IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 January 2024, accepted 7 February 2024, date of publication 8 February 2024, date of current version 15 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3364533

RESEARCH ARTICLE

Leveraging NLP Techniques for Privacy Requirements Engineering in User Stories

GUNTUR BUDI HERWANTO[1,2], GERALD QUIRCHMAYR[1], AND A. MIN TJOA[1,3]

[1]Faculty of Computer Science, University of Vienna, 1010 Vienna, Austria
[2]Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia
[3]Institute of Information Engineering TU WIEN, 1040 Wien, Austria

Corresponding authors: Guntur Budi Herwanto (gunturbudi@ugm.ac.id) and A. Min Tjoa (a.tjoa@tuwien.ac.at)

This work was supported in part by the University of Vienna; in part by the Indonesia Endowment Fund for Education (IEFE/LPDP), Ministry of Finance, Indonesia; and in part by the Faculty of Computer Science, University of Vienna, through Universitas Gadjah Mada, Indonesia. Open Access funding is provided by the University of Vienna Bibliothek and TU Wien Bibliothek.

ABSTRACT Privacy requirements engineering acts as a role to systematically elicit privacy requirements from system requirements and legal requirements such as the GDPR. Many methodologies have been proposed, but the majority of them are focused on the waterfall approach, making adopting privacy engineering in agile software development difficult. The other major issue is that the process currently is to a high degree manual. This paper focuses on closing these gaps through the development of a machine learning-based approach for identifying privacy requirements in an agile software development environment, employing natural language processing (NLP) techniques. Our method aims to allow agile teams to focus on functional requirements while NLP tools assist them in generating privacy requirements. The main input for our method is a collection of user stories, which are typically used to identify functional requirements in agile software development. The NLP approach is then used to automate some human-intensive tasks such as identifying personal data and creating data flow diagrams from user stories. The data flow diagram forms the basis for the automatic creation of privacy requirements. Our evaluation shows that our NLP method achieves a fairly good performance in terms of F-Measure. We are also demonstrate the feasibility of our NLP approach in CamperPlus project. Lastly, we are developing a tool to integrate our NLP approach into the privacy requirements engineering pipeline, allowing for manual editing of results so that agile teams can maintain control over the automated approach.

INDEX TERMS Privacy requirements engineering, natural language processing, agile software development, user stories.

I. INTRODUCTION

The principles of ''privacy by design'' and ''privacy by default'' have a long-standing tradition [1] and are today obligatory by legislation. These principles are most prominently highlighted in the European Union's General Data Protection Regulation (GDPR), which comes into force in 2018. In this context, it is essential to raise awareness among developers and ensure that systems are built in an a-priori privacy-aware way. Since the introduction of the GDPR, the integration of privacy aspects into the software development process has become a key concern and a major challenge for the industry, especially for digital product developers. To address this challenge, privacy engineering has emerged as a research framework that focuses on integrating data protection into organizational and technical measures [2].

Privacy engineering bridges the gap between legal obligations, privacy policies, organizational policies, and the realization of systems or technologies under development [3]. Privacy engineering must be integrated into the software development lifecycle (SDLC), starting with requirements analysis, design, and implementation [4]. The specific activity in the requirements phase can be referred to as Privacy Requirements Engineering (PRE). These activities include analyzing assets, identifying data subjects, and drawing data flow diagrams to determine the necessary

The associate editor coordinating the review of this manuscript and approving it for publication was Sedat Akleylek.

data protection and privacy requirements. The popular approaches to eliciting privacy requirements is conducting a risk analysis [5], threat analysis [6], and privacy impact assessment (PIA) [7]. These analyses are conducted in the early stage of development. However, many privacy engineering methodologies depend heavily on a plan-driven approach that can be time-consuming and not tailored to the agile speed that much of the industry is currently taking [8]. This agility falls under agile software development (ASD) methodologies such as Scrum, which typically cannot capture the full complexity of the system at an early stage [9]. The turn from a plan-driven approach to ASD means adopting new ways to make development faster and more efficient [10]. Consequently, modeling privacy threats or designing a privacy-friendly system becomes more challenging [11]. The steps in PRE methods such as assets identification and modeling the system to identify privacy threats could take an immense amount of time [12], preventing the iterative nature of the agile approach in changing requirements [13]. Researchers have studied these challenges [11], [14], and clearly state the conflicting nature of agile and privacy engineering.

The agile software development methodology promotes the use of user stories as a means to elicit functional requirements (FRs). As a result, user stories have become the primary form of requirement elicitation, representation, and documentation in ASD [15]. Despite their simplicity, user stories have been shown to effectively refine customer requirements and make them more detailed and precise [15]. Typically, customer requirements are expressed as FR and do not encompass non-functional requirements (NFRs), such as security and privacy. Eliciting NFRs requires expertise to be defined and are often specified in separate sessions to ensure their completeness [16]. However, this approach can result in NFR being treated as an afterthought, potentially leading to technical debt or even system failure [16].

Defining NFR early on aligns with the principles of privacy by design. The use of user stories has been proposed as a means to facilitate privacy requirements engineering (PRE) in agile situations [4], [17]. However, relying solely on user stories for NFR can become an issue [18], highlighting the importance of considering functional requirements (FR) in conjunction with NFR, especially when it comes to PRE. Addressing NFRs in conjunction with FRs is essential successful identification and implementation of NFR in ASD [18]. In the context of privacy requirements, this means that agile teams need to be able to identify privacy criteria in FRs, with research showing that agile teams still struggle to identify privacy criteria in user stories [19]. The use of computerized tools has been suggested to assist agile teams in identifying NFRs [16], [20], [21]. In the domain of PRE, several tools have been proposed in literature [22], [23]. Despite the availability of tools, the manual work involved, such as identifying actors, personal data, and completing numerous forms, can hinder the utilization of these tools to their full potential. We very much believe that incorporating an intelligent system (IS), particularly machine learning focused on natural language processing (NLP), can help agile teams to significantly reduce this manual workload [24], [25], [26].

The importance of incorporating IS into ASD has been highlighted by various studies investigating the benefits of utilizing NLP-based models in user stories [25], [27]. These studies have shown the potential benefits of using NLP for PRE activities, such as defect detection and visual model generation [27]. However, a systematic mapping study has revealed that NLP solutions have yet to be fully incorporated into proposed PRE methods [28]. Our research aims to fill this gap by exploring the use of NLP for a comprehensive pipeline of PRE activities.

We present a set of NLP solutions aimed at streamlining various activities within the PRE process. These solutions aim to automate manual and time-consuming tasks, such as identifying assets and creating data flow diagrams from functional requirements, and ultimately reduce the huge burden of eliciting privacy requirements for agile teams. Our approach aims to facilitate the integration of IS in the complete pipeline of privacy requirements by utilizing user stories as the primary input and user stories as the final output. As the process is nearly fully automated, the minimal effort of eliciting NFR alongside FR strongly supports privacy by design principles as a triggering facilitator for agile teams to avoid leaving NFR until later stages [18], thereby supporting privacy by design principles. In this paper, we present several key contributions:

- We present a novel approach to enhance the PRE process by incorporating NLP-based intelligent systems and user stories as the primary input, thus simplifying the integration of PRE into agile workflows.
- We present a collection of privacy requirements presented as user stories, offering a set of patterns for eliciting privacy requirements in an agile development environment.

The core content of the paper is structured as follows: Section II discusses recent approaches to privacy requirements engineering and natural language processing in user stories. Section III describes our proposed approach for incorporating NLP solutions into privacy requirements engineering. Section IV discusses the reliability of the NLP approach in terms of performance compared to the gold standard. Section V puts our approach into a real-world project. Section VI depicts the tool we created to integrate our NLP solution. Section VII discusses the limitations and potential impact of our proposed approach. Section VIII shows the threats to the validity of our approach. Finally, Section IX summarizes the results of our work and discusses potential future work.

II. BACKGROUND AND RELATED WORK

This section describes the privacy requirements engineering process based on recent work [4], [6], [22], [29]. We then discuss the natural language processing techniques that can

be used to facilitate the application of privacy requirements engineering.

A. PRIVACY REQUIREMENTS ENGINEERING

There are numerous techniques for designing privacy requirements. Notario et al. [4] considered it from two perspectives: a goal-oriented and a risk-based perspective. The goal-oriented approach is centered on extracting principles and establishing them as the need for the system to meet. The goal of privacy or data protection can be derived from privacy principles and legal requirements. Hansen et al. [30] defined six data protection goal, which consists of *Confidentiality*, *Integrity*, *Availability*, *Unlinkability*, *Transparency*, and *Intervenability*. The Confidentiality, Integrity, and Availability (CIA) Triad is a commonly used framework for evaluating and defining security requirements, and is considered a cornerstone of information security [31]. In the context of data protection, the CIA Triad can be redefined to meet specific requirements, such as privacy and data protection regulations [22]. These six goals, along with the additional goals, are used by Meis and Heisel [22] to develop a PRE method under the ProPAn method.

Meanwhile, the risk-based approach was more concerned with the threat that could arise from the movement of assets and data in the system. Privacy threat modeling is one discipline that uses a risk-based approach to address privacy. Deng et al. [6] present LINDDUN, a comprehensive privacy threat modeling technique. LINDDUN is a mnemonic that represents a potential privacy concern in a system under consideration: Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of Information, Unawareness, and Non-compliance. Prior to analyzing the threat, the analyst should provide a data flow diagram of the system (DFD). Each component of the DFD can then be linked to multiple threats in LINDDUN.

Privacy Impact Assessment (PIA) is also part of the risk-based approach. As with LINDDUN [6], the risk-based approach should begin with system characterization, which often includes asset identification and system modeling. Oetzel and Spiekermann [7] presented a seven-step process for eliciting privacy threats and documenting the PIA. This process can also be viewed as goal-oriented, as its second step incorporates the use of privacy goal catalogs, which provide a set of predefined privacy goals to guide the PIA process and ensure that all relevant privacy considerations are taken into account. Thus, it is important to recognize that an effective PRE strategy should incorporate both a goal-oriented and a risk-based approach.

Notario et al. [4] combine risk-based and goal-oriented approaches under the PRIPARE methodology. The goal-oriented approach, which is easier for a novice privacy engineer to follow, is the first step in reducing uncertainty in privacy engineering. The remainder of the potential risk is then addressed using a risk-based approach. The combination of goal-oriented and risk-based approaches provides a more comprehensive approach to PRE. P-STORE [29], the latest PRE which combines both approaches begins by establishing and prioritizing privacy objectives based on organizational goals. Prioritization is necessary because of the unworkability to meet all privacy goals simultaneously [30]. For example, if we value confidentiality goals, unlinkability goals will be overridden [22]. Once the goal has been specified, the requirements engineer can select the critical asset that will form the basis of the privacy threat analysis. These threats are mapped and prioritized in LINDDUN to become the ultimate privacy requirement to be addressed.

An approach that relies on extensive planning, such as ProPAn [22], is considered unsuitable for ASD [22]. The fundamental difficulty for ASD, such as Scrum, is the lack of system-wide planning. The PRIPARE method proposes a managerial approach to integrate its approach with ASD. The first is through an incremental approach, which means that privacy planning is only considered in its current iteration. In addition, privacy planning should be included prior to the start of the sprint. PRIPARE also considered including privacy in a dedicated sprint. In contrast to PRIPARE, Peixoto [23] aims to provide tool support to agile teams by guiding them through the specification of privacy requirements. However, extensive knowledge of privacy is required to fill out the form provided by the tools.

Agile methodologies encourage frequent communication between teams. This can be achieved in a variety of ways, one of which is the use of card games, an engaging and interactive approach [12], [32]. An example of this is Threat Poker [32] and LINDDUN GO [12], tools that empower Scrum teams to identify potential threats during the course of the game and subsequently assess the level of risk and the corresponding effort required to counteract these threats [32]. This card game also can serve as an educational instrument, particularly for those team members with limited security and privacy expertise [12]. However, this strategy presents a challenge in that each team member needs to possess the requisite knowledge of potential threats associated with a given user story. In addition, practitioners caution that these tools may not be appropriate for more complex microsystems, highlighting the need for careful tool selection and adaptation based on the project's unique context and requirements.

The current literature presents several limitations in addressing PRE problems in ASD. Despite the comprehensive approach offered by ProPAN [22], it still requires a huge workload of careful planning and expertise to follow its plan-driven steps. LINDDUN needs a significant amount of time for modeling the system [6], also in-depth privacy expertise is needed to specify accurate privacy criteria [23]. The lightweight approaches offered by Threat Poker and LINDDUN GO may not provide a comprehensive view of the system, especially for complex systems. These limitations emphasize the need for a more comprehensive and automated approach to PRE, one that leverages the power of intelligent systems to enhance the capabilities of human experts and streamline the process of eliciting, analyzing, and mitigating privacy risks.

