting personal data in structured datasets by integrating contextual information into the detection process. We propose a novel approach that employs large language models (LLMs), specifically the GPT-4o model, to improve detection accuracy. By comparing this model with established benchmarks, including Microsoft Presidio and CASSED models, we seek to highlight the advantages of context-aware detection methods.

Furthermore, we target an evaluation of these models on realistic real-world data. Previously used synthetic datasets bear the risk of not reflecting the real challenges properly and, hence, leading to wrong conclusions. For this reason, we collected a total of 33 datasets from Kaggle and OpenML and also included MIMIC-Demo-Ext, a real-world dataset containing patient information from critical care units, in our evaluation.

Contributions of this Paper. To summarize, the contributions of this paper are as follows:

(1) **Novel Methodology for Personal Data Detection:** We introduce a novel approach that leverages the capabilities of Large Language Models (LLMs), specifically GPT-4o, to enhance the detection of personal data in structured datasets. By integrating contextual information into the detection process, we aim to improve accuracy and adaptability compared to traditional methods.
(2) **Comprehensive Benchmarking:** We compare GPT-4o with established models like Microsoft Presidio and CASSED. The evaluation utilizes the DeSSI, Kaggle, OpenML, and MIMIC-Demo-Ext datasets to evaluate the detection of personal data. Comparing the performance across diverse datasets allows a better assessment of the strengths and limitations of each detection system.
(3) **Real-world vs. Synthetic Data:** A key element of our evaluation is the comparison of results obtained on synthetic vs. real-world data. With this, we aim to determine if synthetic data can serve as a reliable benchmark.

2 Related Work

The field of personal data detection has gained increasing attention, especially in light of evolving privacy regulations. However, before moving forward, it is essential to define what we mean by *personal data*.

2.1 Definition of Personal Data

The concept of personal data includes several related terms. The most commonly used are:

(1) Personal Data: According to Article 4 of the GDPR, personal data is defined as "any information relating to an identified or identifiable natural person" [8]. Examples include information about a person's physical properties, contact details or identification numbers. Personal data can be divided into two categories: directly identifiable information and indirectly identifiable information.
(2) Personally Identifiable Information (PII): PII is defined as any information that can directly identify a person without the need for additional information, such as a bank account number or an email address[17].
(3) Person-related Data: Person-related data cannot directly identify a person but relates to a natural person and may lead to identification when combined with other data [10]. Information such as gender or age does not directly identify a person but can be used in combination with other data for identification.

When we refer to personal data, we use the definition of the GDPR (1) as a supergroup of PII and person-related data. We specifically used this definition when we labeled data as personal or non-personal in the datasets we collected. Some authors also use the term *Sensitive Data* synonymously with personal data (e.g. [19]) even though the GDPR defines it as a subgroup of personal data[5].

2.2 Detection of Personal Data from Unstructured Data

Most approaches focus on the detection of personal data in unstructured data (e.g. text) [17, 27, 29].

Initial approaches for personal data detection in unstructured data primarily utilized traditional named entity recognition (NER) techniques, which were adapted from broader natural language processing (NLP) applications, including sensitive information detection and PII detection [25, 29]. Very recent approaches also use Large Language Models (LLMs) for this task [29].

2.3 Detection of Personal Data from Structured Data

Research on personal data detection from structured data is much more scarce. One prominent example is Microsoft Presidio[22] which was designed for general entity recognition tasks yet applied to PII detection. It is based on a set of recognizers for predefined classes [23] (compare Tab:8). Presidio employs a combination of approaches including NER, regular expressions, rule-based logic, checksums, and machine learning to identify sensitive data. Presidio also offers options for connecting to external PII detection models and supports customization in PII identification and anonymization. While effective in many scenarios, these approaches can be less effective in handling context-dependent variations of personal data [20].

More recent models use advanced machine learning methods to integrate information from a column's name with the values of different cells in this column. Examples are Sherlock [13], which employs deep neural networks, and SIMON [1], which relies on a character-level neural network and an LSTM architecture.

The advent of language models like BERT [4] allowed for more robust approaches to personal data detection. BERT transforms sequences of text into vector embeddings which are able to capture the semantics of the text much better than previous approaches. As a consequence, variations in a feature's name or its values have a much less detrimental effect. The CASSED model is based on such an approach [19]. More precisely, it employs DistilBERT [26], a lightweight variant of BERT, to convert a feature's name and some exemplar feature values into an embedding. For doing so, the text sequence of the feature's name and the values of the feature are concatenated into a string, separated by delimiters, allowing

DistilBERT to treat the column as a quasi-natural sentence. The maximum token length of 512 elements of DistilBERT limited the amount of information that could be used. For this reason, CASSED is not able to include additional information from the neighboring columns. In parallel to this DistilBERT path, a rule-based path is used to identify personal data. Both paths are then combined via sigmoid functions to convert the scores for all 20 different classes of personal and non-personal data categories CASSED can recognize into probabilities (compare Tab: 7 for a list of the classes). These probabilities are then used to make a decision. CASSED was trained using DeSSI [3], a large dataset annotated with the aforementioned 20 classes.

3 Methodology

In the following, we will first introduce the different datasets we use to evaluate the different approaches. Next, we will describe our GPT-4o-based approach in detail. We will then explain which models we used as benchmarks for our own approach and how they needed to be adapted to be suitable for this comparison. After that, we will also describe the performance metrics we used to benchmark the models.

Our implementation is open-source and publicly available on GitHub[1].

