Privacy-preserving Sharing of Industrial Maintenance Reports in Industry 4.0

Hicham Hossayni
Institut Polytechnique de Paris,
Schneider Electric,
38000 Grenoble, France
hicham.hossayni@se.com

Imran Khan
Schneider Electric,
38000 Grenoble, France
imran2.khan@se.com

Noel Crespi
Institut Polytechnique de Paris, IMT
91011 Evry Cedex, France
noel.crespi@it-sudparis.eu

***Abstract*—Knowledge sharing has proven its worth in many domains as an enabler for optimized processes and improved organizational agility. It makes problem-solving and decision-making much faster for the stakeholders. Knowledge sharing has also proven its effectiveness in the industrial maintenance domain as a key vector for the improvement of maintenance processes, that are still dominated by traditional practices. However, sensitive data disclosure remains one of the major roadblocks that prevent this trend from gaining ground in the manufacturing industry. In this paper, we present a new approach to provide remedies for sensitive data disclosure problems during the knowledge sharing activity. Relying on the semantic web, rule-based, and natural language processing techniques, our approach helps to detect and identify the potentially sensitive data in maintenance reports before being shared with other actors.**

***Index Terms*—Industrial maintenance, Knowledge sharing, Natural Language Processing, Semantic web, Sensitive data disclosure.**

I. Introduction

Knowledge sharing has become commonplace in various domains. It is adopted by several organizations to increase operational efficiency and staff productivity [1]. It makes the expertise of specialists accessible to a large community which helps for individual growth and development. Despite the existence of many standards and international consortiums that promote knowledge sharing for industrial maintenance, this trend has not gained traction in the manufacturing industry yet, where traditional practices are still in force, such as paper-based maintenance reports or the maintenance reports that are never reused. Also, current industry policies and practices make the maintenance knowledge inaccessible with little nay no motivation to share it with others.

In our previous work, we proposed SemKoRe [2], a vendor-agnostic solution that enables maintenance knowledge sharing with Semantic Knowledge Graphs. SemKoRe gathers all failure's related data in the knowledge graph and shares it among all connected customers in order to easily solve future failures of the same type. It received the approval of several important clients of Schneider-Electric located in several countries and from various segments.

During our work on the SemKore Project, we conducted an interview campaign with several experts in the field of industrial maintenance. The results were unequivocal; sharing maintenance knowledge between manufacturers would have a positive impact on the optimization of maintenance routines and on manufacturing productivity. However, a major problem that would prevent such solutions from becoming a reality, is the presence of sensitive information in maintenance reports. It is therefore essential to guarantee data privacy-preserving and avoiding sensitive data disclosure before sharing any maintenance report.

In this paper, we propose a new approach to avoid sensitive data disclosure during maintenance knowledge sharing through the SemKoRe solution. Our approach relies on Semantic Web ontologies combined with different techniques, usually used for data anonymization [3], such as: Named Entity Recognition or rule-based sensitive data detection.

We tackled a couple of challenges in this work. The first challenge is the lack of datasets with a reasonable number of real maintenance reports containing confidential data. Fortunately, the interviews campaign allowed us to understand the nature of sensitive data in maintenance reports and some ways to recognize it. The second challenge is that sensitive data detection techniques, especially the NLP-based ones, require annotated data corpus with a considerable number of samples, while we have almost no maintenance report with sensitive data. So, for the proof of concept needs, we implemented a generic solution to collaboratively collect, annotate, and construct our own data corpus.

The collected data corpus was used to train and evaluate three different Named Entity Recognition (NER) models. Then, we deployed and used on-premise all the trained models on an Edge gateway. The results are promising, they show that our approach can be used for on-premise detection of potentially sensitive data in maintenance reports. We also identified several areas of improvements, as future work, to make our solution usable in real use cases.

This paper is organized as follows: the related work is presented in section II, followed by the requirements imposed by our customers and the SemKoRe project. Section IV presents the main contributions of the paper. Section V details the

challenges and the future works before concluding in Section VI.

II. RELATED WORK

There are many international standards and consortium agreements that promote and adopt the maintenance knowledge sharing approach. Some of the well-known ones are:

- The OREDA project [4]: Offshore and Onshore Reliability Data for oil and gas industries, which led to the ISO 14224 international standard [5].
- ISO 6527 [6]: A Reliability Data Sharing standard for nuclear energy producers.
- SPARTA [7]: System Performance, Availability and Reliability Trend Analysis for wind energy industries in UK.
- WInDPool [8]: stands for Windenergy-Information DataPool, for wind energy industries in Germany.
- OPDE [9]: The International Pipe Failure Data Exchange.
- The Configuration Data Exchange [10]: launched by GE Aviation for the worldwide aviation industry.