B. NATURAL LANGUAGE PROCESSING APPLIED TO USER STORY ANALYSIS

A user story is a brief description of a user's requirements in agile software development. It follows a semi-structured template, recommended by Cohn [33], to ensure clear communication between the development team and the customer. A user story is often utilized to express functional requirements, but it can also be used to express non-functional requirements, such as regulatory requirements [9], [17].

The consistent structure of user stories also allows NLP to assist with tasks such as creating models, identifying ambiguities and defects, understanding the structure of the story, and promoting traceability [27]. Ahmed et al. [26] applied NLP to identify quality attributes from user stories. This was performed by using regular expressions to match the user story pattern with quality attributes. However, the study [26] did not consider security and privacy criteria. In terms of security, NLP has been used to help analysts uncover security-related attributes in product backlogs, as shown in the study by Galster et al. [34].

The use of NLP in detecting privacy-related information in user stories was demonstrated by Casillo et al. [35] and Herwanto et al. [36]. They utilized NLP to assess the potential disclosure of personal data in user stories and its compliance with privacy requirements. Privacy-relevant keywords and entities, such as personal data, data subjects, and processing entities, were considered in the analysis [35], [36]. Herwanto et al. [36] also employed NLP techniques, specifically Named Entity Recognition, to extract privacy entities from user stories. They noted the potential for further privacy analysis through the examination of user stories. Nevertheless, none of this work is integrated into the full process of PRE.

System characterization is also a main challenge when conducting privacy impact assessment [7], or privacy threat modeling [12]. Although not explored specifically in the security and privacy domain, the automatic modeling of the system by NLP that is sourced from user stories has been explored by some researchers [27]. Data Flow Diagram (DFD) is the main model used in privacy threat modeling [6]. The recent work from Herwanto explores the possibility of generating a DFD directly from the text of user stories [37]. Aside from DFD, transforming user stories into conceptual models is one of the tools explored by NLP [38]. Due to the time-consuming nature of system characterization, an automated modeling process is essential for agile teams. These studies also has not been integrated into the PRE process. Our aim is to streamline these processes and integrate them into the full cycle of PRE in order to elicit a comprehensive list of privacy requirements. Additionally, we provide a tool that allows for human intervention in cases where inaccuracies in the NLP process may occur.

III. THE PROPOSED PRE MODEL

This section presents our proposed PRE approach and the corresponding ML-based NLP model assigned for each PRE activity. Figure 1 illustrates the workflow of our model, which is divided into four main steps. The workflow shows that automation is carried out alongside human tasks performed by agile teams, as indicated by the white box in the figure. The automation tools are represented by the blue box, and the primary engine of these tools is the NLP models, depicted in the yellow box. These NLP models will be discussed in the following sections. Our approach incorporates the use of machine learning (ML) to perform the automated quality check, detection of privacy disclosure and generation of data flow diagrams in user stories. While we recognize that a certain degree of human interaction will always be necessary due to the ambiguity of natural language, ML can help analysts by highlighting privacy-related sections of user stories.

A. ELICITATION OF USER STORIES

Our approach is centered on user stories as the primary artifacts of agile requirements engineering. Therefore, instead of using questionnaires and forms to collect information about the system in terms of assets and data flow, we aim to use NLP to collect information from the user stories. The initial phase of the approach focuses on writing out the functional requirement in the form of a well-formed user story. Agile teams do not need to deal with a complete picture of the system to create privacy requirements using our approach, according to the lean and iterative nature of ASD. These user story sets constitute the basis for the next process of the PRE approach.

1) AUTOMATED QUALITY CHECK OF USER STORIES

The free-form nature of user stories can result in poor-quality user stories. This problem obviously affects our NLP engine, especially when generating data flow diagrams. As a result, an automatic quality check is necessary before proceeding to the next steps. Our approach works best on the well-syntactically structured user stories suggested by Cohn's template [33]. There are two well-known quality standards for user stories, called INVEST (Independent, Negotiable, Valuable, Estimable, Scalable, Testable) and QUS (Quality User Story) [39]. In this research, we use the latest approach, QUS, proposed by Lucassen et al. [39]. QUS proposes 13 criteria for user stories in terms of syntactic, semantic, and pragmatic quality. The authors also provide a system called AQUSA[1] (Automatic Quality User Story Artisan), which uses an NLP model to automatically check the quality of user stories. AQUSA is able to automatically check 5 of the 13 quality criteria proposed in QUS. These qualities are (1) well-formed, (2) atomic, (3) minimal, (4) unique, and (5) uniform. The first three qualities are associated with the syntactic quality of user stories, while the last two are associated with the pragmatic quality. These five automatic checks are considered sufficient for our approach. Therefore, we integrate AQUSA into the first step of our approach.

[1] https://github.com/RELabUU/aqusa-core

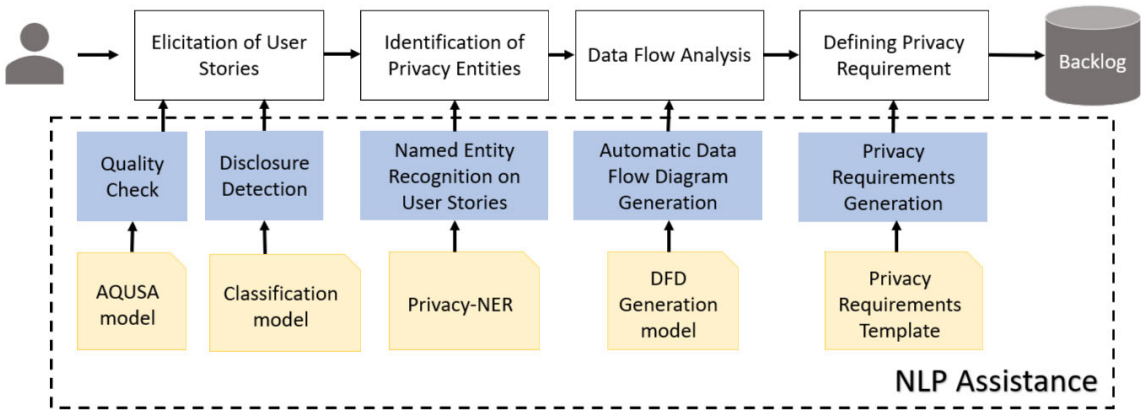

FIGURE 1. Overview of the proposed workflow for privacy requirements engineering. The white box indicates that the process is performed manually, and the blue box indicates that the process is performed automatically with the support of the NLP module in the yellow box.

2) DETECTING PRIVACY DISCLOSURES IN USER STORIES

The functional requirement written in the user story may potentially contain personal data that is subject to further generation of privacy requirements. To reduce the burden on the analyst to perform analysis on each user story, we propose to automatically determine which user story should serve as the basis for generating privacy requirements. We assume that whenever a user story contains information about the processing of personal data, a potential for disclosure exists. Consequently, the automated model must be able to detect the presence of this assumption. Therefore, we develop a supervised text classification approach that classifies which requirements are eligible for further processing in the PRE processing.

Previous research on privacy is mainly based on a list or dictionary of verbs (such as "access" or "ensure") or nouns (such as "data" or "database") to identify the disclosure signals [35]. However, relying only on the existing dictionary could lead to false positives due to loss of context. A supervised machine-learning approach that relies on human labels can overcome this limitation. Since the user story dataset is limited, previous studies have used prior knowledge from other domains, such as social media text, and applied it to the user story text using transfer learning [35]. We intend to build on the concept presented in these previous studies by enhancing the privacy words based on the privacy vocabulary [40] and annotating the privacy-related entities by human experts on each of the user stories. In addition, by embedding semantics in context, the machine learning model can understand not only the word in the dictionary but also its synonyms.

The architecture of our privacy disclosure classification system is illustrated in Figure 2. The process begins by converting user stories into embedding vectors. These vectors are a blend of general word embeddings and contextual embeddings, providing a comprehensive understanding of the text's semantics. The embeddings are concatenated and then fed through a document embedding process, which could be either convolutional or recurrent, to identify and condense local textual patterns into a single document vector. This vector is then fed into a fully connected layer, which employs sigmoid activation to classify the user stories as either disclosure or non-disclosure. Additionally, our model includes a mechanism that transforms individual sentences from the user stories into document embeddings using transformer technology. Further details on the experiments will be discussed in subsequent sections.

B. IDENTIFICATION OF PRIVACY ENTITIES

Our approach focuses on the early identification of the data subject, the personal data, and the processing. In Article 4, Definition 1 of GDPR outlined personal data as "*any information relating to an identified or identifiable natural person ('data subject'). An identifiable natural person is one who can be identified directly or indirectly*". Then, in Article 4, Definition 2, of the GDPR, the definition of processing is outlined as "*any operation or set of operations which is performed on personal data or on sets of personal data*". According to these definitions, we believe that identifying both the personal data and the data subject, as well as the processing that targets the personal data, are critical aspects of PRE. In our viewpoint, these entities can be spotted from the textual user requirements, including user stories. We propose an automatic detection of these entities through the use of Named Entity Recognition (NER). NER is an NLP task that involves identifying a collection of words or phrases associated with a specific Named Entity (NE) [41]. Therefore, we have built a named entity recognition (NER) system that targets three privacy-related entities: (1) personal data, (2) data subject, and (3) processing [36].

The NER model is treated as a form of supervised machine learning. As a result, the model requires ground truth data that has been manually labeled. The ground truth data is created by annotating the user story collection [42] with the three

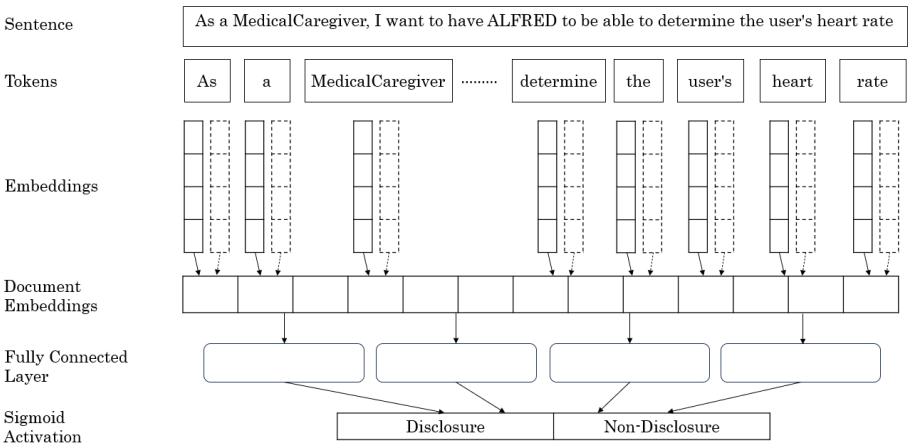

FIGURE 2. The architecture of the privacy disclosure classification.

entities mentioned above. Specifically, the first author took responsibility for the annotation due to limited resources. To ensure validity, the annotation is done according to the guidelines. The primary source for the guideline is the Data Privacy Vocabulary (DPV), which is an established ontology for privacy entities. The ontology includes the list of personal data categories, processing categories, and data subject categories, which is highly applicable to clarify the decision to annotate our selected privacy-related entities. The annotated data is available in our open repository.[2]

Due to the limited availability of annotated data in user stories, we utilize two data augmentation techniques: (1) synonym replacement and (2) mention replacement to enhance our training dataset. The method of synonym replacement involves substituting a given entity with a synonym. For example, the term ''email address'' in a user story could be replaced with its synonym ''electronic mail address.'' This process relies on synonym databases like WordNET and the Paraphrase Database (PPDB).

The second technique, ''Mention Replacement,'' involves substituting an entity with a different entity of the same category, randomly selected from our training data [43]. For instance, the Personal Data entity ''consent forms'' could be replaced with another Personal Data entity such as ''privileged access''. Unlike synonym replacement, where the meaning remains relatively the same, mention replacement introduces a variety of data within the same category. This method modifies the particular entity while maintaining the sentence's general structure, thus enhancing the dataset with varied yet pertinent instances. Table 1 contains an example of the augmentation process.

Alongside these methods, we incorporate non-user-story data sourced from the Data Privacy Vocabulary (DPV) ontology [40]. This involves integrating descriptions of various personal data categories from the ontology. We find this particularly beneficial for expanding the range of personal data entities covered in our dataset. The DPV ontology is specifically designed to standardize a broad spectrum of data privacy terminology, making it an ideal resource for enhancing our dataset with a more comprehensive array of privacy-related terms and concepts. The augmentation aims to improve recall, which is considered critical in automated tools for software engineering tasks [44].