3.1 Dataset Selection

Our evaluation relies on a large range of datasets containing personal information. The first and by far largest dataset we use is the DeSSI dataset (Dataset for Structured Sensitive Information) [3, 19], created to simulate real-world relational database challenges. The authors give no exact definition what they consider *sensitive information*, yet, they reference the GDPR in their work and the list of classes they use (compare Tab: 7) aligns with the definition of personal data of the GDPR [19]. From this, we conclude that they also use the definition of the GDPR as we do. DeSSI consists of over 31,000 database columns with 100 rows each, derived from open-source datasets (e.g., Kaggle), synthetic data generated via Python libraries like Faker, and pseudo-anonymized real-world data[6, 19]. Columns were intentionally designed with randomized or misleading headers to reflect real-world inconsistencies and avoid reliance on misleading column headers. The dataset was randomly split in ratios of 60/20/20 percent among training/validation/test datasets. For training the CASSED model, we use the training and validation part, and for the evaluation of CASSED, Presidio, and our GPT-4o-based approach, we only use the 6272 columns of the test set. For our experiments, we mapped the original 20 semantic classes into two categories: personal and non-personal data (compare Tab: 7 in the appendix). Additionally, we extracted 13 datasets from Kaggle (finance/e-commerce) [16] and 20 from OpenML [7, 28] (Table 9). Finally, we also include MIMIC-Demo-Ext, a curated subset of the MIMIC-III Demo [11, 14, 15]. MIMIC-Demo-Ext contains information from the MIMIC-III Clinical Database Demo[2], which is made available under the Open Database License (ODbL)[3]. MIMIC-III demo comprises deidentified medical records of over 40,000 ICU patients at Beth Israel Deaconess Medical Center (2001–2012). From these, we extracted and curated a small subset of 100 patients' records to focus on the detection of personal data while preserving the relational nature of the MIMIC-III demo. The dataset ensures that the columns remain contextually linked through identifiers such as patient IDs or admission IDs. Additionally, we ensured that each column contains at least some non-empty values across its records, avoiding fully empty columns. This guarantees that every column contributes meaningful information for personal data detection without compromising the integrity or usability of the relational database schema. The Kaggle, OpenML, and MIMIC-Demo-Ext datasets were not annotated for the detection of personal data. For this reason, one of the authors performed a manual binary labeling (personal/non-personal) based on the definition of personal data of the GDPR and the data context.

Table 1: Statistics of the datasets used

Dataset	Personal	Non-Personal	Total
DeSSI (test set)	3413	2859	6272
Kaggle	155	91	246
OpenML	82	176	258
MIMIC-Demo-Ext	43	120	163

As can be seen from Tab. 1 the number of features in the different datasets and the ratio of personal to non-personal features varies a lot from dataset to dataset. DeSSI is very balanced wrt. to personal vs. non-personal features and contains almost ten times as many features as the other datasets taken together. MIMIC-Demo-Ext on the other hand is the smallest dataset and dominated by non-personal features. However, MIMIC-Demo-Ext is the only actual real-world dataset. DeSSI is mainly synthetic and for some of the datasets on Kaggle and OpenML it is not clear if they are truly authentic datasets or only inspired by real data.

3.2 Experimental Procedure GPT-4o

For our GPT-4o-based approach, we integrate information on the column in question with information from all other columns in the dataset (see Sec. A). This is in contrast to e.g. CASSED which evaluates each column independently from all other columns. Input prompts for our approach are structured to include the following information :

- Title of the dataset
- Description of the dataset
- Column Name (feature to be classified)
- Names of other features of the dataset
- Ten most frequent values found in the column

This structured input allows GPT-4o to focus on the immediate context relevant to the column being analyzed, thereby reducing potential overload from extraneous data. The output of each column is a binary classification indicating whether it contains personal data ($True$) or not ($False$).

3.2.1 CRSRF Framework. The CRSRF (Capacity and Role, Statement, Reason, Format)[29], detailed in appendix B, is designed to enhance the effectiveness of prompt-based classification tasks in

[1]The implementation can be accessed at: https://github.com/agishaalbert/personal-data-detection-LLMs/
[2]https://physionet.org/content/mimiciii-demo/1.4/
[3]https://physionet.org/content/mimiciii-demo/view-license/1.4/

machine learning models, particularly in identifying semantic relationships within datasets. This framework emphasizes the necessity of clarifying the model's role in the classification process, articulating specific objectives, and outlining expected output formats. Using this structured approach, the model can better navigate and understand complex data inputs, leading to improved classification accuracy.

3.2.2 Prompting Structure. The design of the prompt is crucial, as it significantly impacts the performance of the model. The prompt is structured into three main components:

(1) **Initial Prompt**: This component introduces the task to GPT according to the CRSRF framework. It emphasizes the importance of the task and outlines how the results should be outputted.
(2) **Example Prompt**: This prompt consists of an example question and answer, providing a single instance of the task (one-shot learning) to demonstrate the expected output format.
(3) **Data Prompt**: This component contains the specific column to be classified, along with meta-information regarding the dataset, such as the title and description.

The complete structure of the prompt provided to the GPT API is illustrated as follows:

```
conversation = [
    {"role": "system", "content": initial_prompt},
    {"role": "user", "content": example_prompt},
    {"role": "assistant", "content": example_answer},
    {"role": "user", "content": data_prompt}
    ]
```

In this structure, roles are defined to guide the model regarding the following entities:

- The `"system"` provides task instructions to the model.
- The `"user"` reflects input data that necessitates classification by the GPT model.
- The `"assistant"` projects the anticipated model response.

An example of a final prompt is presented in Sec. A.

For each part of the prompt, a random seed is set; however, it is essential to note that this does not guarantee reproducibility of the responses of the GPT-4o model.