All these standards and projects recommend applying data anonymization and removing sensitive data before sharing maintenance reports with others. They, however, don't propose any solution for that purpose. And to the best of our knowledge, no existing work on detecting sensitive information in industrial maintenance reports has been published. Several reasons justify this situation, the first one is that most industrial users have no intention to share their maintenance data with others as they feel that they have experienced persons who can take care of the maintenance activities without external help. However, the situation is rapidly changing due to the decline in the working force in industrial countries like Germany and Japan [11]. The second reason is that majority of the industries are closed source in nature. They never had the concept of collaboration or inclination to learn from the experiences of other teams. Another reason is that no maintenance datasets can be found on the internet or open data repositories to help researchers and scientists explore this field.

On the other side, sensitive data detection in textual or unstructured data has been studied in several works in the literature. Most of them target the medical field for which data anonymization is imposed by regulatory rules such as HIPAA (Health Insurance Portability and Accountability Act) [12]. Various techniques are used and range from simple rule-based techniques to more advanced natural language processing (NLP) techniques [13].

Named Entity Recognition (NER) remains one of the most used NLP techniques for sensitive data detection. It consists of identifying, within a text, the entities (words or group of words) that are relative to real-world objects with associated names. There are also many hybrid solutions relying on both, rule-based and NER techniques [14] that usually offer good performances with high accuracy scores, exceeding sometimes 98% [15] which is judged to be sufficient for data privacy purpose [16].

III. REQUIREMENTS

As discussed previously, this project is considered as a feature that will be proposed to our SemKoRe's customers. Thus, a set of requirements needs to be fulfilled in order to satisfy our customer's needs and to make the solution easily integrable within SemKoRe's ecosystem.

The *first* requirement is that the solution must not be restricted to specific types of machines or specific manufacturing domains. Rather, it should support almost any type of machine used in any manufacturing field. The *second* requirement is that the solution should be customizable to cover the different needs of our customers. In fact, our interviews campaigns showed that each manufacturer has its own data sensitivity evaluation criteria, and the needs of a customer are sometimes, different from others even for the same standard machines.

The *third* requirement is that the solution needs to be deployable and executable on an edge gateway. In fact, most of SemKoRe's services are running on an edge gateway directly connected to the machine. The goal is to offer on-premise services to our customers so that they can be used without needing to be permanently connected to the cloud. (please refer to our previous paper [2] for more details).

IV. CONTRIBUTION

A. Survey on maintenance reports and sensitive attributes

To understand the nature of sensitive data within a maintenance report, we reached out to several domain experts from various industries who are involved in/related to the maintenance activities in their organizations. Several questions were asked during our interviews campaign. The goals were mainly to understand the structure of maintenance reports and to know which data or attributes can be judged as sensitive. Hereafter, we present the list of questions and summarized answers we received during the interviews. For more details on these interviews like statistical validity and global conclusions please refer to our previous work [17].

1) Q1- What is the structure of a maintenance report?: Almost all maintenance reports are composed of two parts: the *first part* is a structured section (usually heading) that recalls the context of the maintenance operation such as the maintenance date and time, location, maintenance operator's name or identifier, machine Identifier. The *second part* is the content of the report, it is an unstructured text written by the maintenance operator to describe the reasons for the maintenance operation and the actions applied during the maintenance or repair of the machine.

2) Q2- Which data can be considered sensitive in a maintenance report?: We can distinguish mainly three types of sensitive data in a maintenance report:

- Personal data: all information relative to a specific person within the company. E.g.: Personally Identifiable Information (PII) like employee's name or number, role, addresses, phone number.
- Business data: every information about the company or its activity. This includes the manufacturer's name, products, location, customers, or subcontracting companies.
- Manufacturing data: especially information about the manufacturing process that may leak details about the manufactured products or the trade secrets, e.g. secret recipe of Coca-Cola. Also, some industries believe that the machine configuration is sensitive since it is part of the competitive know-how making it part of the company's industrial property.

3) Q3- How sensitive data appears in a maintenance report's content?: Personal information can only be found in the report's heading. it is very rare or almost impossible to find personal data in a maintenance report's content.

Regarding the business data, the critical data (like financial data or strategy) are never shared with the maintenance technicians, therefore it has no chance to appear in a maintenance report. However, the names of the manufacturer, products, customers, or other companies might likely appear in the report to describe the maintained machine's context.

Finally, manufacturing data can consist of multiple details but the two most important or common ones are:

- The machine configuration: corresponds to the settings of the different machine components, e.g. motor rotation speed, pressure, temperature.
- The manufacturing process: corresponds to a succession of granular steps and actions needed to produce the finished goods.

