ML-based Compliance Verification of Data Processing Agreements against GDPR

Orlando Amaral[a], Sallam Abualhaija[a], and Lionel Briand[a,b]
[a]SnT, University of Luxembourg, Luxembourg
[b]School of EECS, University of Ottawa, Canada
{orlando.amaralcejas, sallam.abualhaija, lionel.briand}@uni.lu

2023 IEEE 31st International Requirements Engineering Conference (RE) | 979-8-3503-2689-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/RE57278.2023.00015

***Abstract*—Most current software systems involve processing personal data, an activity that is regulated in Europe by the general data protection regulation (GDPR) through data processing agreements (DPAs). Developing compliant software requires adhering to DPA-related requirements in GDPR. Verifying the compliance of DPAs entirely manually is however time-consuming and error-prone. In this paper, we propose an automation strategy based on machine learning (ML) for checking GDPR compliance in DPAs. Specifically, we create, based on existing work, a comprehensive conceptual model that describes the information types pertinent to DPA compliance. We then develop an automated approach that detects breaches of compliance by predicting the presence of these information types in DPAs. On an evaluation set of 30 real DPAs, our approach detects 483 out of 582 genuine violations while introducing 93 false violations, achieving thereby a precision of 83.9% and recall of 83.0%. We empirically compare our approach against an existing approach which does not employ ML but relies on manually-defined rules. Our results indicate that the two approaches perform on par. Therefore, to select the right solution in a given context, we discuss differentiating factors like the availability of annotated data and legal experts, and adaptation to regulation changes.**

***Index Terms*—Requirements Engineering (RE), Regulatory Compliance, The General Data Protection Regulation (GDPR), Data Processing Agreement (DPA), Machine Learning (ML), Natural Language Processing (NLP).**

I. Introduction

The exponential growth of artificial intelligence (AI) has significantly impacted modern software systems. Integrating AI technologies in software systems has enabled developing new features that better capture users' needs [1]. Intelligent automation has led to remarkable improvements in diverse application domains such as healthcare [2], transport [3], manufacturing [4], and finance [5]. Much of the progress of AI can be attributed, among other factors, to the increasing availability of large datasets which are paramount to drive the application of machine learning (ML), including the training of complex neural networks [6]. Such data can in many cases be personal, sensitive, or confidential. Handling massive amounts of personal data in adherence to applicable laws has added an additional burden on engineers to properly address legal requirements as part of requirements engineering (RE) practice.

Developing legally compliant software requires specifying explicit legal requirements. This task involves interpreting and transforming the legislative text into such requirements. To extract relevant information from regulations, requirements engineers who understand the functions and properties of the system-to-be should ideally collaborate with legal analysts who understand the law. Even with legal expertise, the task is still challenging, time-consuming and error-prone. First, regulations typically use legal language which can be ambiguous and is normally targeted at governing an entire industry, not specific to a software system [7]. Second, regulations are often composed of long articles and use complex natural language (NL) structures, e.g., cross-references [8].

New regulations are being continuously enforced to address concerns about privacy and data protection. In Europe, the general data protection regulation (GDPR) is the data privacy and security law effective since 2018. GDPR can affect any organization as long as they collect or process personal data of European residents [9]. This also impacts software systems which process personal data. GDPR defines *data processing* (the focus of this paper) as any operation performed on personal data such as collecting, recording, storing, using, or disclosing by transmission or dissemination (GDPR, Article 4(2)). Sanctions for violating GDPR can be substantial. Statistics show that about 337 fines reaching up to 1 billion euros were enforced due to non-compliance to GDPR [10].

The organization that collects personal data (known as *data controller*) often delegates the processing to another organization (known as *data processor*). According to GDPR, individuals must be informed through privacy policies about their rights and the terms of personal data handling. Privacy policies are agreements between individuals and data controllers. To further ensure that personal data remains protected, the controller must also have a *data processing agreement (DPA)* with the data processor. A DPA is a legally-binding contract that lays out the obligations of a data processor with regard to ensuring a secure processing and assisting the controller in its responsibilities concerning personal data protection. Such legal agreements are essential sources for eliciting legal requirements concerning data processing and are different from legal requirements that can be elicited from privacy policies. While the former can be important for software systems used directly by individuals, the latter regulates software that processes personal data beyond individuals.

To illustrate, let institution X (the data controller) be an online shopping firm that collects personal information from customers including name, birth date, postal address, social security and bank account numbers. X shares some customer information (e.g., address) with another institution Y (a logistics firm) to manage the delivery of purchased items to the customers. According to the DPA between X and Y,

2332-6441/23/$31.00 ©2023 IEEE
DOI 10.1109/RE57278.2023.00015

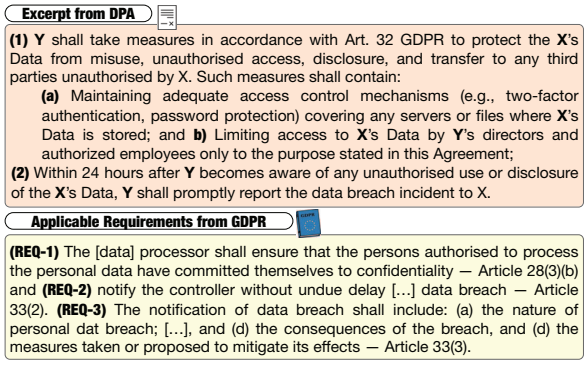

Fig. 1: Example of an excerpt DPA between Institution X (the Controller) and Y (the Processor).

only authorized employees in Y must have access to such information. Fig. 1 shows an excerpt from the DPA between X and Y alongside the applicable provisions from GDPR. If Y restricts access to authorized employees, but does not prevent downloading customer files to a shared space accessible by all its employees, such data is vulnerable to being leaked accidentally or maliciously.

To avoid violating GDPR, the requirements engineers in Y can leverage the DPA to specify explicit requirements about the technical measures needed to secure a system's data flow. Thus, ensuring that the DPA provides complete processor's obligations is paramount to developing compliant software. Fig. 1 shows at the bottom the legal actions for handling data breach in GDPR, e.g., notifying the controller without undue delay (**REQ-2**). The figure further shows on top an excerpt of a DPA where the procedure is more detailed, e.g., reporting the breach within 24 hours (sentence **(2)**). The requirements engineers can translate the different DPA statements into concrete legal requirements, e.g., encrypting data, disabling downloads, and notification alerts.

Regulatory compliance has long been studied in RE [11]–[14]. Existing work relies mostly on model-driven engineering [15], [16], restricted NL and predefined templates [17], [18]. Automated approaches for enabling compliance checking have also been investigated. Applied technologies include semantic parsing [19], rule-based [20], natural language processing (NLP) [21], ML [22], or a combination of the latter two [23], [24]. The majority of existing work focuses on the compliance and completeness of privacy policies against several regulations including GDPR [25]–[28]. While a DPA is yet another legal document imposed by GDPR, it contains legal requirements that impact software systems throughout the data processing activities beyond to what is exposed in privacy policies. For instance, software must include stronger authentication mechanisms to ensure data protection. In a recent paper, Amaral et al. [29] propose eliciting DPA-related requirements from GDPR and documenting them in NL as "shall" requirements. The authors further develop a rule-based automation (referred to as DERECHA) that verifies whether DPAs satisfy GDPR requirements based on the semantics of the DPAs' textual content. Their approach suffers from two drawbacks. First, using NL to represent GDPR requirements makes these requirements prone to various quality issues such as ambiguity and inconsistency. While NL facilitates communication between legal experts and requirements engineers during the elicitation phase, the latter might need to handle emerging quality issues when managing the overall requirements throughout the software development lifecycle. Second, adapting rules to regulation changes would entail major changes in the GDPR requirements and rule-based system, making the significant involvement of legal experts inevitable. It has been acknowledged that regulation changes and understanding the impact of this change on the compliance process is challenging [30]–[35].

To alleviate these drawbacks, we propose leveraging conceptual modeling and a combination of ML and NLP. In the regulatory compliance context, modeling helps define structured domain knowledge from the regulation [36]. In our work, we create a conceptual model that captures the semantics of DPAs, including rights and obligations. We then utilize NLP and ML to automatically classify the textual content of DPAs according to the information types in our conceptual model. Such classification is a prerequisite for detecting GDPR breaches. Incorporating ML in the automated solution has various practical benefits. Most notably, adapting the overall solution to future regulation changes does not require the intensive involvement of legal experts, in contrast to changing rules. Information types that are no longer required can simply be dropped from both the conceptual model and solution. Introducing new information types, however, requires modifying the conceptual model with the help of legal experts by adding concepts and relationships. While the task is still not trivial, it is less likely to lead to inconsistencies in comparison to changing multiple NL requirements and the corresponding rules. Then, for each new type, one must build a respective ML classifier to be plugged into the solution. Compared to the rule-based solution, however, ML requires creating annotated datasets. Developing an ML-based solution is nonetheless advantageous since examples of DPAs are available online and new annotated datasets are required only occasionally (e.g., upon a regulation change). In other words, such an ML-based solution would be practically advantageous in many contexts, provided that it fares as accurately as its alternatives.