3.3 Benchmark Model Selection

We selected Presidio [22] and CASSED [19] as benchmark models against which we compare our GPT-4o-based approach. We selected CASSED because it is a recent model that outperformed Sherlock and SIMON in a previous comparison [19]. We also selected Presidio as a baseline due to its frequent use and in general good performance. We did not include Sherlock and SIMON as they showed significantly weaker performance than the CASSED model in the aforementioned comparison.

To be usable in our experiments, we needed to make some adjustments to the models.

3.3.1 Adjustments Presidio. Presidio contains different modules. The Presidio Analyzer module can detect PII information in textual documents, while Presidio Structured is designed to recognize PII data in tabular datasets. We tested both modules using different approaches. For every dataset, the Presidio Analyzer outperformed the Presidio Structured module. Consequently, we only present results for the Presidio Analyzer module.

For the Presidio Analyzer module, tabular datasets must be converted into textual data. We tested two strategies:

- Columnwise, where all values from a single column, along with the column name, are provided to the model.
- Rowwise, where all values from a single row are transmitted together.

Presidio then predicts all detectable entities for each column or row. Next, the predicted entities are mapped to personal or non-personal (see Sec. C.2).

To optimize the prediction accuracy for the Presidio Analyzer module, two thresholds are implemented. The first threshold defines the minimum number of times an entity must be detected in a column to be considered valid. The second threshold is a minimum for the confidence score of the entity, which indicates how confident Presidio is that the entity is detected correctly. As some entities that Presidio can detect are not necessarily related to a person, the predictions have to be mapped to personal and non-personal. For each dataset, the best presidio analyzer approach was used for comparing Presidio's performance against the other models in the experiments.

3.3.2 Adjustments to CASSED. The CASSED model was used with the settings described in the original work [19]. Predictions are made for each column of a dataset using a column-wise approach. The input to the model is constructed for each column by combining the column header with multiple cell values from the same column, separated by delimiters('.', ',')[19].

CASSED originally is able to detect 20 different classes. To adapt CASSED from multiclass to binary classification, we modified and retrained the model by mapping the original multiclass labels to binary labels using the label mapping described in (Sec. C.1). We trained CASSED using the train and validation split of the DeSSI dataset. Afterward, the model was evaluated on the test set of DeSSI and all other datasets (which served only as test data). The Adam optimizer was used for fine-tuning with a learning rate of 5×10^{-5}. The model was trained with a mini-batch size of 16 for 20 epochs, following the procedure outlined in [19]. In the original work of the CASSED model, the output of the DistilBERT model was combined with some rule-based heuristics, regular expressions, and lookup tables [19]. The publicly available CASSED model published on Github [18] does not contain these enhancements, so only the Transformer model was used in this work. In the results of CASSED's original paper[19], the public model is only slightly worse than the enhanced version. Consequently, the use of the publicly available model instead of the best implementation should not result in significant performance loss.

3.4 Evaluation Metrics

In assessing the models' detection capabilities, we utilize the following metrics:

- **Macro F1 Score**: it assigns equal weight to all classes regardless of frequency by computing the F1 score for each class separately before computing the unweighted mean. As it assigns equal weight to all classes, including the less frequent ones, it is particularly useful when assessing model performance on class-imbalanced datasets.
- **Micro F1 Score:** it calculates the F1 score globally by adding up the total number of true positives, false positives, and false negatives across all the classes before precision and recall calculation. It uses one score to measure the model's performance over all instances combined. Micro F1 is dominated by majority class performance in class-imbalanced data and therefore doesn't work well for evaluating performance in minority classes.
- **Balanced Accuracy:** It computes the average of recall for all classes, ensuring that performance is fairly assessed even in cases of class imbalance. This metric provides a better estimate when certain categories occur less frequently in the test set.

The models are evaluated using macro, micro F1 score, and Balanced Accuracy–valuable metrics for assessing a model's performance in multiclass classification problems, allowing a balanced view of precision and recall across classes.

4 Experimental Results

In this section, we compare the performance of Presidio, CASSED, and our GPT-4o-based approach on the aforementioned data sets.

4.1 F1 Scores and Balanced Accuracy

Our main target is the binary distinction between personal and non-personal data. To achieve this, we adapted CASSED to such a binary classification task (see Sec. 3.3). In the case of Presidio, we performed a multiclass classification and then mapped the detected classes to either personal or non-personal categories (see Sec. C.2).

Table 2: Performance comparison of Microsoft Presidio, CASSED, and GPT-4o across different datasets.

Dataset	Metric	Presidio	CASSED	GPT-4o
DeSSI	Macro F1	0.793	**0.996**	0.766
	Micro F1	0.794	**0.996**	0.772
	Balanced Acc.	0.791	**0.996**	0.764
Kaggle	Macro F1	0.293	0.349	**0.902**
	Micro F1	0.297	0.386	**0.907**
	Balanced Acc.	0.299	0.481	**0.910**
OpenML	Macro F1	0.684	0.501	**0.964**
	Micro F1	0.733	0.686	**0.969**
	Balanced Acc.	0.518	0.535	**0.968**
MIMIC-Demo-Ext	Macro F1	0.662	0.724	**0.829**
	Micro F1	0.730	0.798	**0.859**
	Balanced Acc.	0.667	0.713	**0.852**
Average	**Macro F1**	0.608	0.643	**0.865**
	Micro F1	0.639	0.717	**0.877**
	Balanced Acc.	0.569	0.681	**0.874**

As can be seen from Tab. 2, on the DeSSI dataset, CASSED performed nearly perfectly, with Macro F1, Micro F1, and Balanced Accuracy all reaching 0.996. Microsoft Presidio followed with a Macro F1 of 0.794, while our GPT-4o-based approach performed slightly lower at 0.766.