4) Q4- How to recognize sensitive data in maintenance reports?: The heading of a maintenance report is usually considered sensitive, since it contains information about the company, the location, the maintenance operator, in addition to different identifiers and references with internal significance only. Discarding the maintenance report's heading from the shared information is a wise decision.

Also, the business data differs from one company to another. It is mostly a set of proper nouns and well-known entities such as the names of products, customers, companies, etc...

On the other hand, there is a common pattern of the machine configuration and manufacturing process in the majority of maintenance reports. Actually, for every machine failure, the maintenance technicians usually describe how a machine component has been used or has behaved, or what configuration it had when the failure occurred. So, after the analysis of some maintenance reports, we found that such information is most of the time relative to a machine component since they are the most representative landmarks in a machine to describe a failure or a maintenance operation. This hypothesis was approved by the domain experts to be the most representative of the manufacturing data commonly found in the maintenance reports. Nevertheless, they do not exclude other rare forms not covered in this paper and that will be part of future works.

Finally, it is important to note that for standard machines, most of the machine configurations or manufacturing processes are provided by the OEM or the machine builder, and therefore they are not sensitive nor confidential. The exceptions are more relative to the customization applied on the standard machines such as adding new components or using a secret configuration that requires high engineering expertise. In both cases, only a domain expert can evaluate the sensitivity of each manufacturing data.

B. Our approach

1) Overview: To detect sensitive data in maintenance reports, we adopted a hybrid approach relying on rule-based techniques and Named Entity Recognition. In addition to these techniques, we used Semantic Web ontologies to guarantee the customizability, portability, and useability by different customers for different machine types.

For the detection of business data such as the manufacturer name, products names, customers, and subcontracting companies; we use a dictionary lookup technique. We ask the customer to provide a list of all proper names (e.g. products names, people, companies) typically used in the company. This list can be updated whenever new entities are introduced in the company.

In this paper, we focus on the manufacturing data and within it to the identification of the machine components in the text. Since, as presented above, they are the most common form of machine configuration or manufacturing process leaks. For that purpose, we will use NER techniques.

Finally, as discussed previously, recognizing the sensitive nature of a piece of data can only be done (for the moment) by a human expert. Therefore, to provide a highly customizable solution, we propose an ontology-based approach to allow the domain experts to specify the machine components that they judge as being potentially sensitive.

Figure 1 shows a detailed overview of our approach. It requires four different inputs:

1) A machine components taxonomy bringing together the components' names, their synonyms, and abbreviations.
2) An annotated text corpus that will be used for the training of the NER module. This corpus should be composed of numerous text entries with tagged machine components. As a requirement, all tagged machine components must be part of the machine taxonomy.
3) A specific machine ontology that describes a physical machine and all its components. This ontology must be restricted to the components defined in the machine taxonomy. In this way, we guarantee the synchronization

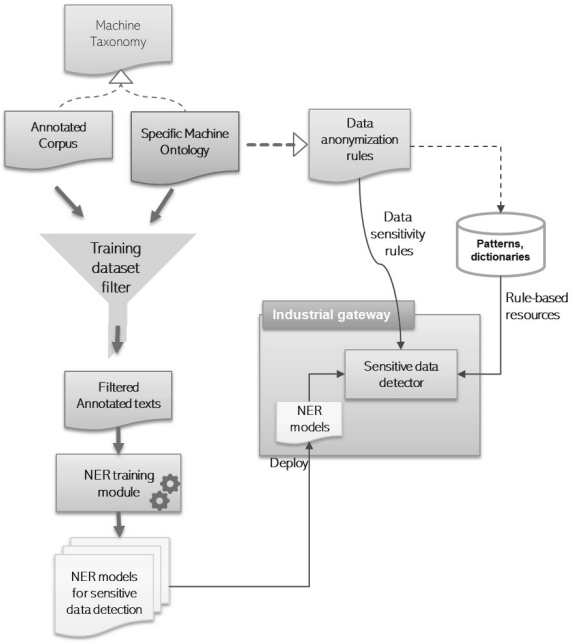

Fig. 1. Detailed overview of sensitive data detection in maintenance reports

between the components identified by the NER model and the components referenced in the machine ontology.

4) A data anonymization rules ontology, which specifies the different ways used to identify the various types of sensitive data. It is imported in the machine ontology to allow domain experts to specify the sensitivity rules or flags for each machine component, i.e. flag a component as potentially sensitive or not sensitive. This ontology also references the rule-based resources (e.g. dictionaries) that are used during the sensitive data search phase.