Once the ground truth data is established, we make use of the state-of-the-art feature representation BERT to represent the user stories. BERT stands for Bidirectional Encoder Representation from Transformers [45]. BERT was pre-trained on the Mask Language Modeling (Mask LM) task and the Next Sentence Prediction (NSP) task, which enables it to comprehend the context of a word based on the preceding and following words, commonly referred to as bidirectional context. Due to this capability, we chose BERT for our NER tasks. Our neural network model will receive the extracted feature representations, allowing it to learn the probability of sequences that appear as privacy-related entities in the user story. The main neural network model we employ is a Bidirectional Long Short-Term Memory (BiLSTM) network, which provides context from both the previous and subsequent words in the user story. This context is deemed crucial due to the dependence of entities on neighboring words. Finally, a Conditional Random Field (CRF) layer is added to incorporate sentence-level tag information rather than information at individual positions [46]. The architecture of the model depicted in Figure 3.

C. DATA FLOW ANALYSIS BASED ON USER STORIES

Data Flow Analysis is an essential step in PRE as it highlights the flow of personal data in relation to functional requirements [22]. The privacy threat analysis framework, LINDDUN, uses Data Flow Diagrams (DFD) to visualize the system's process model and data flow, enabling the identification of potential privacy threats and informing the

[2]https://zenodo.org/doi/10.5281/zenodo.5801369

TABLE 1. Data augmentation example for named entity recognition to identify privacy-related entities.

Method	Story
Original	As a **parent** [Data Subject], I want to be able to **see** [Processing] which **consent forms** [Personal Data] I have **submitted** [Processing], so that I can know what I still need to do.
Mention Replacement	As a **repository manager** [Data Subject] I want to be able to **see** [Processing] which **privileged access** [Personal Data] I have **share** [Processing] so that I can know what I still need to do
Synonym Replacement (WordNet)	As a **parent** [Data Subject] I want to be able to **see**[Processing] which **consent forms** [Personal Data] I have **submit** [Processing], so that I can know what I still need to do
Synonym Replacement (PPDB)	As a **parenting** [Data Subject] I want to be able to **see**[Processing] which **consent formats** [Personal Data] I have **achieved submitted** [Processing], so that I can know what I still need to do

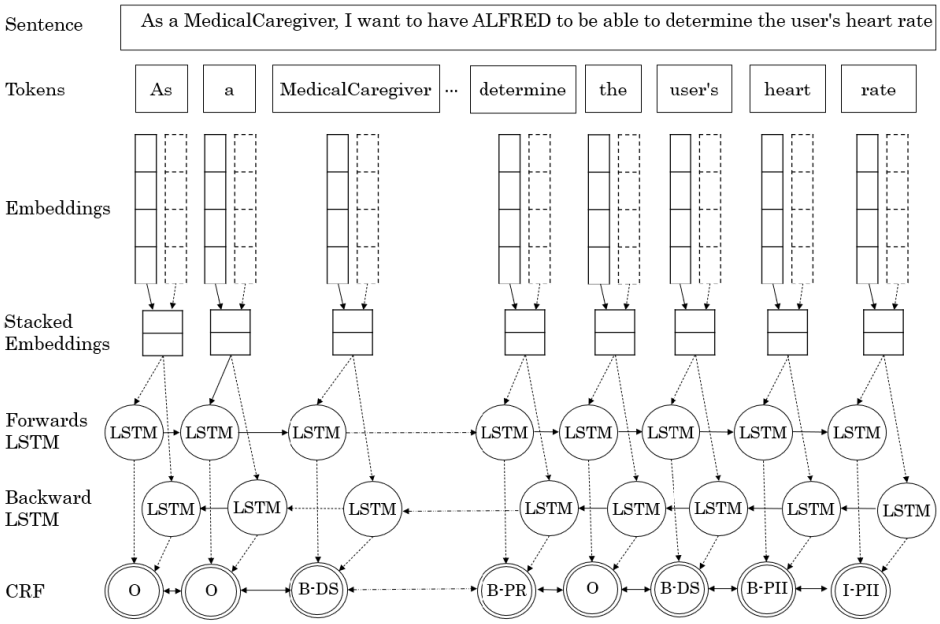

FIGURE 3. The architecture of the named entity recognition model originally introduced in [36].

creation of privacy requirements. DFDs are a concise and expressive modeling tool that is well-suited for the context of privacy threat analysis. However, the significant time and resources required to manually create DFDs pose a challenge for their practical use [7], [12], [47]. To address this, we have incorporated automation using NLP into our PRE approach. We have developed an NLP pipeline that automatically generates DFDs from textual user stories, as described in the research preview by Herwanto et al. [37]. In this section, we will elaborate on the work of Herwanto et al. [37] by focusing on the details of the DFDs, the transformation process from Robustness Diagrams (RDs) [48] to DFDs, and the improvement of the model by introducing an algorithm to transform the end parts of user stories, ensuring full coverage of information and preventing any missing information related to privacy entities. The workflow of automatically creating data flow diagrams can be seen in Figure 4. We publish our code and our results in the open repository.[3]

[3]https://zenodo.org/doi/10.5281/zenodo.5801350

1) THE DATA FLOW DIAGRAM (DFD)

The DFD consists of four elements: processes, data flows, data stores, and external entities. In security analysis tools such as STRIDE [49], an additional element called the trust boundary is included. In order to ensure the accuracy of DFD, certain syntax rules must be followed when drawing them. There are 5 rules stated in Ambler [50]: (1) All processes must have at least one data flow in and one data flow out, (2) All processes should modify the incoming data, producing new forms of out-going data, (3) Each data store must be involved with at least one data flow, (4) Each external entity must be involved with at least one data flow, and (5) A data flow must be attached to at least one process. These rules serve as a syntactic validation in our automation approach. We omit rule (2) for our approach due to the limitation of our information in the user story to be processed in the NLP model. Moreover, we are not focusing on the decomposition of the DFD level in this work, as we are generating entirely from the text of the user stories. The DFD which is used for threat modeling in both security [49] and privacy [6], also

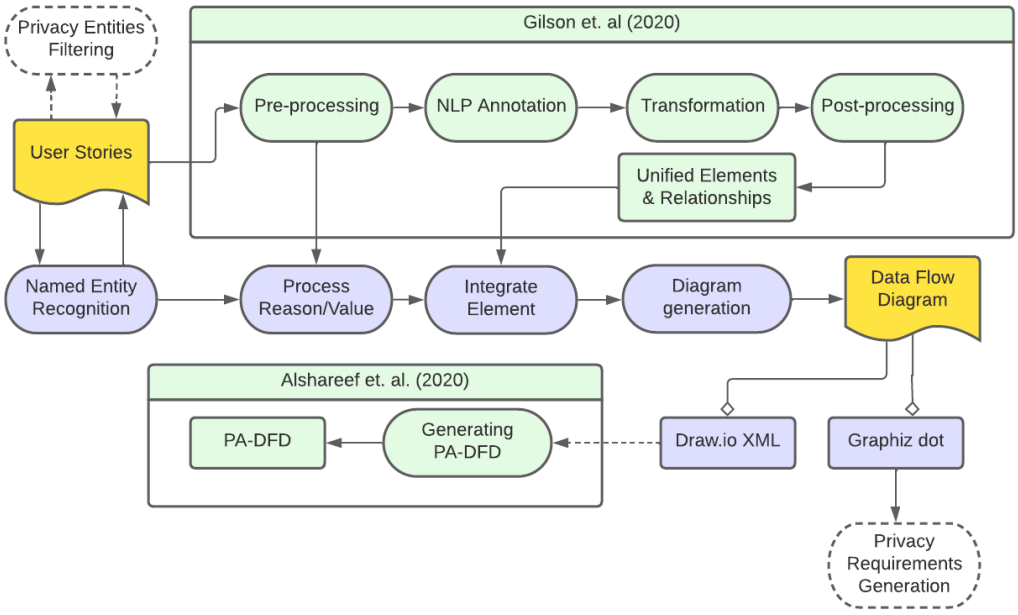

FIGURE 4. The workflow of the automated data flow diagram generation [37]. The main contribution is highlighted in purple, and the use of other work is highlighted in green. The robustness diagram to the data flow diagram can be seen in the connection to the work of Gilson et al. described in detail in section C.2. The "Process Reason/Value" is explained in section C.3 and composed of the existing robustness diagram from Gilson into DFD. The generation of the diagram is explained in section C.4.

ignores DFD decomposition. The most important part of the Data Flow Analysis with DFD is that risk areas or hotspots can be identified. The NLP can help identify concepts and relationships needed to transform a collection of user stories into a DFD.

2) FROM RDs TO DFDs

The approach to identifying concepts and relationships in user stories has been explored by Gilson et al. [48] to generate use case scenarios. The use case scenarios are depicted in the robustness diagram (RD) described in Rosenberg and Scott [51]. A robustness diagram can show a visual use case scenario diagram, showing the flow of the events. The objects in RD are grouped into actor, boundary, control, entity, and property. Due to the similarities of the RD with the DFD, we decided to adapt the model to our PRE pipeline. We use the model mainly to obtain the elements and relationships between the elements in the user stories. The elements may overlap with the entity from our NER model. Therefore, we use the information from the NER elements, such as the PII element, to indicate that the element as personal data, enabling visual hotspot detection directly from the DFD.

To generate the RD, the models separated the steps for generating concepts and relationships from user stories into four stages: (1) preprocessing; (2) NLP annotation; (3) transformation; and (4) post-processing [48]. The preprocessing steps are responsible for correcting pronoun and punctuation errors to improve the accuracy of further processes. After preprocessing, NLP annotation is carried out by part-of-speech (POS) tagging and dependency tree parsing[4] [48]. Based on the knowledge of POS and dependency between the elements, rule-based transformations are applied to understand the linguistic characteristics of the user story. These rules will guide the grouping of the element along with their relationships. A total of 23 rules were used for transformations [48]. Some of the most important rules are explained in the original paper [48]. An example of a simple rule: if a role part contains a noun (N) or a proper noun (PRP), then the N or PRP becomes the actor's object. There are additional rules not explained in the paper, such as *conjunct rule* that aim to track multiple items or processes located around conjunction words. For example, the sentence "create a registration form for both staff and kids" will be captured as two separate processes, which are "create a registration form for both staff" and "create a registration form for kids". Understanding the linguistic features allows the algorithm to categorize the words into the elements of the RD.

We have built a rule-based method that automatically transforms the RD into a DFD [37]. The graphical representation of the rule is depicted in in the Figure 5. Once the RD has been successfully created, it can be converted directly into DFD. The *actor* in the RD can be directly converted to an *external_entity* in the DFD. The *control* in the RD can be converted to a *processing* in the DFD. While the *entity* in the RD can become a potential *data_store* in the DFD [37]. However, several elements need to be omitted, such as *boundary* element in RD. Before omitting, the *boundary*

[4]https://gitlab.com/mde4asd/ucscenario-gen

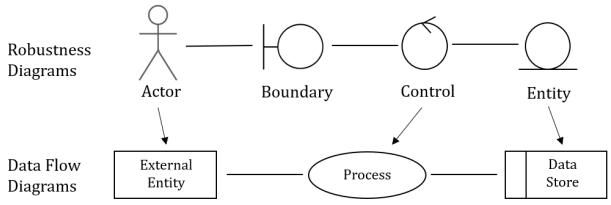

FIGURE 5. The transformation rule from RD to DFD [37].

element in RD can become the indicator to connect the data flow between the *external_entity* and the *process*. The data flow between the *process* and *data_store* in the DFD can be converted from the connection between *control* and *element* in the RD. DFD does not allow data flow between *data_store*, although it is still possible for *element* in the RD to have an interaction. To solve this, we use string matching to find out which control this element comes from and connect them.

Algorithm 1 Processing Reason

input : *userStory*
output: *actor*, *processing*, *dataStore*

role, *means*, *end* = `SplitUserStory` (*userStory*)
if *PII_Entity in* `NER` (*end*) **then**
  processing ← `GetVerb`(*end*)
  if *processing* = ∅ **then**
    actor, *processing*, *dataStore* = ∅
    return
  end
  if `GetSubject` (*end*) = *"I"* **then** *actor* ← `GetSubject` (*role*)
  else if `GetSubject` (*end*) = *"they"* **then** *actor* ← `GetSubject` (*means*)
  else *actor* ← `GetSubject` (*end*)
  dataStore ← PII_Entity
  return *actor*, *processing*, *dataStore*
else
  actor, *processing*, *dataStore* = ∅
  return
end

3) INCORPORATING THE "SO THAT" PART OF THE USER STORY

One drawback of the RD in the approach by Gilson et al. [48] is that they omit the *end* part of the user story that usually starts with "so that". The *end* of the user story usually provides a reason or value that can describe a non-functional requirement. Moreover, our observation found that some user stories might contain valuable information such as personal data. However, we only want to focus when the end part of the user story indicates containing functional requirements. It can be characterized by the existence of *processing* entity. Thus, we aim to capture this functional requirement and add it to DFD. We built a rule-based transformation described in Algorithm 1. The algorithm takes the user story as input and splits it to get the *end* part. We aim to capture three elements of the DFD based on the *end* part. These three elements can only be captured if only the *end* part contains *PII_Entity* and *processing* entity, which we are able to capture from the NER algorithm. Those two entities will become the *dataStore* and *processing*, respectively. The actor or external_entity will be decided by the subject in the *end* part as seen in the *GetSubject* function. There are three possible subjects; when they use "I," they refer to the subject mentioned in *role* of the user story. Meanwhile, when they use "they," the subject refers to the subject detected in the *means* part of the user story. Otherwise, the subject is detected from the *end* part of the user story. The output of this rule-based algorithm will then be integrated with the element and relationship produced by Gilson's et al. approach [48].