Contributions. The paper makes the following contributions:

(1) We create, building on existing work in RE [29], [37], a holistic representation of DPA-related GDPR requirements in the form of a conceptual model that contains a total of 63 information types capturing any content to be expected in a GDPR-compliant DPA. We describe our model in Section IV.

(2) We devise an AI-enabled automation strategy for verifying the textual content of DPAs against the conceptual model. Thereafter, we refer to our approach as $D\iota\kappa AIo$, standing for **D**PA compl**i**ance chec**k**ing using **AI** techn**o**logies. Since compliance against regulations is often checked late in the software development process [38], we aim to prevent unnecessary costs by ensuring that the complete set of legal requirements concerning data processing is captured at an

early stage in RE. We describe the details of $D\iota\kappa AIo$ in Section V, and further discuss in Section VII the benefits of applying ML compared with alternatives [29]. Our replication package provides tool support and empirical data, including our non-proprietary, annotated DPAs [39].

(3) We empirically evaluate $D\iota\kappa AIo$ on 180 real DPAs including a total of $\approx$ 50,000 sentences. As we elaborate in Section VI-C, this dataset was curated as part of our work using third-party (non-author) annotators who have a strong background in law. On an evaluation set of 30 DPAs, $D\iota\kappa AIo$ detects 483 out of 582 actual genuine violations, while introducing 93 false violations, thus yielding a precision of 83.9% and a recall of 83.0%. Though the overall performance of $D\iota\kappa AIo$ is comparable to that of DERECHA [29], the practical benefits of ML, as highlighted above, makes it a more practical solution in the long-term when regulation changes and access to legal expertise is restricted.

Significance. The significance of our work is two-fold: (1) Regulatory compliance regarding data processing is a major concern for requirements engineers, particularly with the new challenges arising from the widespread application of artificial intelligence. We provide a novel, accurate approach to assist both legal experts and requirements engineers in assessing the completeness of DPAs against GDPR; (2) Existing work does not investigate alternative approaches to address this problem. We provide insights, based on a large case study, about how the two main approaches compare.

Structure. The remainder of this paper is structured as follows. Section II provides the necessary background. Section III positions our work against related work. Section IV describes our conceptual model. Section V presents $D\iota\kappa AIo$. Section VI reports on our empirical evaluation. Section VII discusses the practical considerations when devising an automation for regulatory compliance. Section VIII addresses validity considerations. Section IX concludes the paper.

II. Background

In this section, we discuss the relevant background to our compliance checking approach.

Machine Learning (ML). Text classification is a sub-field of supervised ML where a classifier is built using labeled datasets [40]. Each data point in the labeled dataset is a textual fragment which is ascribed to a label from a predefined set of labels. An example of text classification is the categorization of news articles into different genres (politics, sport, etc.). This is called multi-class classification. Binary classification is a specific case where there are two predefined labels. When a news article can be classified into multiple genres at the same time, it is referred to as multi-label classification. Such classification can be achieved by applying multiple binary classifiers, e.g., each one addressing a genre such as "politics" or "not politics", "sport" or "not sport".

Learning Features (LFs). Learning from text requires first transforming the text into numerical representations. Traditional representation methods like bag-of-words represent words by their frequency of occurrence [41]. More advanced representation methods apply embeddings to encapsulate the syntactic and semantic characteristics of the text [42]. Several methods have been proposed for deriving words and sentence embeddings, such as word2vec [43], GloVe [44], and fasttext [45]. These embeddings are context-independent, i.e., a word has always the same representation irrespective of the context in which it appears. For example, the word "bank" will have the same embedding regardless if it means "a financial institution" or "the side of a river". While such representations might be limited in terms of semantic capabilities, they are highly flexible since the embeddings can be efficiently extracted for any text. In contrast, contextualized embeddings are generated for each word depending on the context in which it appears. Such methods rely on recent large-scale language models [46]–[48] such as sentence-BERT (SBERT) [49], which is a variant of BERT [50] that is optimized for learning representations of sentences.

Imbalance Handling. Data imbalance occurs when one class has significantly fewer training examples than the other class. Imbalance can lead to building classifiers that mispredict in favor of the majority class [51]. Imbalance can be handled via random undersampling where the examples are randomly removed from the majority class. An alternative is oversampling the minority class, which can be achieved via the widely-applied (SMOTE) technique [52], that creates synthetic examples using the k-nearest-neighbor approach [53].

III. Related Work

In this section, we discuss related work on legal requirements representation and automated compliance checking.

Extracting and representing legal requirements from regulations and regulated documents (e.g., privacy policies) are extensively studied in the RE literature. In an early work, Breaux et al. [17], [54], [55] apply semantic parameterization for extracting rights and obligations from privacy regulations. Semantic parameterization enables expressing NL domain descriptions of goals as specifications in description logic. A similar method is used by Binsbergen et al. [18] to formalize norms. Hassan et al. [56] extract governance requirements from the law and enterprise regulations and transform them to formal specifications through logic models. Zeni et al. [13] develop GaiusT tool that supports extracting legal requirements from regulations. The tool is based on textual semantic annotation techniques where legal text is annotated based on concepts defined in an ontology. A similar method has been applied by Governatori et al. [57] to represent legal documents. Many approaches rely on model-driven engineering methods [11], [15], [16], [37], [58]–[60]. Usman et al. [38] provide insights into common practices and challenges when checking and analysing regulatory compliance. They provide an empirical evidence on the challenges experienced during regulatory compliance. Other approaches propose extracting descriptions of data practices from privacy policies through manual means such as crowd-sourcing or the involvement domain experts [25], [26], [61], [62]. More recently, Abualhaija et al. [8] propose using question-answering to assist engineers

with retrieving compliance-relevant information requirements from a regulation. Zasada et al. [63] evaluate the expressiveness and lexical complexity of compliance rule languages.

Automated means for checking compliance and completeness of legal requirements have been also investigated in RE. Hamdani et al. [64] present an automated GDPR compliance checking approach that relies on NLP to extract data practices from privacy policies and encodes GDPR rules to check the presence of mandatory information. NLP technologies have been utilized also for solving other problems, e.g., Bhatia et al. [19] identify incompleteness in privacy policies, Lippi et al. [20] automatically detect potentially unfair clauses in online terms of service, and Sleimi et al. [21] extract semantic metadata from legal requirements. Elluri et al. [65] automatically analyze the compliance of privacy policies against GDPR using BiLSTM multi-class classification and BERT. Torre et al. [23] and Amaral et al. [24] describe an automated solution which combines NLP and ML for the compliance verification of privacy policies according to GDPR. More recently, Rahman et al. [66] and Aborujilah et al. [67] presented ML-based techniques to monitor users' compliance with mobile applications. Tesfay et al. [22] utilize ML for summarizing privacy concerns in privacy notices to make such notices more readable and comprehensible for non-experts. Harkous et al. [27] introduce an automated framework based on neural networks for analyzing privacy policies.

We distinguish our work from the above as follows:
(1) We concentrate our work on eliciting from GDPR the legal requirements pertinent to DPA compliance. Compared to the strand of research on privacy policies, we focus on aspects of data protection regulations that must be addressed when personal data is being subsequently shared between the organizations which collect and process data. The legal requirements imposed on data controllers (through privacy policies) are different from those imposed on data processors (through DPAs), the latter being the focus of our work. Concretely, we use conceptual modeling, as commonly done in RE in other contexts, to represent the DPA-related requirements in GDPR. Our work extends the conceptual model presented in [37] with additional information types derived from the requirements provided by Amaral et al. [29].
(2) With regard to automating the compliance checking, many approaches in RE apply NLP technologies or ML. The closest to our work is DERECHA, an existing approach proposed by Amaral et al. [29]. DERECHA is composed of executable rules developed on top of requirements that are represented in NL form. Driven by our motivation to address the limitations of using NL representation and rules as discussed in Section I, our work leverages a combination of NLP and ML. We discuss the advantages of our approach over DERECHA in Section VII

IV. A Conceptual Model of Information Pertinent to DPA Compliance in GDPR

Our work draws on two existing artifacts for checking DPA compliance against GDPR. The first artifact (thereafter referred to as A) [29] is a list of 45 compliance requirements written in NL concerning data processing in GDPR. The second artifact (thereafter referred to as B) [37] is a conceptual model describing 45 information types that can be found in any DPA. In our work, we aim to build a comprehensive conceptual model that acts as an enabler for devising an automated compliance checking approach using primarily ML. To do so, we create a conceptual model by merging the two artifacts.