Note that the outcome of this approach is a custom tool for the detection of (potentially) sensitive data in the maintenance reports of a specific target machine. However, our approach is applicable to a variety of machines from different domains.

Once all the inputs are provided, the *training dataset filter* uses the *specific machine ontology* to filter the *annotated corpus*. It also ignores all tagged machine components that are not part of the target machine. We get a *filtered annotated corpus*, as a result, that only contains samples tagged with components of the target machine. The reason for this choice is that we consider the annotated corpus of general-purpose and that might reference several thousands of machine components, while only a few dozens of components are part of the target machine.

The *filtered annotated corpus* is then used to train a *NER model* for the machine components detection. It is known that reducing the training datasets helps in accelerating the training phase. Also, this allows the NER models to focus on efficiently recognizing a limited number of entities instead of trying to recognize all existing machine components. This step provides

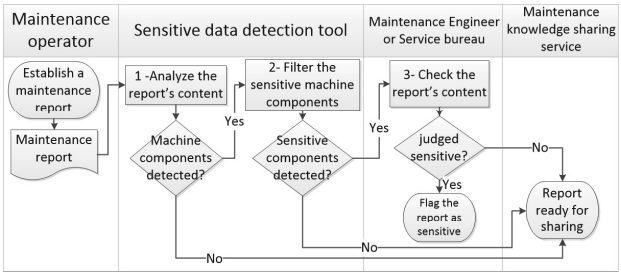

Fig. 2. Sensitive data detection flow for maintenance data sharing

a *custom NER model* trained to detect only the components of the machine for which the maintenance reports are drawn up.

To satisfy the last requirement, we deploy the trained NER model on the industrial gateway connected to the target machine. Then, the *Sensitive data detector* service will use on-premise: 1) the trained NER model, 2) the data sensitivity rules, and 3) the rule-based resources during the analysis of the maintenance operators' inputs for the recognition of (potentially) sensitive data.

2) Sensitive data detection flow in maintenance reports: The Figure 2 describes the flow of sensitive data detection in maintenance reports, it consists of three main steps:

1) The first step consists of automatically analyzing the maintenance report's text and identifying all machine components in it.
2) Once we have a list of machine components detected by the tool, it is filtered in order to keep only the items judged potentially sensitive. When sensitive components are found in the text, the maintenance report is flagged as containing potentially sensitive data and must be verified by a domain expert such as a maintenance engineer or service bureau.
3) Finally, the domain expert decides if the report can be judged sensitive or not. All reports identified as non-sensitive are tagged as ready for sharing and are transferred to the maintenance knowledge sharing service.

In the following, we will describe the design of all these components to provide a customizable sensitive data detection tool for industrial maintenance reports.

C. Data corpus collection & preparation: Industrial Machine Data Pool

The major challenge that we had to face in this project was to find the needed datasets with industrial maintenance vocabulary. Traditional sources such as internal documents or open datasets were not helpful and were not satisfying our needs. Therefore, we decided to collect and construct the required data corpus ourselves. Two complementary steps are needed to build our datasets:

1) Collect structured data and texts about machine components from internet web pages. The structured data will be used to construct our machine components' taxonomy, and the texts will be used to build the data corpus.

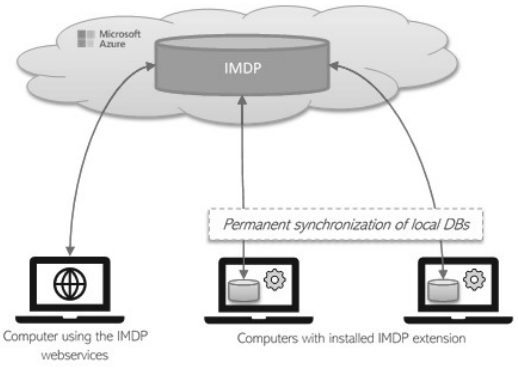

Fig. 3. IMDP's architecture

2) Annotate the collected texts using the machine components taxonomy. This step can be divided into two sub-tasks: a) *Automatic annotation* that consists of using tools to annotate automatically the collected texts and to optimize the annotation effort. b) *Manual annotation*: In this task, the user must annotate and correct the entities not detected or incorrectly annotated by the automatic annotation tools.

Thus, we implemented the Industrial Machine Data Pool that provides a collaborative solution with the necessary services for data collection and automatic and manual annotations.