4) DIAGRAM GENERATION

The DFD can be produced in graphic format or draw.io format. The draw.io format enables the team to modify the DFD for further analysis. Optionally, the draw.io format enables the transformation from the DFD into Privacy-Aware DFD (PA-DFD), putting privacy checks directly in the DFD notation [52]. However, the subsequent process does not require PA-DFD and instead relies solely on the basic DFD for the generation of privacy requirements.

D. PRIVACY REQUIREMENTS GENERATION

In this phase, the understanding of data subjects, personal data, processing, and their interconnection in the DFD is established as the minimum knowledge requirement for recognizing potential privacy threats and requirements [6]. To build upon this knowledge, we propose an automated method for generating privacy requirements. Our approach is based on the best practice of using patterns to define NFRs in the ASD process [18]. Due to the commonness and universality of privacy requirements, we also propose using requirement patterns to define the template of the privacy requirement. The 12 categories of privacy requirements were selected from existing PRE approaches that have been evaluated and validated by the privacy research community. The first approach is ProPAN, which presents a comprehensive taxonomy of privacy requirements [22]. We also follow the refinements of the transparency [53] and intervenability [54] requirements that proposed in ProPAN. The second approach is the risk-based LINDDUN [6] which has demonstrated its practicality in real-world applications [12]. The detail of our selected privacy requirements with both approaches can be seen in Table 2.

We argue that all requirements, including privacy requirements, should be elicited in well-formed user stories to ensure traceability and clarity of user needs. We want to ensure that the user story conveys both the privacy requirement for each personal data processing and the achievable privacy values.

TABLE 2. Privacy requirements mapping.

Privacy Requirement	LINDDUN [6]	ProPAN [22]
Confidentiality	✓	✓
Integrity	-	✓
Availability	-	✓
Unlinkability	✓	✓
Anonymity	✓	✓
Pseudonymity	✓	✓
Undetectability	✓	✓
Transparency	✓	✓
Intervenability	-	✓
Plausible deniability	✓	✓
Content Awareness	✓	-

Hussain et al. [55] discovered that value interception can be incorporated into user stories, specifically in the "so that" part. As a result, we took the approach of framing the privacy threat to privacy values as one that can be avoided if the requirement is implemented.

The patterns require three main elements in DFD that are connected to each other. The DFD consists of a set of external entity, which in our case could be considered as data subject $S = \{s_1, \ldots, s_n\}$, a set of processing $P = \{p_1, \ldots, p_n\}$ and a set of data store, which also could become a personal data store $D = \{d_1, \ldots, d_n\}$. Triples are created whenever there are data flows that connect the three entities or whenever data store and subject connect to the same processing. We formalized the triples as $t_i = \{s_i, p_i, d_i\}$. Within one DFD there will be a set of triples $T = \{t_1, \ldots, t_n\}$. Each element of the triples will be subjected to 11 privacy requirements that will be explained in this section. Every privacy requirement will generate at least one user story. However, the requirements for Transparency and Intervenability can be more granular in that they lead to the creation of 2 and 3 distinct user stories, respectively. Thus, one triple will generate 14 possible privacy requirements $t_i = \{r_1, r_2, \ldots, r_{14}\}$. The detailed user story pattern can be seen in each of the following sub-sections of privacy requirements.

1) CONFIDENTIALITY

Confidentiality refers to the practice of maintaining the secrecy of personal data and preventing its disclosure to unwanted actors [22]. We took the privacy requirement confidentiality from Meis and Heisel [22] and modified it to become a user story. The confidentiality requirement (r_1) requires knowledge of the data subject, personal data, and processing.

> *confidentiality*(s_i, p_i, d_i): As a s_i, I want the d_i data that is processed in p_i to be kept confidential, so that unwanted actors are unable to access it.

2) INTEGRITY

In terms of privacy, integrity refers to the practice of maintaining the correctness and accuracy of personal data. The integrity requirement (r_2) is taken as inspiration from the ProPAN method [22].

> *integrity*(s_i, p_i, d_i): As a s_i, I want the d_i data that is processed in p_i to be prevented from faults or unwanted actors, so that the consistency and correctness of the data is not compromised.

3) AVAILABILITY

In terms of privacy, availability refers to maintaining the accessibility of the data. The availability requirement (r_3) also included the ProPAN method [22].

> *availability*(s_i, p_i, d_i): As a s_i, I want the d_i data that is processed in p_i to be prevented from faults or unwanted actors, so that the consistency, correctness, and availability of the data are not compromised.

4) UNLINKABILITY

Unlinkability is intended to avoid disclosing links between privacy-sensitive data outside the domain that share a common purpose and context [30]. The unlinkability requirement (r_4) requires knowledge of the data subject, personal data, and processing.

> *unlinkability*(s_i, p_i, d_i): As a s_i, I want the d_i data that is used in p_i to be protected from being linked directly or indirectly to other personal data within or outside of our system so that an attacker cannot link it to the identity of the subject in d_i data.

5) ANONYMITY

Anonymity requirements refers to the inability of unwanted actors to identify whether a subject exists in the system or not [22]. The anonymity requirement (r_5) requires knowledge of the data subject, personal data, and processing.

> *anonymity*(s_i, p_i, d_i): As a s_i, I want that the d_i to be anonymized (or pseudonymized) when performing p_i data, so that unwanted actors cannot directly or indirectly identify subject in d_i data.

6) PSEUDONYMITY

The requirements for pseudonymity (r_6) refer to the practice of concealing one's true identity by using an alternative name, commonly known as a pseudonym [22]. The decision to anonymize or pseudonymize is made by the developers, who decide based on the utility of the personal data.

> *pseudonymity*(s_i, p_i, d_i): As a s_i, I want that the d_i to be pseudonymized when performing p_i data, so that unwanted actors cannot directly or indirectly identify subject in d_i data.

7) UNDETECTABILITY

Undetectability refers to the inability of unwanted actors to determine whether an item of interest (IOI) exists in the system or not [22]. We took inspired from Meis and Heisel [22] that defined the requirement pattern for undetectability, and we reform it as the user story. The undetectability

requirement (r_7) requires knowledge of the data subject, personal data, and processing.

undetectability(s_i, p_i, d_i) : As a s_i, I want unwanted actors to be unable to sufficiently distinguish whether or not d_i data is present, so that I can safely perform p_i.

8) TRANSPARENCY

Transparency is focused on informing (r_8) data subjects about how and why their personal data is processed [22]. The transparency requirement requires knowledge of the data subject, personal data, and processing.

transparencyInform(s_i, p_i, d_i): As a s_i, I want to be informed and consented that the d_i data is used in p_i, so that I can exercise my rights when it is used outside of this context.

The transparency requirement (r_9) also mitigates the policy and non-compliance threat mentioned in LINDDUN. In addition, GDPR also mandated the right of access for the data subject in Article 15. Inspired by Bartolini et al. [17], we formulate transparency to access the personal data as:

transparencyAccess(s_i, p_i, d_i): As a s_i, I want to download a copy of d_i data that is used in p_i at camp, so that I can check their correctness.

9) INTERVENABILITY

Intervenability requires the controller to implement procedures that permit data subjects to exercise control over the timing, manner, and purpose of processing their personal data [22]. In addition, Hansen et al. [30] mentions that intervenability includes the right of rectification and erasure of data, as mentioned in GDPR Article 16 and 17. Inspired by Meis and Heisel [22], we formulate three intervenability requirements that focus on the ability to modify data (r_{10}), delete data (r_{11}), and withdraw consent (r_{12}). All of them requires knowledge of data subject, personal data, and processing. Thus, the pattern can be formulated as:

intervenabilityModify(s_i, p_i, d_i): As a s_i, I want to be able to modify the d_i data that have been processed at p_i without undue delay, so that I can prevent the inaccuracy of data.

intervenabilityDelete(s_i, p_i, d_i): As a s_i, I want to be able to delete the d_i data that have been processed at p_i without undue delay, so that I can exercise my right.

intervenabilityWithdraw(s_i, p_i, d_i): As a s_i, I want to withdraw my consent on the processing of p_i to the d_i data, so that I can exercise my right.

10) PLAUSIBLE DENIABILITY

Plausible deniability (r_{13}) refers to the ability to deny having undertaken an activity that other parties cannot confirm or disprove [6].

plausibleDeniability(s_i, p_i, d_i): As a s_i, I want to have the ability to deny performing p_i on d_i data, so that unwanted actors unable to accuse me of doing such a thing.

11) CONTENT AWARENESS

Content Awareness refers to educating the user of the system to be aware of the privacy-sensitive data in their system, and educate them to not overly share their privacy-sensitive data outside of the system [6]. The pattern of the content awareness requirement (r_{14}) only requires the data subject and personal data parameter.

contentAwareness(s_i, d_i): As a s_i, I want to be informed that I should not share the d_i data outside of the platform, so that my privacy or data subject in d_i data is not compromised.

E. PRIORITIZATION OF PRIVACY REQUIREMENTS

The generation of privacy requirements resulted in 14 possible requirements in a triple. We can consider all of the requirements to be privacy requirements because they process personal information. Nevertheless, analysts can choose which privacy requirements to select according to their privacy objectives. It is also possible that the selection of one privacy requirement overrides the other requirements.

Several studies have examined the conflict between security and privacy requirements [22], [30]. Hansen et al. [30] introduced the three pairs of conflicting privacy goals: (1) Confidentiality and Availability, (2) Integrity and Intervenability, and (3) Unlinkability and Transparency. These six protection goals are present in our approach, as seen in the previous section.

To mitigate the conflict between these goals or requirements, it is first necessary to establish Role-Based Access Control (RBAC) corresponding to the primary roles of stakeholders within the system. As RBAC is presumed to be governed by the principal functionality or security aspect, it is not incorporated into our Privacy Requirements Elicitation (PRE) approach. Meis and Heisel [22] introduced a concept of stakeholders and counterstakeholders to identify who can access certain personal data in a process. In our generated requirements, we use only unwanted actors to indicate the person outside the trust boundary.

RBAC effectively manages conflicting privacy requirements. For instance, Confidentiality and Availability can concurrently be set as a requirement, with specific roles designated for data access, and unwanted actors prevented from disrupting data access and availability. The data owner or the subject can solely perform Intervenability, ensuring data integrity. Transparency in r_8 does not give the data directly to the subject and thus does not conflict with Unlinkability. However, Transparency in r_9 may conflict with Unlinkability if the data subject is ignorant of potential linkability. Consequently, we supplement this with content awareness in r_{14} to enhance the data subject's awareness of privacy disclosure.

Nevertheless, there is a prioritization that can be made between the requirements. The unlinkability, anonymity,

pseudonymity, undetectability and confidentiality requirement can potentially be overlapped with each other [22]. The analysts can choose how strong the protection of personal data is. According to Meis and Heisel [22], once personal data becomes confidential to unwanted actors, it should fulfill the unlinkability requirements. However, if the user desires stronger privacy, the user can enforce the unlinkability requirement along with the confidentiality requirements. According to Meis and Heisel [22], Unlinkability can also includes undetectability, anonymity, and pseudonymity requirements. Thus, one of these requirements can be chosen for a particular processing of personal data. Pseudonymity is the weaker data protection level and should only be chosen if the other aforementioned requirements cannot be met.

Plausible deniability may conflict with transparency and intervenability. If a user desires plausible deniability in a particular process, there should be no traces of the data subject conducting a certain process. This would make transparency (r_7) and all intervenability requirements less relevant. Moreover, this could potentially compromise integrity since the trace of change could not be detected. Nevertheless, plausible deniability is usually desired for very sensitive operations such as anonymous sender and recipient messages or electronic voting. Thus, it strongly depends on the purpose of the processing itself [6].