Fig. 2 shows the resulting conceptual model composed of 63 information types organized into four hierarchical levels: **level-1** (shaded grey), **level-2** (shaded blue), **level-3** (shaded yellow), and **level-4** (shaded white). Similar to the original artifacts (A and B), we differentiate between mandatory and optional information types. Mandatory information types originate directly from GDPR provisions, whereas optional types are based on best practices. Missing a mandatory information type leads to a non-compliant DPA, while missing an optional information type raises a warning that the DPA does not follow common practices. In our conceptual model, 33 of the information types are mandatory (font in black) and 30 are optional (font in blue). In the remainder of this paper, referring to an information type implicitly implies the hierarchical label (e.g., referring to *Data Breach* implies *Processor Obligation;Inform Controller;Data Breach*).

The figure further decomposes the conceptual model into regions to highlight the source requirements from A used to derive the information types in the model. Our method for creating this conceptual model is as follows. We first identified the information types in *A* using a qualitative method similar to the one described by Amaral et al. [24]. For example, we identify from **REQ-2** (equivalent to R34 in *A*) and **REQ-3** (equivalent to R28 in *A*) in Fig. 1 the hierarchical information types: *Processor Obligation, Inform Controller, Data Breach*, as well as the different compositions of *Data Breach*. We then rely on trace links to the actual GDPR provisions associated with *A* and *B* to map each requirement in *A* to an information type in *B*. Since we consider additional information from *A*, we adjust the specializations in **B** to extend the model. For instance, in relation with the information types listed above, we added the specialization *Inform Controller* (see Fig. 2) to encapsulate other related requirements, e.g., *Process without Instructions* derived from R10. We applied the following modes for deriving the final information types: (i) one-to-one representation, e.g., R3 in A is represented as the information type *Processing Duration* (region 2); (ii) one-to-many representation, e.g., R28 in A is represented as the four specializations of *Data Breach* (region 5); and (iii) many-to-one representation, e.g., R12 and R32 in A are represented as the information type *Secure Processing* (region 3).

To check the compliance of a DPA according to GDPR provisions, we define a set of 37 criteria, of which 22 criteria are concerned with mandatory information types and 15 are concerned with optional ones. The criteria are defined on the specializations since they can lead to violations in the more generic information types. For instance, we define 18 criteria concerned with mandatory information types under *PO*, including six coming from level-2 and 12 from level-3.

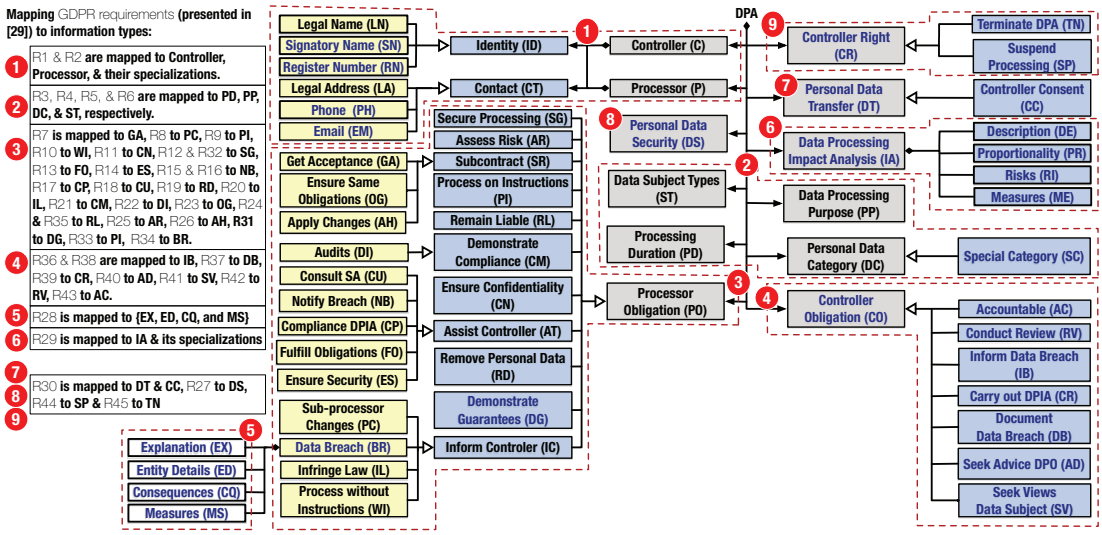

Fig. 2: Conceptual Model of DPA Compliance-related Information in GDPR.

Each criterion verifies whether the DPA satisfies or violates GDPR with respect to a particular information type. In our context, a violation refers to an absent information type, whether mandatory or optional. We denote a criterion using the negation sign ($\neg$) to refer to the absence of an information type, e.g., the criterion $\neg$**PD** checks whether the DPA contains any sentence related to *Processing Duration (PD)*. If no sentence is found, then the DPA violates GDPR. The list of compliance criteria formulated based on the absence of information types are listed in Table V in Section VI. The conceptual model and the compliance criteria are the basis for developing $DiκAIo$, as we explain next.

V. AI-based DPA Compliance Checking Approach

Fig. 3 shows an overview of $DiκAIo$, which is composed of six steps, labeled 1 – 6. $DiκAIo$ takes as input a DPA (see the excerpt in Fig. 1) and returns as output a report with recommendations about the DPA compliance. In Step 1, we preprocess the DPA using an NLP pipeline. In Step 2, we transform the text into embeddings. In Step 3, we use a large dataset of manually-annotated DPAs to train a ML classifier. In Step 4, we use ML to classify the text in the input DPA according to the information types in Fig. 2. In Step 5, we use cosine similarity to classify the text according to GDPR requirements presented in [29]. Finally, in Step 6, we combine the classification results from Steps 4 and 5 to generate recommendations about whether the input DPA is compliant. We elaborate these steps next.

Step 1: Preprocess Text. In this step, we parse the input DPA using an NLP pipeline of three modules, namely tokenization, sentence splitting and lemmatization. The first two modules obtain the sentences, whereas the last one is used to normalize the text since it identifies the canonical forms of the words, e.g., "applied" becomes "apply". The normalized sentences are then passed on to the next step.

Step 2: Extract Features. Step 2 transforms the sentences from Step 1 into embeddings. Embeddings represent the learning features which are a prerequisite for using ML-based and similarity-based classification. As we discuss in Section VI, we experiment with four alternative methods for generating the embeddings. These alternatives include the pre-trained models from word2vec [43], GloVe [44], fasttext [45] which generate 300-dimensional vectors for each word, and SBERT [49] which generates 768-dimensional vectors. SBERT produces sentence embeddings directly, accounting for the context in which the words occur. To compute the sentence embeddings corresponding to the three remaining alternatives, we take the average of the words embeddings in the sentence following common practices [24], [68].

The sentence embeddings in the input DPA are used in Step 4 for predicting the information types in each sentence and in Step 5 for computing the semantic similarity against GDPR requirements. Steps 1 and 2 are also performed on the training data to generate the embeddings needed to train the ML classifiers in Step 3.

Step 3: Train ML Classifier. We use feature embeddings from Step 2 to train an ML classifier on a large set of manually-annotated DPAs. We discuss the annotation process in Section VI-C. In this step, we train a binary classifier for each information type in Fig. 2 to achieve multi-label classification since the same text in a DPA can be about multiple information types.

To train each binary classifier, we use as positive examples all sentences that are annotated with a particular information type (e.g., *Personal Data Security*) and as negative examples all other sentences excluding the ones ascribed to that information type (e.g., all but *Personal Data Security*). The resulting training dataset for each information type was, as expected, highly skewed with significantly more negative examples than positive examples. We restrict ML only for those information types that have at least 20 positive examples, noting that

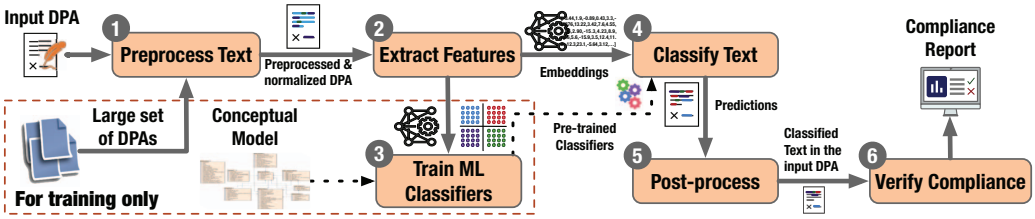

Fig. 3: Overview of $DiKAIo$.

our preliminary experiments showed poor performance for information types with less than 20 positive examples. Inspired by existing work [69], we oversample the positive examples to match the maximum number of positive examples among information types. We then undersample the negative examples to obtain a balanced dataset. The classifiers are trained on a feature matrix in which each row corresponds to a sentence and the columns are the sentence embeddings extracted in Step 2. The binary classes that a classifier predicts indicate the presence or absence of an information type in a sentence. The trained ML classifiers are fed into Step 4.