Across the Kaggle and OpenML datasets, GPT-4o achieved the highest performance, with a Macro F1 score of 0.902 on Kaggle and 0.964 on OpenML. The performance of CASSED dropped drastically for these two datasets to Macro F1 scores of 0.349 and 0.501, respectively. Similarly, Presidio also only achieved Macro F1 scores of 0.293 and 0.684 on these datasets. The differences in performance were less pronounced for the MIMIC-Demo-Ext dataset. Here again, our GPT-4o-based approach leads with a Macro F1 score of 0.865. CASSED and Presidio achieve 0.724 and 0.662, respectively. Hence, CASSED shows clearly superior performance on DeSSI, a dataset on which it was developed. For all other datasets, our GPT-4o-based approach is better. Presidio shows comparable performance to CASSED except for the evaluation on DeSSI. This is also visible when averaging scores over all datasets. Here, our GPT-4o-based approach shows with an averaged Macro F1 score of 0.865 clearly superior performance to CASSED (0.643) and Presidio (0.608). The other measures (Micro F1 and Balanced Accuracy) show a similar behavior.

4.2 Analysis of False Negatives and False Positives

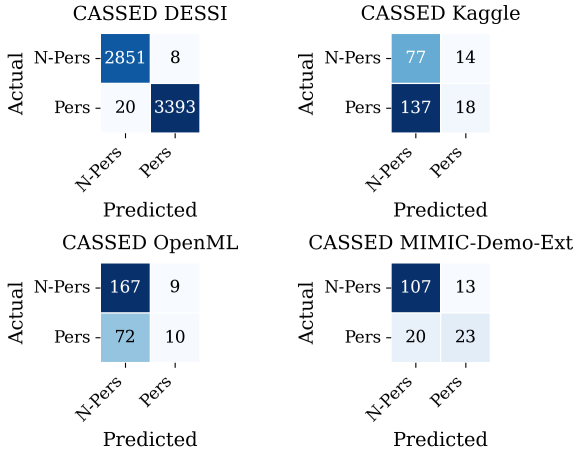

Figure 1: Confusion matrices for CASSED model performance across datasets: DESSI, Kaggle, OpenML, and MIMIC-Demo-Ext. Each matrix shows the classification results for personal (P) and non-personal (N-pers) data categories.

A model's ability to prevent false negatives (FN) has high practical relevance, as failing to detect personal data can pose serious GDPR compliance risks. When looking at the confusion matrices for all three approaches and all 4 datasets (Fig. 1, 2, and 3) one can observe that CASSED performs near perfect on DeSSI yet shows frequent false negatives (personal-related data not detected as such) for the Kaggle and OpenML datasets. This is also visible for the MIMIC-Demo-Ext dataset but less pronounced. The performance of our GPT-4o-based approach is more balanced for all datasets other

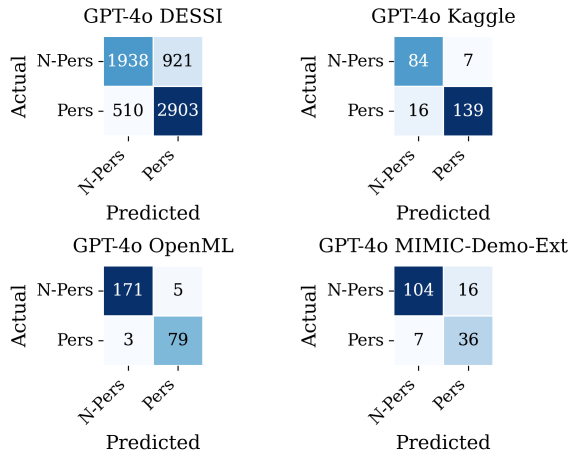

Figure 2: Confusion matrices for our GPT-4o-based model performance across four datasets: DESSI, Kaggle, OpenML, and MIMIC-Demo-Ext. Each matrix shows the classification results for personal (P) and non-personal (N-pers) data categories.

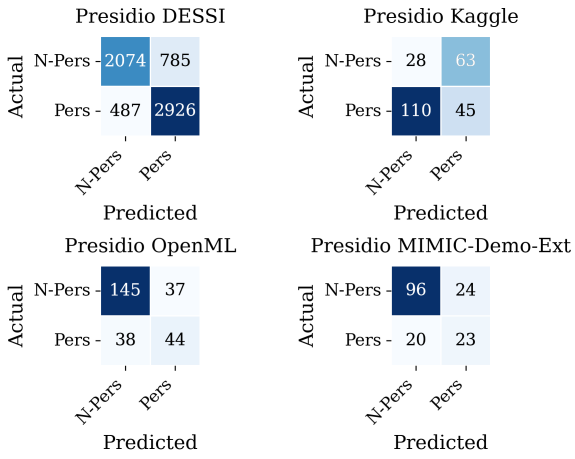

Figure 3: Confusion matrices for Presidio model performance across datasets: DESSI, Kaggle, OpenML, and MIMIC-Demo-Ext. Each matrix shows the classification results for personal (P) and non-personal (N-pers) data categories.

than DeSSI. For DeSSI it shows a notable tendency for false positives (non-person-related detected as person-related). The performance of Presidio is the most balanced yet overall inferior.

Hence, the risk of false negatives is notably smaller for our GPT-4o-based approach compared to CASSED and Presidio for the real-world datasets (Kaggle, OpenML, MIMIC-Demo-Ext) but clearly inferior to CASSED on the synthetic data (DeSSI). Nevertheless, in light of the costs potentially involved, the FN rates on the real-world data are currently still too high for practical applications.