The requirement of anonymity and pseudonymity are not synonymous. Anonymity means that the personal data cannot be associated with the data subject. Pseudonymity, on the other hand, means that personal data can still be associated with the data subject. Depending on the objective of the processing, the analysts can choose whether they would prefer to have the personal data anonymized or pseudonymized.

The content awareness requirement can be considered a soft privacy requirement [6], so it can be considered a lower priority compared to the other hard privacy requirement [6]. Finally, all selected and prioritized requirements can be included in the backlog of privacy requirements.

IV. EVALUATING THE FEASIBILITY OF THE DEVELOPED NLP-BASED APPROACH

The evaluation of our approach focuses on the feasibility of applying the NLP model, including the ML component, in each step of privacy requirements engineering, especially the privacy disclosure detection model and automatic DFD generation. We did not evaluate the AQUSA [39] since we used the same model from the original work. We will conduct a new Named Entity Recognition experiment [36], where we exclude CamperPlus from the training data, as we intend to use it in our pilot example. This also applies in the disclosure detection model where we exclude CamperPlus data.

A. EVALUATION OF DISCLOSURE DETECTION

Disclosure Detection is the first stage in determining if a user story will be issued for developing privacy requirements. Therefore a high recall is desired to avoid missing potential privacy requirements [44]. We start by defining our ground truth and then move on to the experimental setup to find the best model for our PRE purpose.

1) ESTABLISHING THE GROUND TRUTH

Our approach involves treating disclosure detection as a binary text classification task, inspired by the work of Casillo et al. [35], who used text classification to identify user stories with potential disclosure. This model was initially trained on various online forums [56] to detect unintentional disclosure of personal information in user posts.

However, we recognized a limitation in the work of Casillo et al. [35], as the ground truth used for text classification was based on a computer-generated label. This label depended on two conditions: the presence of privacy-related words in a user story, based on a dictionary lookup, and the prediction results of the model trained on online forums. We believe that such ground truth generation isn't suitable for our purposes.

With this in mind, we looked for datasets where the ground truth is assigned by human experts. We chose Herwanto's Named Entity Recognition (NER) dataset [36], as detailed in table 3. Although this NER dataset uses the same user stories as Casillo et al. [35], each user story was manually labeled for the presence of privacy-relevant entities.

To adapt it to our text classification model, we developed a rule that a user story would be assigned a positive label, implying a privacy requirement, if any privacy-related entities were found within it. As shown in Table 3, this rule results in 38% of user stories receiving a positive label, establishing our ground truth.

2) EXPERIMENTAL SETUP

We conducted nine different experiments focusing on text classification, to evaluate the performance of different models using different token and document embeddings. Our goal was to identify the best-performing models. The first eight experiments focused on standard text classification tasks.

For the token embeddings in scenarios 1 to 6, we used a stacked embedding strategy for the textual data: the GloVe model [57], a general word embedding technique, and advanced transformer-based language models, namely BERT [45] and RoBERTA [58]. For document embeddings, we used three different neural network architectures to embed documents: CNN (Convolutional Neural Network), GRU (Gated Recurrent Unit), and LSTM (Long Short-Term Memory). CNNs can effectively capture local dependencies and identify key phrases that contribute significantly to the meaning of a text. The GRU and LSTM architectures are designed to enhance the model's ability to capture sequential data dependencies. GRUs simplify the learning process by using a reduced number of gates compared to LSTMs. This streamlined structure makes GRUs less computationally intensive while maintaining their effectiveness in learning sequence dependencies.

TABLE 3. The overview of the data set used in our evaluation [42]. The ann column denoted the number of user stories annotated by one of the privacy-related entities (either data subject, processing, or personal data attributes).

Project	Description	Ann	Tot	%
Planning Poker	The first version of the PlanningPoker.com	46	53	87%
University of Bath	The University of Bath's institutional data repository.	51	53	96%
Zooniverse	A platform for citizen science that enables everyone to assist with research tasks.	26	60	43%
Federals Spending	Online portal for disseminating open data on US government spending.	17	98	17%
Loudon	Land management information system for Loudon Country, Virginia	30	58	52%
Recycling	An online portal to help people recycle their waste.	31	51	61%
Open Spending	Website intended to provide a transparent overview of government spending.	15	53	28%
FrictionLess	Platform for discovering data insights	20	66	30%
Scrum Alliance	The initial version of the Scrum Alliance Website	26	97	27%
NSF	The redesign of National Science Foundation (NSF) website's	11	73	15%
DataHub	Data portal for searching, exchanging, and publishing data online	22	67	33%
MIS	Duke University's management information system	18	68	26%
CASK	A unified integration platform for Big Data.	38	64	59%
Neurohub	A web-based platform that allows researchers to store their data and collaborate with colleagues.	49	102	48%
ALFRED	A platform that is intended to assist elderly individuals with limited mobility in daily tasks	64	138	46%
BADCamp	Platform for conference registration and management	24	69	35%
RDA-DMP	Software for machine-actionable data management plans	22	83	27%
ArchivesSpace	An open source archives information management platform for maintaining and offering online access to archives, manuscripts, and digital objects.	10	57	18%
Duraspace	Software for controlling and distributing digital content that is free and open source.	10	100	10%
RAC DAM	Platform for Archivist	24	100	24%
CULRepo	Digital content management system for Cornell University	35	115	30%
		635	**1680**	**38%**

For scenarios 8 and 9, we experimented with a pure transformer-based method, which directly transforms sentences into document embeddings without the intermediary of traditional sequence-based approaches.

The last scenario in our study uses the Named Entity Recognition (NER) method proposed by Herwanto et al. [36]. In this approach, we aimed to identify entities in user stories, with a particular focus on the prediction of Personally Identifiable Information (PII) entities as determined by NER predictions.

All text classification experiments were performed using the Flair framework. Detailed information about the hyperparameters used in these experiments can be found in our repository.[5]

3) DISCUSSION OF EVALUATION RESULTS

The evaluation results in Table 4 show the performance of different combinations of token and document embeddings in detecting privacy disclosures. The combination of GloVe with CNN (#1) showed a good balance between precision and recall, resulting in a high F1 measure of 75.06%. Interestingly, stacking GloVe with RoBERTa (#2) slightly improved the F1 score, indicating a marginal improvement in the balance between precision and recall.

In contrast, using LSTM and GRU for document embedding (#3, #4, #5, #6) resulted in a decrease in performance compared to CNN. This suggests that CNN may be more effective for such tasks. BERT and RoBERTa as Transformer embeddings (#7 and #8) showed consistent and comparable results, highlighting the effectiveness of Transformers-based models.

Notably, the NER-PII approach with transformers (#9) showed high recall but lower precision, indicating its strength in identifying relevant instances but with a trade-off in precision. This highlights the different effectiveness and trade-offs of different embedding techniques and document models in detecting privacy violations.

Based on the experiments, we propose to implement two separate models. The first model is based on scenario number 9, which has a high recall and would ensure that minimal user stories are missed during the privacy requirements engineering process. The second model is based on scenario number 2, which can be used when analysts

[5]https://zenodo.org/doi/10.5281/zenodo.10474910

want to prioritize the user stories to be addressed in the first iteration.

TABLE 4. Privacy disclosure detection evaluation.

No	(Token) Embedding	Document Embedding	Precision	Recall	F1-measure
1	GloVe	CNN	79.40%	74.20%	75.06%
2	**GloVe+Roberta**	**CNN**	**78.16%**	**74.21%**	**75.09%**
3	GloVe	LSTM	70.17%	66.22%	66.52%
4	GloVe+Roberta	LSTM	76.51%	73.08%	73.48%
5	GloVe	GRU	71.53%	65.71%	65.83%
6	GloVe+Roberta	GRU	73.27%	67.73%	68.07%
7	BERT	Transformers	74.33%	69.13%	69.51%
8	RoBERTa	Transformers	73.48%	73.10%	73.21%
9	**NER-PII**	**Transformers**	44.5%	**92.8%**	60.1%

B. EVALUATION OF NAMED ENTITY RECOGNITION

Our objective is to identify the most effective model for recognizing privacy-related entities in user stories. In terms of the experimental setup and establishing a reliable standard, we utilize the dataset provided by Herwanto et al., specifically the version that omits the CamperPlus data [36]. For training our model, we have chosen a learning rate of 0.1 and a mini-batch size of 32. We have set the patience parameter to 3 and the maximum number of epochs at 200, ensuring that training concludes due to a plateau in performance improvements rather than simply reaching the maximum number of epochs. The optimization of the model is conducted using Stochastic Gradient Descent. We employ the FlairNLP[6] library for this training process. The evaluation result can be seen in Table 5.

1) DISCUSSION OF EVALUATION RESULT

Table 5 presents the results of an experiment on Named Entity Recognition (NER) using two different models, BERT and RoBERTa, which have been previously recognized for their effectiveness in this domain [36]. The experiment involved the application of data augmentation techniques, specifically synonym replacement (aug-wordnet, aug-ppdb) and mention replacement (aug-mr), to enhance the models' performance.

Upon examining the results, it's evident that the mention replacement technique (aug-mr) significantly improves the performance of BERT models across most metrics. Specifically, BERT with aug-mr outperforms its base and other augmented versions in terms of Precision (Pr), Recall (Rec), and F1-score (F_1) in all entity categories. For the RoBERTa model, however, the results indicate a different trend. Contrary to BERT, the RoBERTa base model outperforms its augmented versions, including the one with mention replacement. This is particularly evident in the Processing category, where the base version of RoBERTa achieves the highest scores in Precision, Recall, and F1-score. This suggests that the base RoBERTa model is already highly effective for NER tasks, and the mention replacement augmentation does not provide additional benefits in this case.

[6]https://github.com/flairNLP/flair/

In terms of synonym replacement (aug-wordnet, aug-ppdb), the improvements are mixed and less consistent across both models. While some metrics show improvement, these are not as pronounced as those observed with mention replacement. For example, BERT augmented with aug-wordnet exhibits a slight decline in its performance in the Data Subject category compared to its base version.

C. EVALUATION OF AUTOMATED DFD GENERATION

The DFD is rendered fully automatically from the text of the user story. It can be rendered based on the combination of user stories and individual user stories. In this evaluation, we focus on automatically generating individual user stories.

1) EVALUATION METRICS

The metric for evaluating DFD generation is based on the syntactic and semantic correctness of the generated DFD.

Syntactic correctness is defined as whether the elements of DFD is followed the legal rule defined by standard DFD notation [59]. We took the syntactic guidelines published by Ambler [50] to evaluate the syntactic correctness of our generated DFD. We determined the syntactic correctness by using two discrete values: whether it is syntactically correct (1) or not (0). Syntactically correct means that all elements of the DFD are present and connected according to the legal rule defined in Ambler [50]. Syntactically incorrect, on the other hand, means that either one of the elements of the DFD is missing or the connection is not properly established.

According to Harel and Rumpe [59], the semantics of DFD can be defined in several ways based on the goal of DFD itself. As our DFD is meant to illustrate the data flow between data subject, processing, and personal data, from which the analyst can derive a potential privacy threat and requirements, we will concentrate on how easily the evaluator can comprehend the language and syntax in DFD. Since the DFD may contain several information, we determined using three discrete values: whether the DFD is semantically correct (1), partially correct (0.5), or not correct at all (0). Semantically correct means that the evaluator is able to understand the DFD by using the correct syntax and language in the sentences used in each DFD element and to locate the position of the personal data. Partially correct means that only half or more of the DFD elements are correct according to the definition of semantically correct. Meanwhile, semantically incorrect usually occurs together with syntactic incorrectness or error in the NLP model, especially the dependency parsing.

2) EVALUATION SETUP

The preliminary evaluation of the DFD generation has been explored by Herwanto et al. [37]. The preliminary evaluation focuses on the CamperPlus project, which we will see in more detail in Section V as the example for the overall approach. For a more comprehensive evaluation, we took all the projects in Dalpiaz's dataset [42] and focused on user stories with

TABLE 5. Experimental results on named entity recognition.

		Data Subject			Processing			Personal Data		
		Pr	Rec	F_1	Pr	Rec	F_1	Pr	Rec	F_1
BERT	base	87.3	92.3	89.7	67.2	67.2	67.2	71.4	69.4	70.4
	aug-mr	**90.7**	**94.2**	**92.5**	67.7	71.9	69.7	**73.9**	**73.2**	**73.6**
	aug-wordnet	86.3	93.0	89.5	64.7	75.8	69.8	69.9	65.6	67.7
	aug-ppdb	88.0	93.6	90.7	68.6	66.4	67.5	70.8	65.1	67.8
RoBERTa	base	85.1	94.9	89.7	**67.4**	**77.3**	**72.0**	66.2	64.6	65.4
	aug-mr	85.1	91.7	88.3	75.2	68.8	71.8	72.8	58.9	65.1
	aug-wordnet	86.5	90.4	88.4	74.6	64.1	68.9	72.7	61.2	66.5
	aug-ppdb	85.4	93.6	89.3	74.0	68.8	71.3	71.9	66.0	68.8

TABLE 6. DFD evaluation based on syntactic and semantic correctness.