Step 4: Classify using ML. In Step 4, we first loop over the sentence embeddings from Step 2 generated for the input DPA. We then apply each binary classifier created in Step 3 to predict in each sentence whether a particular information type is present or not. For example, one ML classifier predicts that the information type *Personal Data Security* is present in sentence **(1)** in Fig. 1. The labels predicted by the ML classifiers for each sentence are then passed on to Step 6.

Step 5: Classify using Similarity. In this step, we use semantic similarity to predict the presence of an information type in a sentence. In particular, we compute the cosine similarity [41] between the embeddings of each sentence in the input DPA and the embeddings of the GDPR requirements related to DPA compliance [29]. The corresponding embeddings for each GDPR requirement is generated via the same process described earlier in Steps 1 and 2. The motivation behind Step 5 is to predict the information types with too few positive examples to effectively train ML classifiers. To increase the confidence of the prediction in Step 6, we apply similarity-based classification on all information types.

We predict an information type using semantic similarity as follows. We first loop over the sentences in the input DPA. We then loop over each GDPR requirement. For each sentence and GDPR requirement, we compute the cosine similarity of their respective embeddings. If the similarity value is above a certain threshold, the sentence is deemed similar to the requirement. We assign an information type to the sentence based on the mapping to GDPR requirements, as illustrated in Fig. 2. For example, sentence **(1)** in Fig. 1 has a similarity value of 0.52 with the requirement R27 [29] which states that "*[...] measures to ensure a level of security can include: (a) pseudonymization and encryption [...]*". The similarity-based classifier predicts the presence of *Personal Data Security (DS)* in the sentence since R27 is mapped to *DS*. An alternative for realizing this step is to group the training data under one information type and measure the similarity of the sentence in the input DPA against the average embedding of all sentences in that group. Again, if the similarity is greater than a threshold, the sentence will be assigned the same information type as the group in the training data. We plan to experiment with this alternative in future work.

According to results from preliminary experiments, we select 0.5 as the threshold value as it produced, on average, the best results across all requirements. Note that it might be beneficial to define different thresholds for different requirements since the sentences corresponding to the requirements in the DPA can contain more content than is needed to comply with the requirement. Such additional content can indeed reduce the similarity between the compliant sentence and its relevant requirement. However, we find the threshold we selected in our work to be sufficient since the purpose of $DiKAIo$ is to identify the presence of information types in at least one sentence in the DPA. The predictions made in this step are passed on to Step 6.

Step 6: Verify Compliance. Step 6 combines the labels predicted in Steps 4 and 5 to conclude a final prediction about the presence of information types in the input DPA, thus determining its compliance. Step 6 concludes that an information type ($\mathcal{I}$) is present in a given sentence (s) only if both ML and similarity-based classifiers predict $\mathcal{I}$ in s. For example, the final prediction for the sentence mentioned in Step 4 will be *Personal Data Security* since this information type is predicted by both the ML classifier in Step 4 and the similarity-based classifier in Step 5.

The output of $DiKAIo$ is generated at a DPA level as follows: If $\mathcal{I}$ is predicted in at least one sentence, then $\mathcal{I}$ is present in the DPA. Otherwise, $\mathcal{I}$ is absent. $DiKAIo$ produces as output a set of violations corresponding to mandatory or optional information types found to be absent in the input DPA.

VI. Empirical Evaluation

In this section, we empirically evaluate $DiKAIo$.

A. Research Questions (RQs)

RQ1. Which ML classification algorithm yields the most accurate results for identifying GDPR-relevant information types in DPAs? Step 4 of $DiKAIo$, which builds an ML classifier for identifying the information types present in a given DPA, can be implemented using several alternative classification algorithms and learning features. RQ1 investigates the accuracy of these alternative classifiers. The most accurate classifier is used to answer the subsequent RQs.

RQ2. How accurate is $DiKAIo$ in identifying information types in practice? Considering the best-performing ML

TABLE I: Data Collection Results.

$\mathcal{I}$ (mandatory types are in **bold**)	$\mathcal{T}$	$\mathcal{E}$	
	Sentences	Sentences	DPAs
Controller	62	20	16
Processor	97	40	27
Data Subject Types	67	56	26
Processing Duration	115	22	17
Data Processing Purpose	136	34	27
Personal Data Category	75	37	27
Processor Obligation	2680	725	29
Controller Right	17	3	3
Controller Obligation	27	5	5
Personal Data Security	1085	462	24
Personal Data Transfer	66	20	20
Data Processing Impact Analysis	4	3	1

classifier from RQ1, RQ2 assesses the accuracy of $DικAIo$ on unseen DPAs. To draw conclusions about the usefulness of devising a hybrid classification method (i.e., combining ML with semantic similarity), RQ2 further compares $DικAIo$ against a baseline that uses only ML.

RQ3. How accurate is $DικAIo$ ***in detecting GDPR violations in DPAs?*** In RQ3, we evaluate how well our approach detects GDPR violations in DPAs. We further compare $DικAIo$, which combines conceptual modeling with ML, against DERECHA, a rule-based automated approach working on GDPR requirements expressed in NL [29]. RQ3 aims to pave the way for discussing the advantages and drawbacks of these approaches, including their application from a practical standpoint, as elaborated in Section VII.

B. Implementation Details

We implemented $DικAIo$ using both Java 8.0 and Python 3.8. Different steps rely on different technologies (displayed in Fig. 3), as described next. In Step 1, we use the DKPro 1.10 toolkit [70] for operationalizing the NLP pipeline. In Step 2, we extract the sentence embeddings from SBERT [49] using the paraphrase-MPNet-base-v2 model [71]. This model is available in the Transformers 4.6.1 library [72] provided by Hugging Face (https://huggingface.co/) and operated in PyTorch [73]. We employ WEKA 3.9.6 [74], [75] in Steps 3 and 4 and implement cosine similarity [41] in Step 5.

C. Data Collection and Preparation

Our data collection focused on procuring a large set of DPAs and manually annotate them with the information types described in the conceptual model of Fig. 2. We collected a total of 180 DPAs, of which 50 were obtained from online sources and 130 were provided by our industry collaborator (name redacted for double-blind review). Data collection was performed by three third-party annotators (non-authors). All annotators are law students and went through a half-day training on GDPR compliance. The annotators produced their annotations over a four-month period, during which they declared an average of 165 hours. To mitigate fatigue, the annotation was organized in three batches, and the annotators were encouraged to restrict their work to two hours at a time.

For better understandability, we also shared with the annotators the original DPAs and the DPA-relevant compliance requirements in GDPR, presented in existing work [29]. Given their background in law, the annotators found it more efficient to read through textual requirements derived from GDPR instead of learning the definitions of the information types in our conceptual model. The four hierarchical levels of the information types in our conceptual model increase the learning duration expected by the annotators. As another measure of fatigue mitigation, we shared CSV files containing automatically generated sentences for each DPA where the annotators could select from a drop list next to each sentence up to three requirements that the sentence satisfies. We also included an additional column of free text where the annotators could add remarks, e.g., when a sentence satisfies more than three requirements.

The annotators were instructed to examine each sentence in the DPA and select all requirements that are satisfied by the sentence. As a quality measure, we compute the interrater agreement using Cohen's Kappa (κ) [76] on a subset of ≈10%, consisting of DPAs that were independently analyzed by two annotators. The interrater agreement is computed for level-1 information types only. The average κ value across pair-wise agreements of the annotations is 0.78, indicating "substantial agreement" [77]. The disagreements were discussed and resolved by the annotators.

The overall document collection resulted in analyzing ≈50,000 sentences, out of which about 4000 (≈8%) are ascribed to at least one information type. We randomly partitioned the analyzed DPAs into 150 DPAs (≈85%) used for training the ML classifiers in our approach, and 30 DPAs (≈15%) used for evaluation. Hereafter, we refer to the training dataset as $\mathcal{T}$, and the evaluation set as $\mathcal{E}$. Table I provides overall statistics about our data collection. For each level-1 information type ($\mathcal{I}$), the table lists the number of sentences ascribed with $\mathcal{I}$ in $\mathcal{T}$ and $\mathcal{E}$. Note that the sentences are not mutually exclusive since one sentence can simultaneously satisfy multiple information types. The table further shows the number of DPAs in $\mathcal{E}$ where $\mathcal{I}$ is present. For example, four sentences are available in $\mathcal{T}$ about *Data Processing Impact Analysis*, three sentences are available in $\mathcal{E}$, and this information type is present only in one DPA in $\mathcal{E}$.