1) IMDP - Industrial Machine Data Pool: Several tools are proposed on the internet for web scraping and corpus construction, e.g.: Data Scrapper from Data Miner [1], OpenLink Structured Data Sniffer [2], or Web Scraper from OpenLink Software [3]. These tools are more designed to automate the collection of structured data, however, the manual effort needed to collect unstructured text data makes them useless. Thus, we decided to develop our own tool to accelerate the data collection task. We designed an extension that can be integrated natively into internet browsers. This allows the user to interact with the data collection tool directly from the visited web pages. In fact, a simple right-click generates a popup form that the user can fill with adequate content, (even with copy and paste) and save it as a new entry in the IMDP. In addition, several users can simultaneously use the extension and participate in the data collection effort. Each instance of the browser extension manages a local database that is permanently synchronized with a central cloud database that is common to all users (see Figure 3). In order to avoid duplicate entries, all the links of the pages that are already collected are highlighted on every visited web page. IMDP server offers also a web service that can be used similarly to the internet extension, but without integration in the visited web pages.

a) Data sources & data structure: We configured IMDP to collect data about machine components, and we choose Wikipedia as a starting point because of the availability of a large number of pages describing the industrial machines and components. Also, the unified structure of Wikipedia's articles makes it possible to automate the data extraction. From each web page, we gather the following details: The component's type (structural, mechanical, or control element), name, synonyms and abbreviations, URLs, and relative texts.

Afterward, the collected texts must be cleaned and annotated. We split all collected texts into paragraphs with less than 800 characters. The goal is to simplify the text annotation and to attract more colleagues for a collaborative text annotation effort.

b) Automatic annotation: We implemented an automatic annotation tool that identifies and automatically tags machine components in the text to accelerate the annotation task. We used three different sources for our automatic annotation:

- Exact keywords and plural forms lookup from the collected machine taxonomy.
- Wikipedia annotations
- Tagme annotations: Tagme[4] is a service that performs on-the-fly semantic annotation of short text via Wikipedia as a knowledge base.

This strategy allows detecting many machine components in the texts, with, however, a non-negligible rate of false-positive annotations. Which requires the involvement of experts for manual annotation.

c) Manual annotation: During the manual annotation task, the user needs to check the correctness of automatic annotations and annotate the missed entities. When a new entity is identified by the user, it is added to the machine taxonomy either as a synonym of an existing entity or as a new entity for which the user is asked to provide the different details required by the IMPD. This allows to keep track of all entities in the texts and to know to which entity every annotation refers.

d) SUMMARY: At the time of writing these lines, we built a dataset composed of: 193 taxonomy entries with 283 synonyms and abbreviations, 188 articles texts, 3523 cleaned paragraphs, and 333 fully annotated paragraphs containing 1591 sentences.

D. Machine ontology definition

As discussed previously, our approach requires the definition of an ontology that describes the physical machine. This ontology should be defined by a domain expert (e.g. machine designer), and an ontology expert (see Fig. 4). For this purpose, we defined a T-Box for the Machine Ontology Model (see Fig. 5) that must be used to create an instance of the physical machine. The T-Box ontology describes the components of the machine and defines the characteristics of each component such as its attributes or its configuration (figure 6). Every machine is described as a hierarchical assembly of the different physical elements (Unit, Sub-unit, Part, and

[1]https://dataminer.io/
[2]https://osds.openlinksw.com/
[3]https://webscraper.io/
[4]https://tagme.d4science.org/

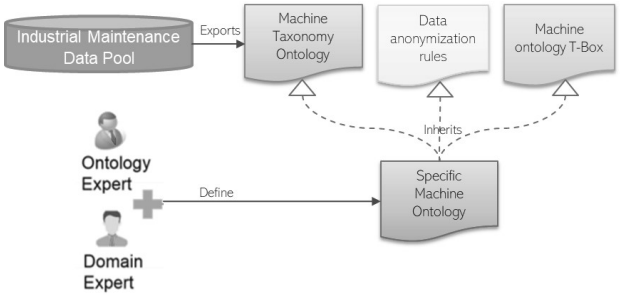

Fig. 4. Machine ontology design

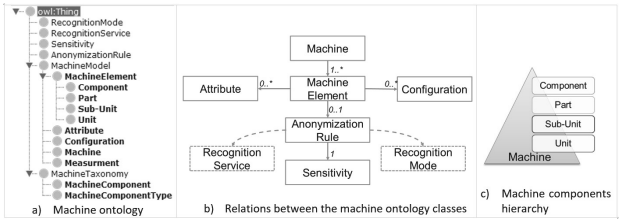

Fig. 5. T-Box & Machine ontology specifications

Component) as shown in the Pyramid in Fig.5.c. One of the Machine T-Box requirements is that all machine components must be referenced in the Machine taxonomy ontology that was generated previously by IMDP.