Data	Number of Stories	Syntactic	Semantic
Priv	635	87.25%	80.79%
Priv + QC	497	**88.93%**	**82.70%**

privacy-related entities. These user stories can be obtained from the same data we used in the NER evaluation and shown in Table 3. By focusing only on user stories that have privacy-related entities, we can align the evaluation with the semantic definition of the DFD.

3) DISCUSSION OF EVALUATION RESULT

Table 6 shows the evaluation of our automatic DFD generation model. The first row shows the evaluation based on 635 user stories based on Table 3. Meanwhile, the second row eliminates some user stories that did not fulfill the quality check according to AQUSA automatic checking [39]. The results show that by fulfilling the quality proposed by AQUSA, our DFD generation can produce better syntactic and semantic quality. The best results show that 88,93% DFDs generated are syntactically correct, and 82,70% are semantically correct.

The results show that syntactic and semantic errors are still possible, even if the user story has passed the AQUSA quality check. The error is caused by NLP parsing, such as failure to recognize the dependent clause and negation. This error results in incomplete sentences or grammar, which normally occurs in the process part of the DFD.

We also found that several automatically generated DFDs still required human intervention. Some PII is not explicitly mentioned in the stories. Consider a user story "As a parent, I want to be able to message my child's counselors, so that I can voice my concerns or check on my child's progress.". Our NER model identifies the message as processing but not PII, although it also functions as PII. Concerns and progress, however, are identified as PII, which implies the message itself. This example shows the drawbacks of our approach, which will obtain a literal word from the user stories, which sometimes requires further grouping and aggregation of the same attribute.

V. APPLICATION OF THE PRE MODEL

This section demonstrates the implementation of our model from the collection of user stories [42].

A. THE CAMPERPLUS PROJECT

We used the open-source project CamperPlus (*Camper+*)[7] as the pilot example for our approach. CamperPlus is a web-based application designed for camp administrators and parents responsible for managing camps and supervising their children. We chose CamperPlus because, based on prior research findings, it was deemed a privacy-sensitive project [36]. The user stories are accessible to the public via their repository page. The project includes 54 user stories that feature four actors:

- **Camp administrators** are responsible for the enrollment of campers, scheduling activities for the various camp groups, uploading consent forms, and many more.
- **Parents** can use the platform to register their children for camp, monitor the activities of their children while at camp, make a payment, and communicate with camp administrators and counselors.
- **Camp workers** are responsible for managing field work of camps such as reporting a necessary repairs, reporting camp supplies, and being in charge of camp and campers.
- **Camp counselors** are responsible for taking attendance of the children.

The camp administrator was mentioned most often with 35 stories, followed by the parents with 13 stories, the camp worker with 6 stories, and the camp counselor with one story. In addition to the actors mentioned above, there are a number of data subjects that are not mentioned in the actor part of the user story, but in the main functionality of the user story. For example, children as the campers is frequently mentioned but since they do not have the main functionality in the system, it is not mentioned in the actor part of a user story. Nevertheless, they are also important data subject to be considered in the privacy requirement. In GDPR Article 8.1 it stated that *"Where the child is below the age of 16 years, such processing shall be lawful only if and to the extent that consent is given or authorised by the holder of parental*

[7]https://github.com/Notabela/Camper-Plus

responsibility over the child". Thus the user story will have an additional detail when child is involved in the functional requirement. Therefore, it is necessary to extract these data subjects and the actions they perform.

B. AUTOMATED QUALITY CHECK OF USER STORIES

The AQUSA was utilized to conduct a quality analysis of the 54 individual user stories belonging to CamperPlus. The results of this analysis indicated that the AQUSA was able to identify a total of 7 syntactic and 5 pragmatic criteria problems within the user stories.

Among the syntactic criteria, one user story was deemed unwell-formed, four user stories were non-atomic, and one user story was not minimal. Non-atomic user stories are typically identified by the presence of conjunctions and may be broken down into two or more individual user stories.

The pragmatic criteria identified by the AQUSA included 1 instance of duplication and 4 user stories that were deemed non-uniform. It should be noted that the lack of uniformity was attributed to several user stories lacking the mean part, potentially resulting in confusion for the NLP model.

Additionally, there was a repetition of certain indicators identified, which could also lead to potential confusion for the NLP model. Table 7 provides an illustrative example of the user story quality check. These user stories can then be edited to meet the quality criteria, making the analysis more accurate as the process continues.

C. DETECTING PRIVACY DISCLOSURES

We apply the model to determine the disclosure probability of 54 individual user stories from Camperplus. We applied the NER-based model based on Section IV-A, and we found that all of the user stories have a disclosure potential. We decided to rely on the NER-based model since the time to vet out the false positive would not be a burden [60], especially with the tool that we will introduce in Section VI.

One of the user stories that have disclosure potential is shown in Figure 6. An example of a user story that has no disclosure potential is, *"As a camp administrator, I want to be able to create, modify rules that campers and camp workers has to follow."*. Each of the user stories that have disclosure potential will undergo to the next process.

D. IDENTIFICATION OF PRIVACY ENTITIES

The user stories that has been identified to have a disclosure potential will be going through NER prediction. The example of a user story to which the NER was applied is shown in Figure 6. The complete entities that are detected from the NER model can be seen in Table 8. Due to the nature of user stories, several identified data subjects have the same meaning such as child, kids, and children. This also happens in personal data entities such as parent information, parents, and guardian's information. The advantage of NER is, it can identify the roles not only in the *role* part of user story, but also in *mean* and *end*. Examples of these data subjects are

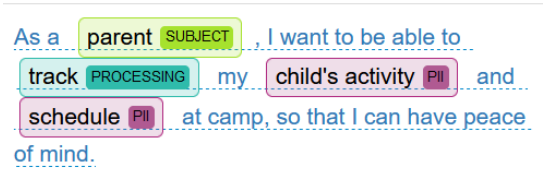

FIGURE 6. The example of named entity recognition on user stories.

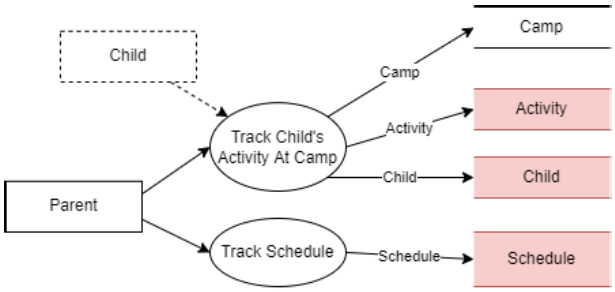

FIGURE 7. The data flow diagram for user story:"As a parent, I want to be able to track my child's activity and schedule at camp, so that I can have peace of mind."

manager, child, and staff member, which appear in the *means* of user stories rather than the *role*.

Here are two user stories involving data subjects who were identified as a child: (1) "As a camp administrator, I want to provide bi-weekly feedback to camper's parents, so that they can be aware of their child's behavior and performance at camp.". (2) "As a parent, I want to be able to track my child's activity and schedule at camp, so that I can have peace of mind". As can be seen, the primary *role* in the first user story is the camp administrator, and the second user story is the parent. However, from the privacy perspective, the child is the data subject that the personal data is compromised. Our NER model is able to capture these entities. The example of NER detection from the first user story can be seen in Figure 6.

E. DATA FLOW DIAGRAM GENERATION

The generation of DFD can be conducted in two ways. The first way is to generate DFD for each of the individual user stories. The example of the generation of DFD for a single user story can be seen in Figure 7.

The second way is to combine user stories to become one DFD. The combination can be based on particular filters such as the same data subject, the same personal data, or other filters that the analysts might be interested in. For example, consider user stories that have "child" as data subject below:

- As a camp administrator, I want to provide bi-weekly feedback to camper's parents, so that they can be aware of their child's behavior and performance at camp.
- As a parent, I want to be able to message my child's counselors, so that I can voice my concerns or check on my child's progress.
- As a parent, I want to be able to share any photos the camp has taken of my child.

TABLE 7. Example of user story quality check.

ID	User Story	Defect kind	User Story Modification
12	As a camp administrator, I want to be able to create, modify rules that campers and camp workers has to follow.	Atomic (Conjunctions)	(1) As a camp administrator, I want to be able to create rules that campers and camp workers has to follow. (2) As a camp administrator, I want to be able to modify rules that campers and camp workers has to follow.
14	As a camp administrator, I want to create an avenue so parents can submit feedback and general concerns, so that I can keep improving the services I provide.	Minimal (Indicator Repetition)	As a camp administrator, I want to create an avenue for parents can submit feedback and general concerns, so that I can keep improving the services I provide.
50	As a camp worker, I would be able to submit a completion report for the tasks which was assigned to me.	Well formed (No Means) & Uniform (As a, I want to, So that)	As a camp worker, I want to be able to submit a completion report for the tasks which was assigned to me, so that I can keep update my progress.

TABLE 8. Entities extracted from camperplus project.

Data Subject	Processing	Personal Data
camp administrator, campers, individual camper, parents, camper, staff members, camp worker, camp workers, parent, child, staff, kids, enrolled campers, camp management, counselors, children, manager, camp counselor	add, keep track, remove, keep, upload, schedule, automatically create, suspend, set a, assign, organize, warn, create, modify, submit, store, call, provide, make, see, notify, log, track, delete, read, sign up, assigned, message, voice, check, sign, deal, share, connect, enroll, edit, know, report, take	camper records, consent forms, camper submitted, forms, activities, who is where, nametags, who had behavioral problems, reminders, tasks, positions, responsibilities, rules, usage of internal camp facilities, feedback, general concerns, emergency information, immediate, guardian's information, registration form, information, medical forms, attendance, groups, camp groups, events scheduled, camp group, events, parents information, parents, groups information, child's activity, schedule, account, kids, concerns, payments, photos, children, enrolled children, completion report, list of supplies, necessary repairs, assigned kids

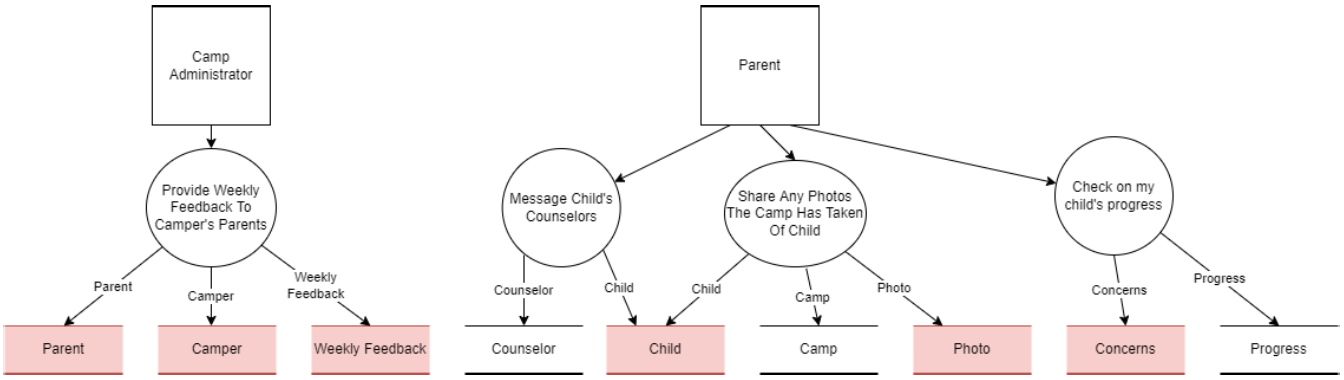

FIGURE 8. The data flow diagram that generated from three user stories that has "child" as data subject.

Our automatic generation can combine the three user stories into one DFD which can be seen in Figure 8. Based on the DFD, we can identify eight triples. However, we only consider the six triples that process personal data, which, based on the red highlight in the data store element. The processing that leads to the "Counselor" and "Camp" data is not counted as a triple that is considered when creating the privacy requirements. These six triples are further processed when creating the privacy requirement.