D. Evaluation Procedure

To answer our RQs, we conduct the experiments below.

EXPI. This experiment addresses RQ1. In EXPI, we evaluate the different alternative ML classifiers trained over different learning features (LFs). EXPI examines six widely-applied ML classification algorithms [78], namely decision tree (DT), feed-forward neural network (FNN), Linear Discriminant Analysis (LDA), logistic regression (LR), random forest (RF), and support vector machine (SVM). We train each ML classifier over different LFs. The first three LFs are based on the pre-trained embeddings from word2vec (LF1), GloVe (LF2), fasttext (LF3), whereas the last one (LF4) is based on the sentence embeddings extracted from SBERT. More details

TABLE II: Accuracy of Alternative ML Classifiers for Identifying Information Types (**RQ1**).

	DT			FNN			LDA			LR			RF			SVM		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
LF1	67.4	73.5	69.0	71.8	74.1	72.5	69.5	76.3	71.4	69.6	74.8	71.1	73.8	74.9	74.1	76.6	82.8	78.7
LF2	67.0	73.6	68.6	70.5	72.9	71.2	67.5	73.1	69.0	68.3	72.8	69.5	72.9	73.9	73.2	75.7	82.2	77.9
LF3	66.7	69.5	67.4	70.6	72.3	71.1	70.3	75.0	71.6	67.8	71.9	68.9	71.2	72.6	72.6	74.8	80.9	76.8
LF4	**75.2**	**80.0**	**76.8**	**81.7**	**80.2**	**81.1**	**82.4**	**82.9**	**82.6**	**79.5**	**79.5**	**79.5**	**80.7**	**82.0**	**81.2**	**83.8**	**86.2**	**84.7**

LF1 – LF4: embeddings from word2vec, GloVe, fasttext, and SBERT, respectively.

on the ML classifiers and LFs can be found in Section II. Over the training set $\mathcal{T}$, we tune the hyperparameters [79] to optimize the accuracy of each alternative and further evaluate the alternative classifiers using ten-fold cross-validation.

To compare the classifiers, we define, for each information type $\mathcal{I}$, true positives (TPs) as the sentences correctly classified as $\mathcal{I}$, false positives (FPs) as the sentences that are falsely predicted as $\mathcal{I}$, and false negatives (FNs) as the sentences that should be predicted as $\mathcal{I}$ but are missed by the classifier. For each alternative classifier and LF, we aggregate the resulting TPs, FPs, and FNs for all information types and then compute the overall precision (P), recall (R) and F1 measure as: $P = |TP|/(|TP| + |FP|)$, $R = |TP|/(|TP| + |FN|)$, and $F1 = 2 * P * R/(P + R)$.

EXPII. This experiment answers RQ2. Given the best-performing alternative from EXPI, EXPII assesses how well $D\iota\kappa AIo$ can identify information types on the unseen DPAs in our evaluation set, $\mathcal{E}$. To evaluate $D\iota\kappa AIo$, we redefine TP, FP, and FN to better fit the context of compliance checking, as follows: TPs are DPAs where $D\iota\kappa AIo$ correctly predicts the presence of $\mathcal{I}$, i.e., at least one sentence is about $\mathcal{I}$. FPs are DPAs where $D\iota\kappa AIo$ falsely assumes the presence of $\mathcal{I}$, i.e., $D\iota\kappa AIo$ identifies at least one sentence for an absent $\mathcal{I}$ in the DPA. Finally, FNs are DPAs where $D\iota\kappa AIo$ falsely predicts the absence of $\mathcal{I}$, i.e., $D\iota\kappa AIo$ does not find any sentence about $\mathcal{I}$. We then report P, R, and F1, computed as in EXPI. We further compare $D\iota\kappa AIo$ against a baseline that uses ML only. The baseline is built according to Step 3 in $D\iota\kappa AIo$ (see Fig. 3) and used as in Step 4.

EXPIII. To address RQ3, we report in EXPIII the results of $D\iota\kappa AIo$ in detecting GDPR violations in the DPAs in $\mathcal{E}$. Recall from Section V that a violation corresponds to an absent information type. To evaluate $D\iota\kappa AIo$ in EXPIII, we define TPs, FPs, and FNs in converse with EXPII. Concretely, a TP is a genuine violation that is correctly detected by $D\iota\kappa AIo$. Similarly, an FP is a violation that is falsely introduced by $D\iota\kappa AIo$, and an FN is a violation that is missed by $D\iota\kappa AIo$. We then compute P and R as in EXPI. EXPIII compares the performance of $D\iota\kappa AIo$ on the same evaluation set against a reference approach from the RE literature (DERECHA) [29].

E. Answers to the RQs

RQ1. Table II reports the accuracy of 24 ML-based alternatives considered in our study for identifying information types in DPAs. Recall from Section II that LF1, LF2, and LF3 are context-independent embeddings, in contrast to LF4 which provides contextual embeddings. The table shows that LF1, LF2, and LF3 yield similar accuracy across the different ML alternatives, with an average F1 of 72.8%, 71.6%, and 71.4%, respectively. Since such embeddings learn similar information from text without any consideration of context, varying the source from which these embeddings are extracted has little impact on the accuracy of the ML classifiers in our study. Training over LF4, in contrast, significantly improves the accuracy, reaching an average F1 of 81.0%. LF4 yields an average gain in F1 of 8.2 percentage points (pp), 9.4 pp, and 9.6 pp compared to LF1, LF2, and LF3, respectively. The 768-dimensional embeddings in LF4 enable obtaining more knowledge about syntax and semantics in comparison with the 300-dimensional LF1 – LF3 embeddings.

TABLE III: Results of Information Type Identification (**RQ2**).

Solution	$\mathcal{I}$	TPs	FPs	FNs	TNs	P	R	F1
Hybrid†	Mandatory	368	93	69	130	79.8	84.2	82.0
	Optional	60	11	25	354	84.5	70.6	76.9
	Summary	428	104	94	484	**80.5**	**82.0**	**81.2**
ML	Mandatory	332	118	105	105	73.8	76.0	74.9
	Optional	52	21	33	344	71.2	61.2	65.8
	Summary	384	139	138	449	73.4	73.6	73.5

† Our approach combines ML with semantic similarity.

Focusing on LF4, Table II shows that SVM outperforms alternative classifiers across all three metrics. Pair-wise analysis using a paired t-test [80] shows statistical significance in favor of SVM ($p - value < 0.05$). Our results are not surprising considering the robust performance of SVM reported in the RE literature [68], [81], [82]. While RF is often reported to perform on par with SVM, our results rather show that LDA is comparable to SVM, with an average loss of 2.1 pp in F1. LDA has been investigated in diverse contexts and achieved promising results [83]–[85].

The answer to RQ1: SVM trained over LF4 is the best-performing alternative for identifying information types in DPAs, with an average precision of 83.8% and recall of 86.2%. We answer the subsequent RQs using this alternative.

RQ2. Table III lists the results of $D\iota\kappa AIo$ compared with a baseline that uses ML only (discussed in Section VI-D) for identifying the information types in DPAs. Note that the baseline is only applicable to information types with more than 20 positive examples (see Step 3 in Section V) and thus it has zero precision and recall for other information types. In

TABLE IV: Accuracy per Information Type (**RQ2**)

$\mathcal{I}$	P	R	$\mathcal{I}$	P	R	$\mathcal{I}$	P	R
PD	61.1	64.7	**CN**	88.9	92.3	**RL**	81.8	81.8
PP	91.7	81.5	**SG**	88.9	66.7	**AR**	66.7	88.9
DC	92.0	85.2	**FO**	91.7	95.7	**AH**	68.2	93.8
ST	92.0	88.5	**CP**	69.0	73.1	DS	84.2	66.7
GA	75.0	88.2	**RD**	90.5	94.4	IB	95.8	85.2
DC	73.9	89.5	**IL**	63.0	100	CC	75.0	95.5
PI	96.3	89.7	**CM**	86.7	100			
WI	63.3	100	**OG**	76.0	86.4			

[1] Mandatory information types are in bold; See Fig. 2 for the full names.
[2] Information types with zero precision and recall are omitted.

contrast, our approach is applicable to all information types. Overall, the table shows that $D\iota\kappa AIo$ extracts mandatory and optional information types with an average F1 of 82.0% and 76.9%, respectively. $D\iota\kappa AIo$ significantly outperforms the baseline with a gain of $\approx$ 7 pp in F1 for identifying mandatory types and $\approx$ 11 pp for identifying optional ones. The results indicate the necessity of devising a hybrid classification method to overcome the complexity of the hierarchical classification problem which can be clearly seen in our conceptual model in Fig. 2. This conclusion is in line with the RE literature [24].