We also created a Data Anonymization Rules ontology that defines, at this stage, the sensitivity flags relative to each machine component. In other words, it classifies a machine component as "sensitive" or "not sensitive". The sensitivity flag can also be defined for the other machine elements. In that case, the sensitivity flag is inherited by all the machine components belonging to the flagged element unless a different sensitivity has been defined. This ontology will be extended in the future to cover more data anonymization aspects such as managing multiple sensitive data detection services (e.g. dashed concepts in Fig.5.b) or supporting the sensitive data masking or replacement rules.

1) Training Named Entity Recognition models for machine components detection: To train the Name Entity Recognition models, we predefined two categories for entities' classification: 1) Machine Component and 2) Machine Equipment. These categories represent more than 95% of the tagged entities in the collected texts. So, other categories (Maintenance process, Manufacturing process, Material, and Machine failure) are not included in our study. We trained three different NER models: 1) A custom Spacy NER model, 2) a NER model with CRF (Conditional Random Fields), and 3) a BERT-based NER model.

To train all these models, we first export the annotated texts from IMDP in the needed formats (e.g. BIO, BILUO), then we use the exported data to feed the NER training modules. Each training module applies the pre-processing steps required for the training phase. Finally, a NER model is generated and is ready for use for machine components detection.

a) Custom Spacy NER model: Spacy [18] is a very known open-source framework for NLP. It offers several features including an efficient deep CNN-based NER system achieving state-of-the-art performances. Spacy comes with a set of pre-trained NER models for different languages to detect the most common entities: Person, Location, Organization, Values. It is also possible to retrain the pre-trained models to include additional categories or even to train a new model based on an empty NER model. In our case, we chose to train an empty NER model to detect the desired categories only: Machine equipment and Machine component.

b) CRF NER model: Conditional random fields are a class of statistical methods designed for the analysis of sequential data (such as text, images, DNA) [19]. They're often used in pattern recognition and machine learning. One reason for their good performances in the NER task is that they consider the input's context by taking into account the neighboring or surrounding samples. Therefore, to train a CRF-based NER model, we need to pre-process the annotated texts in order to apply tokenization and extract different features for each token such as POS tags, lemma, shape, and other flag features like: *is uppercase, is capital, is a stop word, is a hyphen*. The features extraction is done within a window of 11 tokens, it includes the features of the current token, and 5 next and 5 previous tokens since some machine components are composed of many tokens such as: *"glass-ceramic-to-metal seal"* that is composed of 8 tokens (5 words and 3 hyphens).

c) NER model with BERT: BERT (for Bidirectional Encoder Representations for Transformers) is a deep learning model that has given state-of-the-art results on a wide variety of natural language processing tasks. It has been pre-trained on Wikipedia and BooksCorpus [20]. and requires task-specific fine-tuning [21]. In our case, we applied BERT to build our NER model for machine components detection. To train our NER model, we followed the same approach and implementation proposed in [22] as it matches our problem space. We used a 12-layer BERT model with an uncased vocabulary, and we set the target learning rate to 2*10-5 with a batch size of 16 and 7 training epochs.

2) Implementation details: We used multiple technologies to implement our proof of concept. For IMDP we used:

- Google Chrome as a target for the browser extension as it is quite a popular browser worldwide.
- HTML+CSS and Javascript to implement the web UI, and the different chrome extension services, such as data collection, cleaning, annotation, and data export.
- PouchDB, as a storage database on both, local storage and on the cloud storage. PouchDB is a no-SQL database with native data synchronization features between the local and cloud databases.

The creation of the different T-Box ontologies (Machine Taxonomy, Data Anonymization Rules, Machine Model) was mainly done with the Protégé tool, which is widely used for the creation of semantic web ontologies.

We used the Python programming language to implement the "Training dataset filer" service, and for the NER training module. Specific python libraries were used for the implementation and training of the different NER models:

- Spacy library for the custom spacy NER model.
- pyCRFsuite for the NER model with CRF.
- PyTorch library for NER model with BERT.

For the training of the different NER models, we used a capable laptop setup running, Ms. Windows 10, with 32Gb of RAM and an octa-core i7 processor and a CUDA-enabled GPU (NVIDIA QUADRO M2020) of 12Gb of memory. As an industrial gateway, we used RaspBerry Pi 4, with 4Gb of RAM, ARM-Cortex-A72, and running Linux Ubuntu 5.4. Docker was used as a containerization technology for the different Edge services such as the *Sensitive data detector* service.

Finally, we used Microsoft Azure to host the IMDP server containing the master PouchDB. Currently, the training dataset filter and NER training module services were implemented and tested on a laptop, but we plan to deploy them on a dedicated Ms. Azure server. On the Raspberry Pi, we deployed the trained NER models and the *sensitive data detector* service.