Table 3: Examples in DeSSI Data

Features	oib.1	accountid.3	kzwwskxzgjgc
Values	ZZ 563634 T	Jasper	g-h@hotmail.com
	ZZ 140837 T	vocativ	s.draksic0@ribaric.com
	ZZ 88 70 27 T	kinzo berlin	g-dominguez@wu.net
Context	Synthetic data	Synthetic data	Synthetic data
True Label	Personal	Personal	Personal
CASSED	Personal	Personal	Personal
GPT-4o	Non-personal	Non-personal	Non-personal

Table 4: Examples in Kaggle Data

Features	Cabin	Ticket	Reason Absence
Values	C103	A/5 21171	26
	C123	STON/O2,3101	0
	E46	374910	19
Context	Titanic data	Titanic data	Absenteeism
True Label	Personal	Personal	Personal
CASSED	Non-personal	Non-personal	Non-personal
GPT-4o	Personal	Personal	Personal

Table 5: Examples in OpenML Data

Features	Email Address	Location	Customer City
Values	alexandra@example.org	Rebeccachester	sao bernardo
	holland@example.com	sao paolo	niteroi
	elizabeth31@example.net	Port Deborah	campinas
Context	Customer data	Customer data	Customer data
True Label	Personal	Personal	Personal
CASSED	Non-personal	Non-personal	Non-personal
GPT-4o	Personal	Personal	Personal

Table 6: Examples in MIMIC-Demo-Ext Data

Features	marital_status	discharge_location	last_careunit
Values	MARRIED	HOME	MICU
	DIVORCED	SNF	CCU
	SEPARATED	DEAD/EXPIRED	TSICU
Context	Medical data	Medical data	Medical data
True Label	Personal	Personal	Personal
CASSED	Non-personal	Non-personal	Non-personal
GPT-4o	Personal	Personal	Personal

4.3 Analysis of Selected Examples

As we could see above, the performance of the different models varied significantly depending on the dataset. We will now have a closer look at detailed results for individual features in the different

datasets. As Presidio showed inferior performance we limit this analysis to CASSED and our GPT-4o-based approach.

Tab. 3 shows some examples from the DeSSI dataset which CASSED successfully recognized as personal and our GPT-4o-based approach failed. CASSED seems to have learned the corresponding relations quite well and is in the case of the last feature able to detect the e-mail addresses. Our GPT-4o-based approach seems to be confused by the rather uninformative feature name. CASSED seems also to know (via BERT or have learned from other examples) that "OIB" is a permanent national identification number of every Croatian citizen and correctly identifies it as personal. Again, our GPT-4o-based approach struggles here, presumably because the names of the other features in the dataset - contextual cues our approach integrates - do not give enough hints or because of insufficient representation of Croatia in the training data of GPT-4o. The situation for "accountid" is similar.

When looking at the examples for the Kaggle data in Tab. 4 we see that our GPT-4o-based approach was able to use the context given by the dataset description and the names of the other features to correctly identify the personal information. Here CASSED struggled as some context is required to infer that "Cabin", "Ticket" and "Reason Absence" might reveal personal information. Most likely for similar reasons CASSED also misclassified domain-specific attributes such as "Workclass" and "Income".

The examples where CASSED failed in the OpenML data in Tab. 5 are a bit surprising. "Email Address" and "Customer City" should be identifiable as personal also without context. Possibly a lack of variation in the synthetically created training data of CASSED prevents it from detecting e-mail addresses containing "example" in the domain. It is also a possibility that these types of e-mail addresses were explicitly labeled as non-personal in the training data. CASSED also missed crucial identifiers like "User ID". Conversely, it produced false positives, incorrectly flagging anonymized "Address" and "Postcode" fields as sensitive, and mislabeling generic terms like "Referee" and "Species" as personal data. These errors highlight CASSED's problems in dealing with domain-specific scenarios where contextual interpretation is crucial for accurate classification. Yet our GPT-4o-based approach did also not perform perfectly on the OpenML data. In some instances, it misclassified fields like "userid" and "customer_id.13" as non-personal, likely due to their generic naming conventions.

The mistakes of CASSED we see on the MIMIC-Demo-Ext dataset (compare Tab. 6) might be due to its special nature: medical data. It is possible that medical data was not represented sufficiently in the training data of CASSED. Nevertheless, it is surprising that it did not recognize a feature as "marital_status" correctly. This hints at other reasons than the unfamiliarity with medical data for its poor performance.

5 Discussion

The results showed a very strong performance of CASSED on the DeSSI dataset (compare Tab. 2). On the other hand, CASSED's performance decreased notably when evaluated on the MIMIC-Demo-Ext dataset, and it performed rather poorly on Kaggle and OpenML. Based on our current analysis it is difficult to determine the reasons for this. One possible explanation is overfitting of CASSED on DeSSI, the data set that was developed in conjunction with CASSED.

This could be due to DeSSI representing only a narrow subset of the true variation in personal data or a consequence of the train/dev/test split. Notably, the authors do not specify whether precautions were taken to ensure that features from the same dataset were not distributed across different splits. If this was not accounted for, the system might have leveraged information from the training set when processing the test set, as these features cannot be assumed to be entirely independent. It could also stem from the synthetic generation of features, where the same underlying patterns may have been used for features appearing in both the training and test splits. Another possible explanation is that the individual features in the dataset are largely independent, making them highly compatible with CASSED's core assumptions. Based on our more detailed analysis of some examples in Tab: 4-6 we suspect at least some overfitting of CASSED on DeSSI. Otherwise, it is hard to explain why it would have difficulties recognizing "Email Address", "Customer City" and "marital_status" correctly. Another possibility could be a misalignment in the annotations we used and those used for DeSSI. However, when looking at their classes and our mapping (compare Tab: 7) it is unlikely that this explains the results.