In Table IV, we provide a breakdown of the results for each information type ($\mathcal{I}$). We note that the automated analysis in this paper excludes the identity and contact details of *Controller* and *Processor*. The reason is that such details are often mentioned in the same sentence at the beginning of a DPA. Automated means such as named entity recognition or regular expressions are thus not adequate for differentiating *Controller* from *Processor*. Such details, which are often known to the human analyst, can be provided as input to $D\iota\kappa AIo$ in order to enable their automatic detection.

We obtained zero precision and recall for 15 information types, three of which are mandatory, the remaining ones being optional. For better readability, we omit these information types from Table IV. The reason for the low performance on these information types is the very few training examples in our dataset to develop accurate ML classifiers. As shown earlier in Table I, optional information types have substantially fewer examples compared to mandatory ones. The only exception is *Personal Data Security* which captures the technical measures to ensure the level of security that a DPA must include, often expressed in a list spanning multiple sentences. Consequently, our approach tends to systematically predict these information types as absent. We elaborate in RQ3 the impact of these results on the compliance verification.

The answer to RQ2: $D\iota\kappa AIo$ identifies information types in DPAs with an average precision, recall, and F1 of 80.5%, 82.0%, and 81.2%, respectively.

RQ3. Table V shows the results of $D\iota\kappa AIo$ in detecting violations according to the list of compliance criteria. Recall from Section IV that a violation is related to missing mandatory or optional information types. Overall, $D\iota\kappa AIo$ identifies GDPR violations in DPAs with an average precision of $\approx$84% and recall of 83%. Following the above discussion in RQ2, the information types wrongly predicted as absent by $D\iota\kappa AIo$ resulted in a perfect recall for 15 criteria. These criteria correspond to three missing mandatory information types and 12 missing optional ones as can be seen from Table V. However, the average precision achieved by $D\iota\kappa AIo$ for the exact same criteria is $\approx$94.0%. The precision for detecting absent mandatory information types only is $\approx$84.5%. Such results show that these information types are often left out from the DPAs and are thus actually absent. Pinpointing the absence of mandatory types is essential in our context. Warning the user about missing optional types can also be informative as they capture best practices. For the remaining criteria, we analyze the root causes of errors (both FPs and FNs) made by $D\iota\kappa AIo$ and provide our conclusions below.

** Complex text:* Some information types are expressed in complex sentences which can be formulated in a different way than the corresponding GDPR requirement. Though mandatory, this information type is often a part of even a longer sentence. For example, the sentence *"The provider shall process client data only on the written instructions of the client as specified in the services agreement and this addendum includes with regard to transfer of personal data to third country or an international organization as set forth in Article 6(1)..."* describes a complex text containing multiple information types, among which is *Process Without Instructions*. Thus, considering the entire sentence as the unit of analysis negatively affects both the ML training and the measurement of semantic similarity. Examples on this error include the 11 FNs in ¬**WI**.

** Similar information types:* Some information types are very similar. Consequently, the automated classification does not properly distinguish them, leading to false violations. Examples include the FPs and FNs in ¬**CP** and ¬**CM**.

DERECHA [29], which solves the same problem, has a precision of 89.1% and recall of 82.4%. $D\iota\kappa AIo$ achieves a slightly higher recall with a gain of 0.6 pp at the cost of a drop in precision of $\approx$5 pp. Our analysis shows that DERECHA performs slightly better in verifying most of the criteria concerned with information types that have $<$100 positive examples in our training set. Conversely, $D\iota\kappa AIo$ performs better when there is a sufficient number of examples. ML is not expected to fare well in such situations, thus making the two approaches complementary. Both approaches trigger FPs and FNs which entail the need for manual work by a human analyst to correct them. While too many FPs might require the analyst to review the entire DPA, filtering out FNs can also be effort-intensive if many sentences are found to be about a particular information type.

Though neither approaches provide a perfect precision or recall, such an automation is meant to assist the human analyst in compliance checking. $D\iota\kappa AIo$ can be used, for example, to categorize the content of the DPA into a set of pre-defined information types, helping thereby the analyst to quickly browse through the DPA text and decide about its compliance. The time needed by $D\iota\kappa AIo$ to analyze the longest DPA in our collection (with 600 sentences) is $\approx$12 minutes, which is practical since the approach is typically applied offline. In Section VII, we reflect on the benefits of using ML in contrast with defining rules in our context.

TABLE V: Accuracy of our Compliance Checking Approach (**RQ3**).

ID	TPs	FPs	FNs	P	R	ID	TPs	FPs	FNs	P	R	ID	TPs	FPs	FNs	P	R	ID	TPs	FPs	FNs	P	R
¬PD	6	6	7	50.0	46.2	**¬SG**	4	8	2	33.3	66.7	**¬OG**	2	3	6	40.0	25.0	¬DP	28	2	0	93.3	100
¬PP	1	5	2	16.7	33.3	**¬FO**	5	1	2	83.3	71.4	**¬RL**	4	4	4	50.0	50.0	¬AC	26	4	0	86.7	100
¬DC	1	4	2	20.0	33.3	**¬ES**	26	4	0	86.7	100	**¬AR**	4	2	8	66.7	33.3	¬SP	29	1	0	96.7	100
¬ST	2	3	2	40.0	50.0	**¬NB**	23	7	0	76.7	100	¬TN	27	3	0	90.0	100	¬SC	30	0	0	100	100
¬GA	8	2	5	80.0	61.5	**¬CP**	0	1	9	0.0	0.0	¬DS	3	8	3	27.3	50.0	¬SV	30	0	0	100	100
¬PC	5	2	6	71.4	45.5	**¬CU**	27	3	0	90.0	100	¬BR	2	4	1	33.3	66.7	¬DG	30	0	0	100	100
¬PI	0	3	1	0.0	0.0	**¬RD**	2	7	2	22.2	50.0	¬IA	29	1	0	96.7	100	¬DB	30	0	0	100	100
¬WI	0	0	11	0.0	0.0	**¬IL**	2	1	10	66.7	16.7	¬CC	0	0	2	0.0	0.0	¬RV	30	0	0	100	100
¬CN	1	2	3	33.3	25.0	**¬CM**	0	0	4	0.0	0.0	¬IB	29	1	0	96.7	100	¬AD	30	0	0	100	100
¬AH	7	1	7	87.5	50.0	**Summary**				**TPs** = 483			**FPs** = 93			**FNs** = 99			**P** = 83.9%			**R** = 83.0%	

We use the negation sign (¬) to refer to the absence of an information type; criteria concerning mandatory information types are in bold.

The answer to RQ3: $D\iota\kappa AIo$ correctly detects 483 out of 582 genuine violations, while introducing 93 false violations. This corresponds to a precision of 83.9% and a recall of 83.0%. These results are comparable with the ones achieved by DERECHA [29].

VII. Discussion

Below, we provide insights regarding the advantages of using conceptual modeling and ML in $D\iota\kappa AIo$.

(1) Representing legal requirements as a conceptual model: Manual work is always needed to reach a precise interpretation of the regulation to represent in an analyzable form. Creating such a representation should typically involve both legal experts and requirements engineers. Legal experts might not be familiar with conceptual modeling but our observation is that they can learn with relative ease. Modeling the GDPR regulation entails defining, in a structured way, the different actors (e.g., controller and processor) as well as their obligations and rights. Creating such conceptual models is a common task in RE since these models can be used by engineers at a later stage in software development [86], [87]. Changes in regulations entail adding or removing classes and relationships in the conceptual model, thus reducing the chances of introducing inconsistencies compared to solutions requiring multiple changes in NL requirements.

(2) Automating compliance checking using ML: In this paper, we show that $D\iota\kappa AIo$ performs on par with DERECHA [29], an existing rule-based approach. We conclude, therefore, that accuracy is not a differentiating factor for selecting the best enabling technology. When automating compliance checking, other factors must be considered a priori:

- The availability of legal experts: $D\iota\kappa AIo$ requires less of their involvement than DERECHA.
- The availability of both qualified annotators and data (DPAs): Any ML approach has such prerequisites but DPAs are readily available and we provide such annotated DPAs in our replication package.
- The budget constraints for developing the automated solution: ML-based solutions require less implementation work since they are based on learning.