3) Evaluation: To evaluate our approach, we defined a sample machine ontology containing all the components of our machine components taxonomy, and we used the complete annotated dataset, generated by IMDP, to train the NER models. We also defined a dictionary of some names, found in the annotated texts for the rule-based part. We can deduce that the accuracy of our approach is equal to the accuracy of the NER models since the rule-based resources are defined to achieve (biasedly in our PoC) an accuracy of 100%. However, in order to fully evaluate this approach, we need to have much more data than we currently use.

To evaluate the NER models, the current dataset contains 1591 annotated sentences. 70% of the dataset (i.e. 1114 sentences) was used for the training, and 30% (477) was used for the test. We adopted also the F1-score as an evaluation metric. It is defined as follows:

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

Where the precision is the percentage of named entities found by the learning model that is correct, and the recall is the percentage of named entities present in the corpus that are found by the model.

Table 1 shows the recall, precision, and F1 score of the different models. We can see that even with a relatively small dataset we can achieve good scores exceeding 83%. CRF and BERT-based models achieve a similar F1 score of around 0.84. While the best results were achieved by Spacy's NER model getting close to 0.9 of F1 score.

TABLE I
PERFORMANCE RESULTS FOR THE DIFFERENT NER MODELS

Model	Recall	Precision	F1 score
SPACY NER	0.884	0.908	0.896
NER with CRF	0.821	0.857	0.837
NER with BERT	0.858	0.818	0.838

TABLE II
EXECUTION TIME OF THE NER MODELS ON LAPTOP AND RASPBERRY PI (MS/SAMPLE OF 1000 WORDS)

Model	Time on Laptop	Time on Raspberry Pi
SPACY NER	41	183
NER with CRF	8	14
NER with BERT	0.17	9239

We also evaluated the execution times of the different NER models, since our main execution target is an embedded industrial gateway. We found that the time needed to analyze a text and extract the sensitive entities is perfectly linear with the number of words in the text. Table 2 shows the time needed to analyze a sample of 1000 words. BERT showed exceptional performances with the ability to analyze a sample in less than 0.2ms on the laptop. However, it takes more than 9 seconds for the same task on Raspberry PI which makes it not suitable for our case. CRF's model was also fast with 8ms on the laptop and 14ms on the RPI, while Spacy's NER model takes more execution time on both, the Raspberry Pi with 183ms and the laptop with 41ms, which also remains reasonable for an on-the-fly text analysis feature. Finally, once a machine component is detected, looking in the ontology if it is sensitive or not takes a negligible time ($< 10^{-4} ms$).

V. CHALLENGES AND FUTURE WORK

In this paper, we propose a new approach for sensitive data detection in industrial maintenance reports. With the current implementation, we conducted some tests on a limited set of maintenance reports, for which we created a simple machine ontology and found promising results. However, we found several areas of improvement for the future.

The first area of improvement is that we used a restrictive hypothesis as a basis of our approach. In fact, not all sensitive data are relative to machine components. As an example, the maintenance operator could describe a manufacturing step instead of the machine component, e.g. "The packets sealing has small holes". Also, the machine components might be sometimes described by their use such as using "the milk container" instead of the "liquid tank or reservoir". This can be improved by adopting digital tools to prepare maintenance reports like a UI based on a set of ontologies for taking input from the operators by suggesting standard terms.

The second challenge is the nature of the language used for real maintenance reports, and that makes the automatic analysis awkward. In fact, maintenance operators do not provide literature texts, they usually use short informal texts, or even use street slang or urban vocabulary with frequent typos and

missing punctuation. Similarly, the operator's vocabulary understanding may not be coherent with the normative definition, which often results in the use of non-standard abbreviations. Here again, the use of digital tools can be helpful to facilitate the operator to provide details as per normative definitions. Also, using real maintenance reports to train the NLP models could help in capturing the specificities of the maintenance reports' language.

The third challenge is the use of multiple local languages to create maintenance reports. We focused so far on English maintenance reports. This can be improved by proposing a multi-lingual sensitive data tool. Such a tool will require NER models trained with various maintenance data corpus and taxonomies from each of the supported languages.

Another future work is the simplification of the data annotation process by decoupling the data annotation and the machine taxonomy. Some works like [23] showed that it is possible, for some NLP tasks, to train models on a corpus annotated with a taxonomy different from the one it is designed to output annotations for.