The poor performance of CASSED and Presidio on the Kaggle and OpenML datasets could also be explained by the contextual information required to deal with these datasets. If features like "Cabin", "Ticket", and "Location" contain personal information, it depends, in general, on context. This helps explain the superior performance of our GPT-4o-based approach on these datasets. Another reason might be that these well-known and widely distributed datasets were contained in GPT-4o's training data. This might give GPT-4o an advantage even though they are, to our knowledge, not available with annotations for the detection of personal data (this annotation is our own). Finally, on the MIMIC-Demo-Ext dataset, the differences in performance between the different models were less striking. Here our GPT-4o-based approach obtained inferior results to those on Kaggle and OpenML. Nevertheless, it quite clearly outperformed CASSED and Presidio. One reason for this could be that medical data was not well represented in DeSSI, CASSED's training data. This would be rather unfortunate as personal data detection is a very important topic in the medical domain. GPT-4o, with its vast amount of training data, might hence be better able to cope with this. Another reason might be the benefits of contextual information for this dataset. From the examples we analyzed in Tab: 6 it is difficult to make conclusions about this.

Additionally, the datasets we investigated can also be split across another dimension: real-world (Kaggle, OpenML, MIMIC-Demo-Ext) vs. synthetic (DeSSI). Here, the conclusion might be that the DeSSI dataset does not represent the real world sufficiently well, which leads to the very clear drop in the performance of CASSED when applied to real-world data. However, when looking at the results, it has to be kept in mind that DeSSI alone contains roughly 10 times as many features as the other datasets taken together. More thorough conclusions will require the analysis of more and larger real-world datasets. However, the requirements for data protection complicate the access to such datasets.

5.1 Limitations and Future Research

We demonstrated that LLMs, particularly when incorporating context, can enhance the detection of personal data, particularly in real-world datasets. However, current performance is still insufficient to minimize the risk of personal data disclosure and ensure full GDPR compliance. As main avenues for further improvement, we see:

- Dataset Diversity: The study has highlighted some potential limitations of current methods resulting from the use of synthetic data. It seems likely that DeSSI, the only large-scale dataset for personal data detection, is not diverse enough to cover real-world variations. Additional large and diverse real-world datasets are needed to make further progress.
- Influence of context: We could give some hints that the use of contextual information in our GPT-4o-based model was beneficial yet a more detailed analysis is needed to better assess what role context plays and how it can be most effectively used.
- Privacy Constraints: Since GPT-4o operates as an online service, its reliance on cloud-based processing raises significant privacy concerns. Transmitting personal data to external servers poses compliance risks under data protection regulations. Future research needs to explore secure on-premise solutions to overcome this.
- Computational Demand: GPT-4o is a very powerful yet also very computationally demanding model requiring orders of magnitude more computational resources than Presidio or the BERT-based CASSED. Hence, in addition to the need to find on-premise solutions also much smaller LLMs need to be investigated.
- Hybrid models: It can be expected that integrating ideas from all three models (rule-based approaches, classical machine learning, and LLMs) will help to further improve results.
- False Negatives: Concerning the high priority of not accidentally revealing personal information, adaptations to the models need to be made to better control false negatives.

6 Conclusion

Despite significant performance differences across datasets, we conclude that our GPT-4o-based approach is the most effective model for detecting personal data in structured datasets. It demonstrated strong performance on synthetic data in DeSSI, outperformed CASSED and Presidio on medical data in MIMIC-Demo-Ext, and clearly surpassed both approaches on Kaggle and OpenML—likely due to its ability to leverage contextual information.

However, this high performance comes at the cost of substantial computational demands. Future research should explore whether smaller, locally running LLMs can achieve comparable results with lower resource requirements.

Additionally, further investigation is needed to determine whether CASSED's exceptional performance on DeSSI reflects genuine model capabilities or is merely an artifact of overfitting to this artificial dataset. A more comprehensive analysis, along with additional real-world datasets, will be crucial for a robust evaluation of these models.

Overall, we have demonstrated that high-performance personal data detection in structured datasets is achievable. However, further advancements are necessary to minimize the risk of unintended data exposure and ensure these methods meet acceptable privacy standards.

Acknowledgments

This work was conducted in the context of the project KI-Allianz BW: Datenplattform funded by Ministerium für Wirtschaft, Arbeit und Tourismus Baden-Württemberg.

References

[1] Paul Azunre, Craig Corcoran, Numa Dhamani, Jeffrey Gleason, Garrett Honke, David Sullivan, Rebecca Ruppel, Sandeep Verma, and Jonathon Morgan. 2019. Semantic Classification of Tabular Datasets via Character-Level Convolutional Neural Networks. doi:10.48550/arXiv.1901.08456

[2] California Attorney General. [n. d.]. California Consumer Privacy Act (CCPA). https://oag.ca.gov/privacy/ccpa Accessed: January 26, 2025.

[3] Sensitive Detection. 2022. DeSSI Dataset for Structured Sensitive Information. https://www.kaggle.com/datasets/sensitivedetection/dessi-dataset-for-structured-sensitive-information. Accessed: December 14, 2024.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. doi:10.18653/v1/N19-1423

[5] European Union. [n. d.]. General Data Protection Regulation (GDPR). https://gdpr.eu/ Accessed: January 26, 2025.

[6] Faker Community. [n. d.]. Welcome to Faker's Documentation! https://faker.readthedocs.io/ Accessed: January 26, 2025.