F. PRIVACY REQUIREMENTS GENERATION

Based on the model that been explained in Section III-D and DFD in Figure 8, it is known that the DFD triples are:

- $t_1 = (s_1, p_1, d_1)$
- $t_2 = (s_1, p_1, d_2)$
- $t_3 = (s_1, p_1, d_3)$
- $t_4 = (s_2, p_2, d_4)$
- $t_5 = (s_2, p_2, d_5)$
- $t_6 = (s_2, p_3, d_6)$

where s_1 = "Camp Administrator", s_2 = "Parent"; p_1 = "Provide Weekly Feedback to Camper's Parents",

p_2 = "Share Any Photos the Camp has Taken of Child", p_3 = "Check on my child's progress"; d_1 = "Parent", d_2 = "Camper", d_3 = "Weekly Feedback", d_4 = "Child", d_5 = "Photo" and d_6 = "Concerns".

Our approach will generate all requirements that stated in Section III-D The example of the generated privacy requirement for t_4 can be seen in Table 9. It is based on a user story: *"I want to be able to message my child's counselors, so that I can voice my concerns or check on my child's progress."*.

G. PRIORITIZATION OF PRIVACY REQUIREMENTS

The privacy requirement listed in Table 9 was created for a messaging action between a parent and a child's counselor to check the child's progress. Due to the sensitive nature of the child's data, strong privacy measures are needed. Firstly, the RBAC system should only allow the parent and counselor to access the message data to ensure confidentiality. The messaging process must also be accurate and timely, requiring specifications for integrity and availability. Maintaining the unlinkability of the child's data within the message is essential, meaning it should not be linked to any other personal data, such as weekly feedback or a photo of the child. End-to-end encryption of the message could achieve this. If the unlinkability requirement is enforced, it could override other requirements. Anonymization or pseudonymization of the child's identity is unnecessary since the unlinkability requirement already targets the message containing the data. The unlinkability requirement also overrides undetectability, as it should be difficult to determine if personal data is present in the message. Transparency and intervenability are critical to ensure the parent understands the purpose and limitations of the processing, even with end-to-end encryption in place. The parent must also be able to change their consent or the message itself. This helps the team comply with GDPR requirements, and plausible deniability would not be possible with these measures in place. Finally, content awareness could be added to prevent the sharing of the message by the parent as a preventive measure.

All of these prioritized requirements will be systematically added to the privacy backlog to ensure that each is appropriately addressed in the development cycle. This methodical approach facilitates a comprehensive and agile response to privacy concerns.

VI. SUPPORTING TOOL

We have developed a GUI-based tool[8] to support the workflow integration of our proposed approach, which was explained in Section III. The tool consists of 5 main modules [61] covering the five main functions of our PRE pipeline, as shown in Figure 1. A user can then perform the PRE in any of the user stories. The tools integrate our NLP model with a simple interface of a web-based system. The user interface of the tools can be seen in Figure 9. The video demonstration of the tools can be found at the following link.[9] The tool is called PrivacyStory [61]. The tool is a standalone web-based system built using the Flask Framework. The Flask Framework is a web development framework written in Python. Because it is written in Python, we can incorporate open-source tools from previous studies such as AQUSA [39] and Robustness Diagram Generation [62]. The tool allows users to perform specific interventions, such as editing the data flow diagram and the privacy requirement pattern. In addition, users can perform group analysis by filtering the user story based on privacy-related entities, such as data subjects or personal data entities.

VII. DISCUSSION

In this section, we revisit several critical issues that we have addressed in this paper. First, we briefly discuss the challenges faced by our approach in closing the gap through adopting PRE in agile software development (ASD). Next, we discuss in more detail the potential limitations that could result from our decision to rely only on user stories as input to PRE. We then examine the performance of the core NLP-based algorithms. Finally, we discuss the implications of our PRE approach.

A. ADOPTION OF PRE IN ASD

Integrating privacy engineering with agile methodology has proven to be a challenge, as reported in many studies [10], [14]. This challenge affects a wide range of activities throughout the development life cycle. One specific challenge is that the requirements for a system may change during the development process, which can impact the characterization of the system. As a result, the privacy requirements must be revisited whenever the assets that need to be protected or the type of processing change. To address this challenge, we propose using automation to review the requirements. Since the requirements are typically expressed in natural language text, we have used natural language processing (NLP) techniques to automate this process. Based on the approach and results of this research, we plan to conduct further evaluations in the form of a pilot study. The purpose of this pilot study is to assess the practicality of our method in a real-world agile environment.

B. USER STORIES AS INPUT FOR PRE

Our proposed PRE approach focuses on user stories, a widely-used method for agile teams to create functional requirements. This strategy has already shown its value and has led to significant changes in how user requirements are captured and identified. A well-known example of this approach is Checkland's and Poulter Soft Systems Methodology, which fully utilizes rich pictures to facilitate the requirement-gathering process [63]. NLP has been shown to be effective in a wide variety of different application scenarios. Therefore, we have based our methodology on NLP to

[8]https://zenodo.org/doi/10.5281/zenodo.7598314

[9]https://bit.ly/privacystoryvideo

TABLE 9. Privacy requirement generation result.

Requirement	User Story
Confidentiality	As a **parent**, I want the **child** data that processed in **message child's counselor** to be kept confidential, so that unwanted actors are unable to access it.
Integrity and Availability	As a **parent**, I want the **child** data that processed in **message child's counselor** to be prevented from faults or unwanted actors, so that the consistency, correctness, and availability of the data is not compromised.
Unlinkability	As a **parent**, I want the **child** data that used in **message child's counselor** to be protected from being linked directly or indirectly to other personal data within or outside of our system, so that an attacker cannot link it to the identity of subject in **child** data.
Anonymity or Pseudonimity	As a **parent**, I want that the **child** data to be anonymized (or pseudonymized) when performing **message child's counselor**, so that unwanted actors cannot directly or indirectly identify subject in **child** data.
Undetectability	As a **parent**, I want unwanted actors to be unable to sufficiently distinguish whether or not **child** data is present, so that I can safely perform **message child's counselor**.
Transparency	(1) As a **parent**, I want to be informed and consented that the **child** data is used in **message child's counselor**, so that I can exercise my rights when it is used outside of this context. (2) As a **parent**, I want to download a copy of **child** data that used in **message child's counselor** data at camp, so that I can check their correctness.
Intervenability	(1) As a **parent**, I want to be able to modify the **child** data that have been processed at **message child's counselor** without undue delay, so that I can prevent the inaccuracy of data. (2) As a **parent**, I want to be able to delete the **child** data that have been processed at **message child's counselor** without undue delay, so that I can exercise my right. (3) As a **parent**, I want to withdraw my consent on the processing of **message child's counselor** on the **child** data, so that I can exercise my right.
Plausible Deniability	As a **parent**, I want to have the ability to deny performing **message child's counselor** on **child** data, so that unwanted actors unable to accuse me of doing such a thing.
Content Awareness	As a **parent**, I want to be informed that I should not share the **child** data outside of the platform so that my privacy or data subject in **child** data is not compromised.

provide a solution that enables analysts to analyze user stories and identify potential privacy issues. However, the sole use of user stories can limit the trade-off between speed and change that agile methods require. Regarding generating privacy requirements, the completeness of requirements derived from user stories can become an issue. Other PRE approaches, such as ProPAN, require extensive user input to characterize the system [22]. We provide a much leaner approach by performing an automated PRE in each user story. Such a lean approach allows the team to have an early awareness of privacy requirements. In the future, conducting a specific study on whether a user story is sufficient to cover all privacy needs that arise in a given context may be possible.

C. THE USE OF NLP IN PRE

Natural Language Processing (NLP) for Requirements Engineering (NLP4RE) has been extensively studied and is effective in helping software developers with hairy tasks [60]. However, privacy engineering, which heavily relies on the requirements phase, remains a challenge for NLP4RE. This study uses established NLP models, particularly in text classification for disclosure detection, named entity detection for asset identification, and dependency analysis for data flow diagramming. This automation of PRE tasks has the potential to lead to unwanted consequences such as misinterpretation and bias. Despite using state-of-the-art deep learning models to train our model, we recognize that its performance still has room for improvement. The results of our feasibility analysis show that no model achieves perfect precision and recall, which is common for machine learning models. Research indicates that higher recall is desirable for tasks with significant requirements. In the case of the PRE process, we assume that a higher recall is preferable to minimize undetected privacy risks.

D. ENVISAGED IMPLICATIONS

The main foreseeable impact of our contributions is to increase the level of automation in dealing with privacy requirements in agile development projects. We see our model and prototype as enablers for a more efficient and, thus, more feasible analysis of privacy requirements. Beyond the impact of these focus contributions, the developed model can be adapted to other types of requirements. As NLP and especially NER have proven their value in our research context, this paper aims to motivate a much wider use of these technologies to support requirements engineering. While none of the existing solutions can provide a complete set of privacy requirements, a high level of automation is an

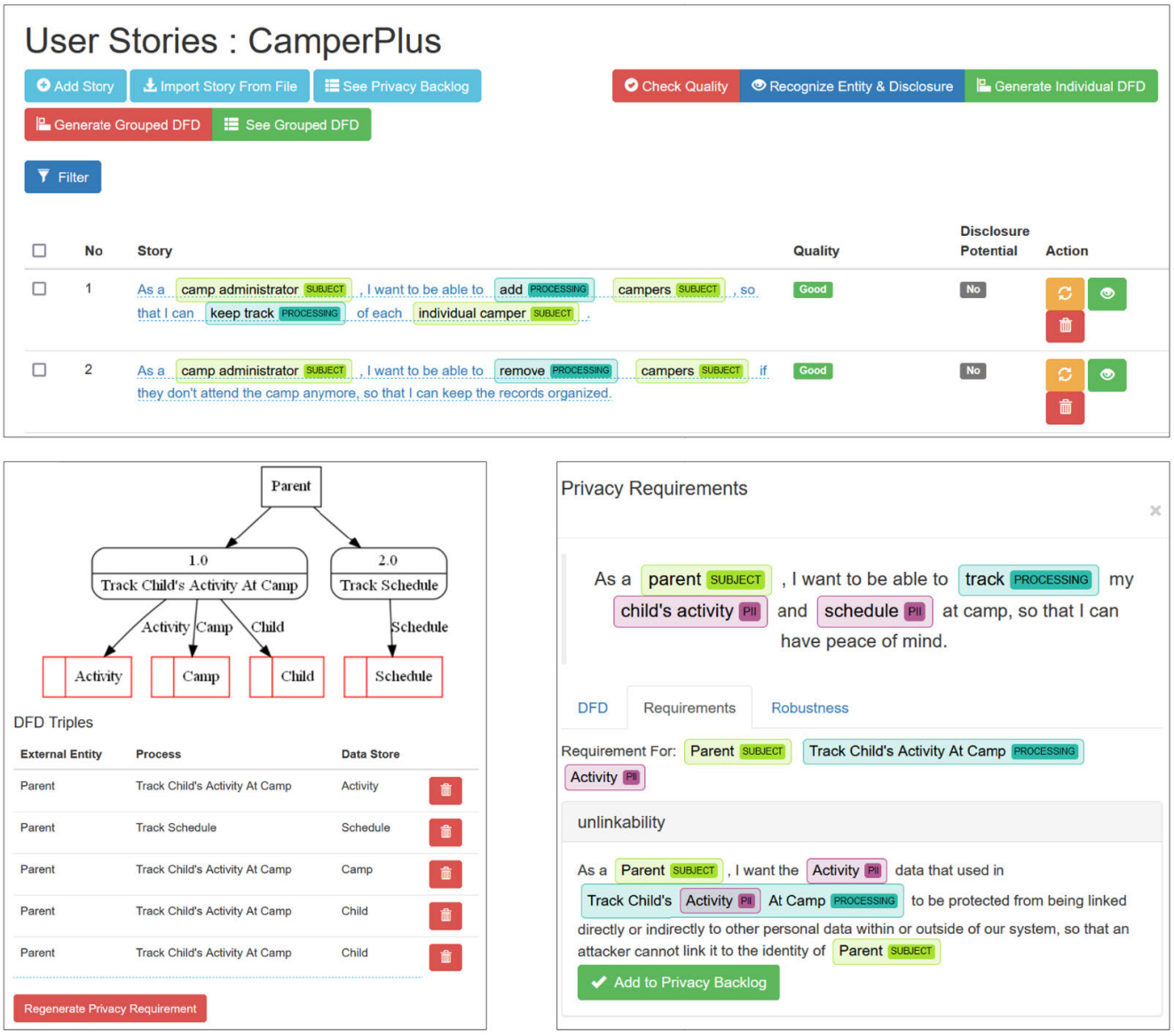

FIGURE 9. The three screenshots of the PrivacyStory tool. (1) The first screenshot on the top shows a project dashboard with several action buttons and the user story table. (2) The second screenshot on the bottom left shows an example of an automatically generated DFD along with the DFD triples. (3) The last screenshot on the bottom right shows an example of the unlinkability requirements generated from the first triple.

important step in freeing privacy professionals from being drowned in routine work and allowing them to focus on the requirements that require thorough legal review. On a more strategic level, our approach has demonstrated that fundamental gaps between different paradigms, such as agile approaches and the classic waterfall model, can be bridged. Moreover, the solution proposed in this paper could be seen as a template for dealing with legal issues that would otherwise significantly slow down agile development processes if not addressed.