VI. CONCLUSION

Several standards and international consortiums promote the maintenance data-sharing approach and have proven its efficiency even on limited application scope. However, sensitive data disclosure remains one of the major roadblocks of knowledge sharing in several domains and industrial maintenance is not an exception. For this reason, we proposed a new approach to avoid sensitive data disclosure during the maintenance data sharing activity. We conducted interviews with several domain experts to understand the needs and expectations from such a feature. As a result, the needs differ from one user to another even for the same machines types, and, judging -with certitude- data to be sensitive or not, remains the role of a human expert such as a manufacturing engineer. In our approach, we aim to simplify the work of the human expert by identifying the potentially sensitive data in maintenance reports' content. This approach relies on Semantic Web technologies to allow the user to customize his solution and specify the items that can be considered as potentially sensitive. In this paper, we evaluated the use of three well-known NER models: Spacy's pre-trained NER model, CRF-based, and BERT-based NER models. The evaluation results show that even with a small dataset (with less than 1600 samples) these models have F1 scores between 0.84 and 0.90. We were able to deploy them on a Raspberry Pi 4 and we found that Spacy's NER model and CRF are more suitable for on-premise execution on an edge gateway. Several future work items have also been identified to continue working on this topic.

REFERENCES

[1] A.-U.-H. Muhammad, S. Anwar. "A systematic review of knowledge management and knowledge sharing: Trends, issues, and challenges," Cogent Business & Management, vol. 3, no. 1, p. 1127744, 2016.

[2] H. Hossayni, I. Khan, M. Aazam, A. Taleghani-Isfahani, N. Crespi, "SemKoRe: Improving Machine Maintenance in Industrial IoT with Semantic Knowledge Graphs," Applied Sciences, vol. 10, 2020.

[3] M. Nuno, B. Jorge, D. Francisco, Automated anonymization of text documents, IEEE Congress on Evolutionary Computation, 2016.

[4] SINTEF Technology and Society, Norges teknisk-naturvitenskapelige universitet, OREDA. Offshore Reliability Data Handbook, 2002.

[5] ISO 14224:2016. Petroleum, petrochemical and natural gas industries – Collection and exchange of reliability and maintenance data for equipment. International Organization for Standardization (ISO).

[6] ISO 6527, 1982: Nuclear power plants - Reliability data exchange - General guidelines, International Organization for Standardization.

[7] Portfolio Review 2016, System Performance, Availability and Reliability Trend Analysis (SPARTA), Northumberland, UK, 2016.

[8] *WInD-Pool: Wind-energy-Information-Data-Pool,* [Online].

[9] J. R. Bengt Lydell, "OPDE—The international pipe failure data exchange project," Nuclear Engineering and Design, vol. 238, 2008.

[10] GE Aviation, "GE Aviation launches Configuration Data Exchange to reduce maintenance costs,". [Online]. [Accessed 18 11 2020].

[11] Katharina Buchholz, "The Threat of Declining Working Age Populations," [online][Accessed 03 10 2021].

[12] Department of Health and Human Services, "The health insurance portability and accountability act of 1996," T. Report 65 FR, 2000.

[13] V. Veronika, R. Farkas. "De-identification in natural language processing.", 37th International Convention on Information and Communication Technology, Electronics and Microelectronics. IEEE, 2014.

[14] Y. Vithya, P. Bernhard, M. Michael, "A review of automatic end-to-end de-identification: Is high accuracy the only metric?," Applied Artificial Intelligence, vol. 34, no. 3, pp. 251-269, 2020.

[15] M. Montserrat, G.-A. Aitor, I. Ander, "Automatic De-identification of Medical Texts in Spanish: the MEDDOCAN Track, Corpus, Guidelines, Methods and Evaluation of Results," IberLEF@ SEPLN, 2019.

[16] S. Personal Data Protection Commission, "Guide To Basic Data Anonymisation Techniques," 2018. [Online]. [Accessed 29 09 2021].

[17] H. Hossayni, I. Khan, N. Crespi, "Data Anonymization for Maintenance Knowledge Sharing," IEEE IT-Professional,vol 23, no 5, 2021.

[18] M. Honnibal, M. Ines . "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing." *To appear 7.1* (2017): 411-420.

[19] Y. N. LEE, W. Sun, H. L. CHIEU, et al. Conditional random fields with high-order features for sequence labeling. Advances in Neural Information Processing Systems 22 (NIPS 2009), 2009, p. 2196-2204.

[20] Z. Yukun, K. Ryan, Z. Rich, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," IEEE international conference on computer vision, pp. 19-27, 2015.

[21] D. Jacob, C. Ming-Wei, L. Kenton: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[22] Face, Hugging, *BERT PyTorch GitHub*, 2019. [Online].

[23] S. Soufian, H. Nicolas, M. Emmanuel, "Dialogue act taxonomy interoperability using a meta-model," International Conference on Computational Linguistics and Intelligent Text Processing, 2017.