[7] Matthias Feurer, Jan N. van Rijn, Arlind Kadra, Pieter Gijsbers, Neeratyoy Mallik, Sahithya Ravi, Andreas Müller, Joaquin Vanschoren, and Frank Hutter. 2021. OpenML-Python: an extensible Python API for OpenML. *Journal of Machine Learning Research* 22, 100 (2021), 1–5. http://jmlr.org/papers/v22/19-920.html

[8] Michèle Finck and Frank Pallas. 2020. They who must not be identified—distinguishing personal from non-personal data under the GDPR. *International Data Privacy Law* 10, 1 (03 2020), 11–36. doi:10.1093/idpl/ipz026

[9] Somchart Fugkeaw, Ananya Chaturasrivilai, Pitchayapa Tasungnoen, and Weerapat Techaudomthaworn. 2021. AP2I: Adaptive PII Scanning and Consent Discovery System. In *2021 13th International Conference on Knowledge and Smart Technology (KST)* (2021). 231–236. doi:10.1109/KST51265.2021.9415803

[10] GDPR EU. [n. d.]. GDPR personal data – what information does this cover? https://www.gdpreu.org/the-regulation/key-concepts/personal-data/ Accessed: January 26, 2025.

[11] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet. *Circulation* 101, 23 (2000), e215–e220. doi:10.1161/01.CIR.101.23.e215

[12] Graham Greenleaf. 2023. Global Data Privacy Laws 2023: 162 National Laws and 20 Bills. *Privacy Laws and Business International Report* 181 (2023), 1, 2–4. doi:10.2139/ssrn.4426146 UNSW Law Research Paper No. 23-48.

[13] Madelon Hulsebos, Kevin Hu, Michiel Bakker, Emanuel Zgraggen, Arvind Satyanarayan, Tim Kraska, Çagatay Demiralp, and César Hidalgo. 2019. Sherlock: A Deep Learning Approach to Semantic Data Type Detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. ACM, New York, NY, USA, 1500–1508. doi:10.1145/3292500.3330993

[14] Alistair Johnson, Tom Pollard, and Roger Mark. 2019. MIMIC-III Clinical Database Demo (version 1.4). PhysioNet. doi:10.13026/C2HM2Q

[15] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3, 1 (2016), 1–9. doi:10.13026/C2HM2Q

[16] Kaggle Inc. [n. d.]. How to use Kaggle. https://www.kaggle.com/docs/api Accessed: January 26, 2025.

[17] Poornima Kulkarni and N. K. Cauvery. 2021. Personally Identifiable Information (PII) Detection in the Unstructured Large Text Corpus using Natural Language Processing and Unsupervised Learning Technique. *International Journal of Advanced Computer Science and Applications* Vol. 12, 9 (2021). https://www.proquest.com/scholarly-journals/personally-identifiable-information-pii-detection/docview/2655113578/se-2

[18] Vjeko Kuzina. [n. d.]. CASSED github implementation. https://github.com/VKuzina/CASSED Accessed: January 26, 2025.
[19] Vjeko Kužina, Ana-Marija Petric, Marko Barišić, and Alan Jović. 2023. CASSED: Context-based Approach for Structured Sensitive Data Detection. *Expert Systems with Applications* 223 (2023), 119924. doi:10.1016/j.eswa.2023.119924
[20] Han Liu, Alexander Gegov, and Frederic Stahl. 2014. Categorization and Construction of Rule Based Systems. In *Engineering Applications of Neural Networks* (Cham, 2014), Valeri Mladenov, Chrisina Jayne, and Lazaros Iliadis (Eds.). Springer International Publishing, 183–194. doi:10.1007/978-3-319-11071-4_18
[21] GDPR Local. [n. d.]. GDPR Fines: Understanding Percentages and Penalties. https://gdprlocal.com/gdpr-fines-understanding-percentages-and-penalties/ Accessed: March 23, 2025.
[22] Microsoft. 2023. Presidio: Open-source tool for personal data detection. https://github.com/microsoft/presidio Accessed: January 26, 2025.
[23] Microsoft contributors. [n. d.]. PII entities supported by Presidio. https://microsoft.github.io/presidio/supported_entities/ Accessed: January 26, 2025.
[24] Chris Moschovitis. 2021. *Privacy, regulations, and cybersecurity: The essential business guide.* John Wiley & Sons. 416 pages. doi:10.1002/9781119660156
[25] Ji-sung Park, Gun-woo Kim, and Dong-ho Lee. 2020. Sensitive Data Identification in Structured Data through GenNER Model based on Text Generation and NER. In *Proceedings of the 2020 International Conference on Computing, Networks and Internet of Things* (Sanya, China, 2020) *(CNIOT '20)*. ACM, New York, NY, USA, 36–40. doi:10.1145/3398329.3398335
[26] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. doi:10.48550/arXiv.1910.01108
[27] Md Hasan Shahriar, Anne V. D. M. Kayem, David Reich, and Christoph Meinel. 2024. Identifying Personal Identifiable Information (PII) in Unstructured Text: A Comparative Study on Transformers. In *Database and Expert Systems Applications*, Christine Strauss, Toshiyuki Amagasa, Giuseppe Manco, Gabriele Kotsis, A. Min Tjoa, and Ismail Khalil (Eds.). Springer Nature Switzerland, 174–181. doi:10.1007/978-3-031-68312-1_14
[28] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. 2014. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter* 15, 2 (June 2014), 49–60. doi:10.1145/2641190.2641198
[29] Jianliang Yang, Xiya Zhang, Kai Liang, and Yuenan Liu. 2023. Exploring the Application of Large Language Models in Detecting and Protecting Personally Identifiable Information in Archival Data: A Comprehensive Study*. In *2023 IEEE International Conference on Big Data (BigData)* (2023). 2116–2123. doi:10.1109/BigData59044.2023.10386949