VIII. Threats to Validity

Internal Validity. Bias is the main concern for internal validity. To mitigate this threat, we curated the manual annotation through third-parties (non-authors). The annotators were not exposed to our implementation details at any time. Another potential threat is related to the creation of our conceptual model (presented in Fig. 2). To this end, we note that we based our work on substantial experience gained from close collaboration with legal experts. Future improvements may lead to accuracy improvements. Both our model and the basis artifacts are publicly available and thus open to scrutiny.

External Validity. Our evaluation was based on a relatively large dataset with real DPAs from different sources. The results obtained by $D\iota\kappa AIo$ over the evaluation set are reflective of a real scenario since the approach had no exposure to any of the DPAs in the evaluation set during training. This provides some confidence about the generalizability of $D\iota\kappa AIo$. Further examination through user studies would nonetheless be beneficial to improve external validity. Another threat is related to the overfitting of ML classifiers for some information types that had very few learning examples. To reduce the effect of overfitting, we improve the distribution of the minority classes using oversampling techniques and further combining ML with semantic similarity-based classification.

IX. Conclusion

In this paper, we proposed an automated approach ($D\iota\kappa AIo$) for verifying compliance of data processing agreements (DPAs) against the general data protection regulation (GDPR). $D\iota\kappa AIo$ relies on a comprehensive conceptual model that we created based on two different representations previously introduced in the RE literature [29], [37]. By leveraging a combination of natural language processing (NLP) and machine learning (ML), $D\iota\kappa AIo$ then automatically classifies the content in DPAs according to information types in the conceptual model. The resulting classification is then used to provide recommendations about the compliance of the DPAs. Over an evaluation set of 30 real DPAs, $D\iota\kappa AIo$ can detect GDPR violations in DPAs with a precision of 83.9% and a recall of 83.0%.

Acknowledgment. This paper was supported by Linklaters, Luxembourg's National Research Fund (FNR) under grant BRIDGES/19/IS/13759068/ARTAGO, and NSERC of Canada under the Discovery and CRC programs.

REFERENCES

[1] R. Feldt, F. G. de Oliveira Neto, and R. Torkar, "Ways of applying artificial intelligence in software engineering," in *6th IEEE/ACM International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)*, 2018.

[2] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015.

[3] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *International Conference on Robotics and Automation (ICRA)*, 2019.

[4] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *Journal of Manufacturing Systems*, vol. 48, 2018.

[5] T. L. D. Huynh, E. Hille, and M. A. Nasir, "Diversification in the age of the 4th industrial revolution: The role of artificial intelligence, green bonds and cryptocurrencies," *Technological Forecasting and Social Change*, vol. 159, 2020.

[6] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, 2015.

[7] T. D. Breaux and T. Norton, "Legal accountability as software quality: A us data processing perspective," in *30th IEEE International Requirements Engineering Conference*, 2022.

[8] S. Abualhaija, C. Arora, A. Sleimi, and L. C. Briand, "Automated question answering for improved understanding of compliance requirements: A multi-document study," in *30th IEEE International Requirements Engineering Conference*, 2022.

[9] J. P. Albrecht, "How the gdpr will change the world," *European Data Protection Law Review*, vol. 2, 2016.

[10] "Fines statistics: Highest fine by type of violation," https://www.enforcementtracker.com/?insights, accessed: 2010-09-30.

[11] S. Ghanavati, D. Amyot, and L. Peyton, "Compliance analysis based on a goal-oriented requirement language evaluation methodology," in *17th IEEE International Requirements Engineering Conference*, 2009.

[12] S. Ghanavati, A. Rifaut, E. Dubois, and D. Amyot, "Goal-oriented compliance with multiple regulations," in *22nd IEEE International Requirements Engineering Conference*, 2014.

[13] N. Zeni, N. Kiyavitskaya, L. Mich, J. R. Cordy, and J. Mylopoulos, "Gaiust: supporting the extraction of rights and obligations for regulatory compliance," *Requirements Engineering*, vol. 20, 2015.

[14] O. Akhigbe, D. Amyot, and G. Richards, "A systematic literature mapping of goal and non-goal modelling methods for legal and regulatory compliance," *Requirements Engineering*, vol. 24, no. 4, 2019.

[15] D. Torre, G. Soltana, M. Sabetzadeh, L. C. Briand, Y. Auffinger, and P. Goes, "Using models to enable compliance checking against the GDPR: an experience report," in *22nd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems,*, 2019.

[16] P. Pullonen, J. Tom, R. Matulevicius, and A. Toots, "Privacy-enhanced BPMN: Enabling data privacy analysis in business processes models," *Software & Systems Modeling*, vol. 18, no. 6, 2019.

[17] T. D. Breaux, M. W. Vail, and A. I. Anton, "Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations," in *14th IEEE International Requirements Engineering Conference*, 2006.

[18] L. T. van Binsbergen, L.-C. Liu, R. van Doesburg, and T. van Engers, "Eflint: A domain-specific language for executable norm specifications," in *Proceedings of the 19th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences*, 2020.

[19] J. Bhatia, M. C. Evans, and T. D. Breaux, "Identifying incompleteness in privacy policy goals using semantic frames," *Requirements Engineering*, vol. 24, no. 3, 2019.

[20] M. Lippi, P. Pałka, G. Contissa, F. Lagioia, H.-W. Micklitz, G. Sartor, and P. Torroni, "Claudette: an automated detector of potentially unfair clauses in online terms of service," *Artificial Intelligence and Law*, vol. 27, no. 2, 2019.

[21] A. Sleimi, N. Sannier, M. Sabetzadeh, L. C. Briand, and J. Dann, "Automated extraction of semantic legal metadata using natural language processing," in *26th IEEE International Requirements Engineering Conference*, 2018.

[22] W. B. Tesfay, P. Hofmann, T. Nakamura, S. Kiyomoto, and J. Serna, "Privacyguide: Towards an implementation of the EU GDPR on internet privacy policy evaluation," in *4th ACM International Workshop on Security and Privacy Analytics, IWSPA@CODASPY*, 2018.

[23] D. Torre, S. Abualhaija, M. Sabetzadeh, L. C. Briand, K. Baetens, P. Goes, and S. Forastier, "An ai-assisted approach for checking the completeness of privacy policies against GDPR," in *28th IEEE International Requirements Engineering Conference*, 2020.

[24] O. Amaral, S. Abualhaija, D. Torre, M. Sabetzadeh, and L. Briand, "AI-enabled automation for completeness checking of privacy policies," *IEEE Transactions on Software Engineering*, vol. 48, no. 11, 2021.

[25] S. Wilson, F. Schaub, R. Ramanath, N. M. Sadeh, F. Liu, N. A. Smith, and F. Liu, "Crowdsourcing annotations for websites' privacy policies: Can it really work?" in *Proceedings of the 25th International Conference on World Wide Web*, 2016.

[26] T. Linden, R. Khandelwal, H. Harkous, and K. Fawaz, "The privacy policy landscape after the gdpr," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, no. 1, pp. 47–64, 2020.

[27] H. Harkous, K. Fawaz, R. Lebret, F. Schaub, K. G. Shin, and K. Aberer, "Polisis: Automated analysis and presentation of privacy policies using deep learning," in *27th {USENIX} security symposium ({USENIX} security 18)*, 2018.

[28] E. Poplavska, T. B. Norton, S. Wilson, and N. M. Sadeh, "From prescription to description: Mapping the GDPR to a privacy policy corpus annotation scheme," in *Legal Knowledge and Information Systems - JURIX 2020: The Thirty-third Annual Conference, Brno, Czech Republic, December 9-11, 2020*, V. Serena, J. Harasta, and P. Kremen, Eds., 2020.

[29] O. Amaral, M. I. Azeem, S. Abualhaija, and L. C. Briand, "Nlp-based automated compliance checking of data processing agreements against gdpr," *arXiv preprint arXiv:2209.09722*, 2022.

[30] E. Union, "Commercial sector: launch of the adoption procedure for a draft adequacy decision on the eu-u.s. data privacy framework," Accessed Jun. 02, 2023 [Online]. [Online]. Available: https://commission.europa.eu/law/law-topic/data-protection/international-dimension-data-protection/eu-us-data-transfers_en

[31] P. Breitbarth, "The impact of gdpr one year on," *Network Security*, vol. 2019, no. 7, pp. 11–13, 2019.

[32] S. Shastri, V. Banakar, M. Wasserman, A. Kumar, and V. Chidambaram, "Understanding and benchmarking the impact of gdpr on database systems," *arXiv preprint arXiv:1910.00728*, 2019.

[33] A. Shah, V. Banakar, S. Shastri, M. F. Wasserman, and V. Chidambaram, "Analyzing the impact of gdpr on storage systems," in *USENIX Workshop on Hot Topics in Storage and File Systems*, 2019.

[34] H. Li, L. Yu, and W. He, "The impact of gdpr on global technology development," pp. 1–6, 2019.