A Prompt to GPT for Datasets with Context

Prompt to GPT for Datasets with Context

Initial Prompt: As a classifier of person-related data in tabular datasets, your task is to analyze the provided columns (each containing up to ten distinct values) and determine whether they contain information that originates from or relates to a person, even if it is not directly identifiable. Detecting person-related information helps ensure compliance with data protection regulations and safeguards individuals' privacy and security. Output your results in a dictionary format with a boolean indicating if the column contains person-related data or not.
Example Prompt:
You can use the following example as a guideline: Classify the following column with careful consideration of the dataset description:
Dataset:
Title: 'Test Dataset'
Description: 'This dataset was used for a linear regression.'
Features: `['first_name_en_10', 'last_name_en_10', 'email_en_10', 'phone_number', 'address_en_10', 'city_en_10', 'country_en_10', 'date', 'target']`
Column of the dataset to classify: `'first_name_en_10': ['Tom', 'Walter', 'Mia', 'Lena', 'John', 'Jack', 'Felice', 'Anna', 'Lukas', 'Will']`
Does this column, in the context of the dataset, contain information relating to a natural person?
Example Answer: `{'first_name_en_10': true}`
Data Prompt: Classify the following column with careful consideration of the dataset description.
Dataset: Title: *Absenteeism at Work*
Description: Context - The database was created with records of absenteeism at work from July 2007 to July 2010 at a courier company in Brazil.
Features: `Index(['ID', 'Reason for absence', 'Month of absence', 'Day of the week', 'Seasons', 'Transportation expense', 'Distance from Residence to Work', 'Service time', 'Age', 'Work load Average/day', 'Hit target', 'Disciplinary failure', 'Education', 'Son', 'Social drinker', 'Social smoker', 'Pet', 'Weight', 'Height', 'Body mass index', 'Absenteeism time in hours'], dtype='object')`
Column of the dataset to classify: `'ID': [3, 20, 28, 11, 15, 34, 10, 33, 14, 36]`
Does this column, in the context of the dataset, contain information relating to a natural person?
Note: The description of the dataset has been shortened for better readability.

B CRSRF Framework for Constructing Prompts

The CRSRF framework provides a structured approach for designing prompts that effectively guide large language models (LLMs) in detecting and safeguarding personal information within archives. It consists of the following key components:

- **Capacity and Role**: This element establishes the LLM's task by defining its function as a detector and protector of personal information within text-based archives. The prompt may begin with: *"As a comprehensive identifier of personal information within text-based archives…"*
- **Statement**: This defines the specific objective, explicitly stating the types of personal information the LLM should identify. A sample statement could be: *"Search for and flag any occurrences of personal names, unique identification codes such as identity card numbers or passport numbers, telephone numbers, home addresses, and mentions of family members…"*
- **Reason**: This section provides justification for the task, emphasizing the significance of protecting personal data. A well-structured reason may be: *"These details, if exposed, can compromise an individual's privacy and security. It is crucial to identify them to ensure the confidentiality and integrity of the archived documents."*
- **Format**: This specifies the preferred output format for the extracted information, ensuring structured and clear presentation. Given the nature of the data, a list format is recommended: *"Present the identified personal information in a list format, with categories such as 'Name,' 'Identification Code,' 'Telephone Number,' 'Address,' and 'Family Members' as keys."*

C Mapping of Classes

C.1 Mapping for CASSED

For the DeSSI labels, columns were labeled as personal if at least one entity belonged to the personal-related category (Table 7)

Table 7: Classification of entities in the DeSSI dataset

Personal	Non-Personal
Phone number	Other data
Address	Organization
Person	GPE
Email	SWIFT/BIC
NIN	Geolocation
Date	
Passport	
CCN	
ID Card	
Sexuality	
Gender	
Nationality	
Race	
Religion	
IBAN	

C.2 Mapping for Presidio

The classification follows the logic that direct identifiers and sensitive attributes related to individuals fall under personal, while business-related and general references are non-personal unless they reveal individual identity. This table provides an overview of the PII entities that Presidio can detect using its predefined recognizers.

Table 8: Classification of MS Presidio Recognizers

Personal	Non-Personal
CREDIT_CARD	DATE_TIME
CRYPTO	IP_ADDRESS
EMAIL_ADDRESS	LOCATION
IBAN_CODE	URL
NRP (Passport)	AU_ABN
PERSON	AU_ACN
PHONE_NUMBER	
SSN	
US_BANK_NUMBER	
US_DRIVER_LICENSE	
US_ITIN	
US_PASSPORT	
US_SSN	

D Kaggle and OpenML Datasets

To evaluate the performance of our methods on diverse and real-world data, we utilized datasets from two prominent platforms: Kaggle and OpenML.

Table 9: List of Kaggle and OpenML Datasets used.

Kaggle Datasets
Absenteeism at Work, Adult Census Income, Agriculture, Bank Marketing Campaigns, Diabetes, Graduate Admission 2, Indian Companies Registration Data, Indian Liver Patient Records, London House Price, Phishing Email, Pixar Movies, Student Performance, Titanic
OpenML Datasets
Amazon Prime Fiction, APL_20_24, CSM, DATASETBANK, company quality and valuation finance, FitBit HeartRate, HousingPrices, mango detection australia, Oilst Customers Dataset, TVS Loan Default, Avocado Prices (Augmented), echoMonths, fishcatch, forest fires, FOREX chfjpy minute Close, iris, Marvel Movies Dataset, nyc taxi green dec 2016, vowel, wine quality