[35] E. Johansson, K. Sutinen, J. Lassila, V. Lang, M. Martikainen, and O. M. Lehner, "Regtech – a necessary tool to keep up with compliance and regulatory changes," *ACRN Journal of Finance and Risk Perspectives, Special Issue Digital Accounting*, vol. 8, pp. 71–85, 2019.

[36] D. Torre, M. Alferez, G. Soltana, M. Sabetzadeh, and L. Briand, "Modeling data protection and privacy: application and experience with gdpr," *Software and Systems Modeling*, vol. 20, no. 6, 2021.

[37] O. Amaral, S. Abualhaija, M. Sabetzadeh, and L. Briand, "A model-based conceptualization of requirements for compliance checking of data processing against gdpr," in *29th IEEE International Requirements Engineering Conference Workshops*, 2021.

[38] M. Usman, M. Felderer, M. Unterkalmsteiner, E. Klotins, D. Mendez, and E. Alégroth, "Compliance requirements in large-scale software development: An industrial case study," in *Product-Focused Software Process Improvement*, 2020.

[39] O. Amaral, S. Abualhaija, and L. Briand. (2023) Shared material for "DPA compliance checking using AI technologies (DIKAIO)". Available at https://figshare.com/s/eb830963da23591862fc, March 2023.

[40] C. C. Aggarwal, *Machine Learning for Text*. Springer Publishing Company, Incorporated, 2018.

[41] C. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, 2008.

[42] O. Levy and Y. Goldberg, "Dependency-based word embeddings," in *52nd Annual Meeting of the Association for Computational Linguistics*, 2014.

[43] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings*, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[44] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[45] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[46] T. Hossain, R. L. Logan IV, A. Ugarte, Y. Matsubara, S. Young, and S. Singh, "Covidlies: Detecting covid-19 misinformation on social media," in *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, 2020.

[47] S. Jolly and S. Kapoor, "Can pre-training help vqa with lexical variations?" in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 2863–2868.

[48] A. Waldis, T. Beck, and I. Gurevych, "Composing structure-aware batches for pairwise sentence classification," in *Findings of the Association for Computational Linguistics: ACL 2022*, 2022, pp. 3031–3045.

[49] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019.

[50] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[51] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, 2019.

[52] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, 2002.

[53] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2016.

[54] T. D. Breaux and A. I. Antón, "Analyzing goal semantics for rights, permissions, and obligations," in *13th IEEE International Conference on Requirements Engineering (RE'05)*, 2005.

[55] T. D. Breaux, A. I. Antón, and J. Doyle, "Semantic parameterization: A process for modeling domain descriptions," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 18, no. 2, 2008.

[56] W. Hassan and L. Logrippo, "Governance requirements extraction model for legal compliance validation," in *2009 Second International Workshop on Requirements Engineering and Law*, 2009.

[57] G. Governatori, M. Hashmi, H.-P. Lam, S. Villata, and M. Palmirani, "Semantic business process regulatory compliance checking using legalruleml," in *20th EKAW International Conference*, 2016.

[58] G. Soltana, M. Sabetzadeh, and L. C. Briand, "Model-based simulation of legal requirements: Experience from tax policy simulation," in *24th IEEE International Requirements Engineering Conference*, 2016.

[59] D. Torre, M. Alférez, G. Soltana, M. Sabetzadeh, and L. C. Briand, "Model driven engineering for data protection and privacy: Application and experience with GDPR," *CoRR*, vol. abs/2007.12046, 2020.

[60] E. Vanezi, G. M. Kapitsaki, D. Kouzapas, A. Philippou, and G. A. Papadopoulos, "Diálogop-a language and a graphical tool for formally defining gdpr purposes," in *14th International Conference RCIS: Research Challenges in Information Science*, 2020.

[61] T. D. Breaux and F. Schaub, "Scaling requirements extraction to the crowd: Experiments with privacy policies," in *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, 2014, pp. 163–172.

[62] S. Wilson, F. Schaub, A. A. Dara, F. Liu, S. Cherivirala, P. Giovanni Leon, M. Schaarup Andersen, S. Zimmeck, K. M. Sathyendra, N. C. Russell, T. B. Norton, E. Hovy, J. Reidenberg, and N. Sadeh, "The creation and analysis of a website privacy policy corpus," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. ACL, 2016, pp. 1330–1340.

[63] A. Zasada, M. Hashmi, M. Fellmann, and D. Knuplesch, "Evaluation of compliance rule languages for modelling regulatory compliance requirements," *Software*, vol. 2, no. 1, 2023.

[64] R. E. Hamdani, M. Mustapha, D. R. Amariles, A. Troussel, S. Meeùs, and K. Krasnashchok, "A combined rule-based and machine learning approach for automated gdpr compliance checking," in *18th International Conference on Artificial Intelligence and Law*, 2021.

[65] L. Elluri, S. S. L. Chukkapalli, K. P. Joshi, T. Finin, and A. Joshi, "A bert based approach to measure web services policies compliance with gdpr," *IEEE Access*, vol. 9, 2021.

[66] M. S. Rahman, P. Naghavi, B. Kojusner, S. Afroz, B. Williams, S. Rampazzi, and V. Bindschaedler, "Permpress: Machine learning-based pipeline to evaluate permissions in app privacy policies," *IEEE Access*, vol. 10, 2022.

[67] A. Aborujilah, A. Z. Al-Othmani, Z. A. Long, N. S. Hussien, and D. A. Ghani, "Conceptual model for automating gdpr compliance verification using natural language approach," in *2022 International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IoE)*, 2022.

[68] S. Ezzini, S. Abualhaija, C. Arora, and M. Sabetzadeh, "Automated handling of anaphoric ambiguity in requirements: A multi-solution study," in *44th International Conference on Software Engineering*, 2022.

[69] R. Rasiman, F. Dalpiaz, and S. España, "How effective is automated trace link recovery in model-driven development?" in *Requirements Engineering: Foundation for Software Quality*. Springer International Publishing, 2022, pp. 35–51.

[70] R. Eckart de Castilho and I. Gurevych, "A broad-coverage collection of portable NLP components for building shareable analysis pipelines," in *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, 2014.

[71] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "Mpnet: Masked and permuted pre-training for language understanding," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[72] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020.

[73] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *33rd Conference on Neural Information Processing Systems*, 2019.

[74] J. H. Hayes, W. Li, and M. Rahimi, "Weka meets tracelab: Toward convenient classification: Machine learning for requirements engineering problems: A position paper," in *1st International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, 2014.

[75] F. Eibe, M. Hall, and I. Witten, "The weka workbench. online appendix for data mining: Practical machine learning tools and techniques," *Morgan Kaufmann*, 2016.

[76] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement (EPM)*, vol. 20, no. 1, 1960.

[77] A. J. Viera, J. M. Garrett *et al.*, "Understanding interobserver agreement: the kappa statistic," *Family Medicine*, vol. 37, no. 5, 2005.

[78] S. Abualhaija, C. Arora, M. Sabetzadeh, L. Briand, and E. Vaz, "A machine learning-based approach for demarcating requirements in textual specifications," in *27th IEEE International Requirements Engineering Conference*, 2019.

[79] R. Kohavi, *Wrappers for performance enhancement and oblivious decision graphs*. Stanford University, 1996.

[80] C. Nadeau and Y. Bengio, "Inference for the generalization error," *Machine Learning*, 2001.

[81] A. Sainani, P. R. Anish, V. Joshi, and S. Ghaisas, "Extracting and classifying requirements from software engineering contracts," in *28th IEEE International Requirements Engineering Conference*, 2020.

[82] F. u. Hassan and T. Le, "Automated requirements identification from construction contract documents using natural language processing," *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, vol. 12, no. 2, 2020.

[83] C. H. Park and H. Park, "A comparison of generalized linear discriminant analysis algorithms," *Pattern Recognition*, vol. 41, no. 3, 2008.

[84] J. Ghosh and S. B. Shuvo, "Improving classification model's performance using linear discriminant analysis on linear data," in *10th International Conference on Computing, Communication and Networking Technologies*. IEEE, 2019.

[85] H. Wang, C. Ding, and H. Huang, "Multi-label linear discriminant analysis," in *Proceedings on Computer Vision, Greece, September 5-11*. Springer, 2010.

[86] G. Lucassen, M. Robeer, F. Dalpiaz, J. M. E. Van Der Werf, and S. Brinkkemper, "Extracting conceptual models from user stories with visual narrator," *Requirements Engineering*, vol. 22, 2017.

[87] B. Ramesh and V. Dhar, "Supporting systems development by capturing deliberations during requirements engineering," *IEEE Transactions on Software Engineering*, vol. 18, no. 6, 1992.