VIII. THREATS TO VALIDITY

There are several threats to the validity of our research. Following the recommendation by Wohlin et al. [64], we divide the threat types into construct validity, internal validity, external validity, and conclusion validity.

Construct validity refers to the generalization of our experiment to the PRE process. The fact that we only use user stories as input to the PRE model may threaten the construct validity. In order to mitigate this potential threat to the validity of our results, we have considered the well-known PRE pipeline, such as LINDDUN [6] and ProPAN [22], and have applied our proposed NLP model within the pipeline of this framework. In addition, our NLP pipeline includes two external works on user story quality checking [39] and generating robustness diagram [48]. Limitations present in both studies may impact the construct validity of our studies. To minimize this, we have incorporated a human-in-the-loop interface that enables the review and modification

of the quality check results and generated DFDs, mitigating the potential impact of these limitations.

Internal validity of our research refers to whether the pilot example of our model and feasibility evaluation of the NLP model makes a difference in the validity. To minimize the internal validity, we took several approaches. Since our approach focuses on the privacy-sensitive project, we chose to use CamperPlus as the pilot example of our methodology. We took samples of a story to show the generated privacy requirements. Another user story in CamperPlus might generate different requirements that might be invalid, thus subject to manual validation. Nevertheless, our method can be used not only for privacy-sensitive projects. Thus, the feasibility of the NLP model was evaluated not only on the CamperPlus but also in the public dataset with 22 projects from different domains [42].

External validity refers to the generalization of our findings to another environment. The evaluation approach with 22 different projects was also chosen to mitigate threats to external validity. Nevertheless, our approach requires a well-formed user story, which might be difficult to implement in practice. We mitigate this by applying automatic quality checking. Since the model quality check relies on AQUSA, it relies on the completeness and validity of the AQUSA model. In the feasibility evaluation, we only show an overall result of the performance of each NLP model without specifying the individual projects to save space due to the number of our experiments.

Lastly, threats to the *conclusion validity* refer to problems that affect the ability to draw the correct conclusion about the relationship between treatment and outcome. To mitigate this threat, we conducted two evaluation treatments, which have been discussed concerning internal validity. The first evaluation focused on the feasibility of our NLP approach through statistical analysis based on ground truth labeled by human annotators. The statistical measures use F1-measure, which measures the harmonic mean between precision and recall. Since there is no benchmark to measure the significance of our NLP approach, we only compare it within the context of our internal experiments. In privacy disclosure detection, where we find the F1-measure unsatisfactory, we seek a method with the best recall measure, which is preferable in the hairy requirement task [60]. The second evaluation is based on a pilot example with CamperPlus, which follows our approach's step-by-step process that shows our model's real output.

IX. CONCLUSION

This paper introduces an agile approach to Privacy Requirements Engineering (PRE) that capitalizes on advancements in machine learning (ML)-based Natural Language Processing (NLP). Based on practical examples, it has been observed that numerous PRE tasks, such as system characterization, asset identification, and diagram modeling, can be quite demanding. Automation of these tasks could potentially enhance their completeness and facilitate the identification of prospective privacy concerns. We have chosen user stories as the main input for our approach because they are the primary artifact of agile methodologies such as Scrum. The semi-structured text of user stories also allows for exploration with NLP techniques. The feasibility evaluation of our NLP approach demonstrates its ability to supervise PRE tasks such as disclosure detection, entity detection, and data flow diagram (DFD) creation. The feasibility evaluation of the NLP model ensures the correctness of the generated privacy requirements. We also provide an example case of our approach in the CamperPlus project. Finally, we have developed a graphical user interface (GUI) tool to integrate our PRE pipelines and enable collaboration between requirement engineers and the automation approach to achieve the best possible outcome for privacy requirements.

The primary contributions offered by this research lies in automating the end-to-end process of PRE, taking a functional user story as input and delivering a privacy requirements user story as output. We assert that this seamless integration of user stories simplifies the adaptation of our PRE model for agile teams. Although the NLP solution utilized in our study doesn't introduce a novel NLP technique, it has been specifically adapted to align with our PRE approach. We anticipate that such incorporation of the NLP solution into requirements engineering will foster further advancements in the field of NLP4RE research.

To further validate our research, we plan to conduct a comprehensive case study that builds on our initial experimental results. We also plan to evaluate the usefulness of our PrivacyStory tool by presenting it to a development team that wants to integrate privacy requirements into their product. This will be a topic for future research. The results obtained through our research allow us to perform privacy requirements engineering in a distinctive way, freeing up extensive developer resources from pure search tasks. Extending our approach to other requirements engineering tasks potentially gives us a chance to fundamentally change the way requirement engineering is approached in practice today. Besides the expected high practical impact, our research substantially changes the requirement engineering process. We plan to evaluate the usability of our PRE approach in a controlled experiment with agile teams that want to integrate privacy by design into their process. More detailed observations and research need to be conducted to prove the practical usability of our PRE approach and our proposed tool. In addition, we plan to extend the tool to cover the solution in terms of Privacy Enhancing Technology and the Privacy Design Pattern. We believe that deploying ML-based solutions in the field of privacy engineering has the potential to automate the selection of appropriate design solutions for each of the privacy requirements.

ACKNOWLEDGMENT

The authors acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Programme.

REFERENCES

[1] A. Cavoukian, "Privacy by design—The 7 foundational principles—Implementation and mapping of fair information practices," Inf. Privacy Commissioner, ONT, Canada, Tech. Rep., 2009.

[2] S. Gürses and J. M. del Alamo, "Privacy engineering: Shaping an emerging field of research and practice," *IEEE Secur. Privacy*, vol. 14, no. 2, pp. 40–46, Mar. 2016.

[3] M. F. Dennedy, J. Fox, and T. R. Finneran, *The Privacy Engineer's Manifesto: Getting From Policy to Code to QA to Value*. Berlin, Germany: Springer, 2014.

[4] N. Notario, A. Crespo, Y.-S. Martin, J. M. Del Alamo, D. L. Metayer, T. Antignac, A. Kung, I. Kroener, and D. Wright, "PRIPARE: Integrating privacy best practices into a privacy engineering methodology," in *Proc. IEEE Secur. Privacy Workshops*, May 2015, pp. 151–158.

[5] S. J. De and D. L. Métayer, "PRIAM: A privacy risk analysis methodology," in *Data Privacy Management and Security Assurance* (Lecture Notes in Computer Science), vol. 9963, 2016, pp. 221–229.

[6] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, "A privacy threat analysis framework: Supporting the elicitation and fulfillment of privacy requirements," *Requir. Eng.*, vol. 16, no. 1, pp. 3–32, 2011.

[7] M. C. Oetzel and S. Spiekermann, "A systematic methodology for privacy impact assessments: A design science approach," *Eur. J. Inf. Syst.*, vol. 23, no. 2, pp. 126–150, Mar. 2014.

[8] E. Bozdag, "Privacy at speed: Privacy by design for agile development at uber," Tech. Rep., Jan. 2020.

[9] M. Miri, F. H. Foomany, and N. Mohammed, "ISACA: Complying with GDPR, an agile case study," *ISACA J.*, vol. 2, Apr. 2018.

[10] S. Gürses and J. van Hoboken, "Privacy after the agile turn," in *The Cambridge Handbook of Consumer Privacy*, 2018, pp. 579–601.

[11] R. Galvez and S. Gurses, "The odyssey: Modeling privacy threats in a brave new world," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops*, Apr. 2018, pp. 87–94.

[12] K. Wuyts, L. Sion, and W. Joosen, "LINDDUN GO: A lightweight approach to privacy threat modeling," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops*, Sep. 2020, pp. 302–309.

[13] R. Meis, "Problem-based privacy analysis (ProPAn)—A computer-aided privacy requirements engineering method," Ph.D. thesis, 2018.

[14] B. Kostova, S. Gürses, and C. Troncoso, "Privacy engineering meets software engineering. On the challenges of engineering privacy by design," 2020, *arXiv:2007.08613*.

[15] X. Wang, L. Zhao, Y. Wang, and J. Sun, "The role of requirements engineering practices in agile development: An empirical study," in *Requirements Engineering*, D. Zowghi and Z. Jin, Eds. Berlin, Germany: Springer, pp. 195–209.

[16] W. Alsaqaf, M. Daneva, and R. Wieringa, "Quality requirements challenges in the context of large-scale distributed agile: An empirical study," *Inf. Softw. Technol.*, vol. 110, pp. 39–55, Jun. 2019.

[17] C. Bartolini, S. Daoudagh, G. Lenzini, and E. Marchetti, "GDPR-based user stories in the access control perspective," in *Quality of Information and Communications Technology*, M. Piattini, P. da Cunha, I. D. Guzmán, and R. Pérez-Castillo, Eds. Cham, Switzerland: Springer, 2019, pp. 3–17.

[18] A. Jarzebowicz and P. Weichbroth, "A systematic literature review on implementing non-functional requirements in agile software development: Issues and facilitating practices," in *Lean and Agile Software Development*, A. Przybyłek, J. Miler, A. Poth, and A. Riel, Eds. Cham, Switzerland: Springer, 2021, pp. 91–110.

[19] E. D. Canedo, A. T. S. Calazans, A. J. Cerqueira, P. H. T. Costa, and E. T. S. Masson, "Agile teams' perception in privacy requirements elicitation: LGPD's compliance in Brazil," in *Proc. IEEE 29th Int. Requirements Eng. Conf. (RE)*, Sep. 2021, pp. 58–69.

[20] F. Ramos, A. A. M. Costa, M. Perkusich, H. Almeida, and A. Perkusich, "A non-functional requirements recommendation system for scrum-based projects," in *Proc. SEKE*, 2018, pp. 148–149.

[21] M. Oriol, P. Seppänen, W. Behutiye, C. Farré, R. Kozik, S. Martínez-Fernández, P. Rodríguez, X. Franch, S. Aaramaa, and A. Abhervé, "Data-driven elicitation of quality requirements in agile companies," in *Quality of Information and Communications Technology*. Ciudad Real, Spain: Springer, 2019, pp. 49–63.

[22] R. Meis and M. Heisel, "Computer-aided identification and validation of privacy requirements," *Information*, vol. 7, no. 2, p. 28, May 2016.

[23] M. Peixoto, C. Silva, R. Lima, J. Araújo, T. Gorschek, and J. Silva, "PCM tool: Privacy requirements specification in agile software development," in *Proc. Anais Estendidos do X Congresso Brasileiro de Software, Teoria e Prática (CBSoft Estendido)*, Sep. 2019, pp. 108–113.

[24] T. Xie, "Intelligent software engineering: Synergy between AI and software engineering," in *Dependable Software Engineering. Theories, Tools, and Applications*, X. Feng, M. Müller-Olm, and Z. Yang, Eds. Cham, Switzerland: Springer, 2018, pp. 3–7.

[25] M. Perkusich, L. Chaves e Silva, A. Costa, F. Ramos, R. Saraiva, A. Freire, E. Dilorenzo, E. Dantas, D. Santos, K. Gorgônio, H. Almeida, and A. Perkusich, "Intelligent software engineering in the context of agile software development: A systematic literature review," *Inf. Softw. Technol.*, vol. 119, Mar. 2020, Art. no. 106241.

[26] M. Ahmed, S. U. R. Khan, and K. A. Alam, "An NLP-based quality attributes extraction and prioritization framework in agile-driven software development," *Automated Softw. Eng.*, vol. 30, no. 1, p. 7, May 2023.

[27] I. K. Raharjana, D. Siahaan, and C. Fatichah, "User stories and natural language processing: A systematic literature review," *IEEE Access*, vol. 9, pp. 53811–53826, 2021.

[28] A.-J. Aberkane, G. Poels, and S. V. Broucke, "Exploring automated GDPR-compliance in requirements engineering: A systematic mapping study," *IEEE Access*, vol. 9, pp. 66542–66559, 2021.

[29] M. T. J. Ansari, A. Baz, H. Alhakami, W. Alhakami, R. Kumar, and R. A. Khan, "P-STORE: Extension of STORE methodology to elicit privacy requirements," *Arabian J. Sci. Eng.*, vol. 46, no. 9, pp. 8287–8310, Sep. 2021.

[30] M. Hansen, M. Jensen, and M. Rost, "Protection goals for privacy engineering," in *Proc. IEEE Secur. Privacy Workshops*, May 2015, pp. 159–166.

[31] S. Samonas and D. Coss, "The CIA strikes back: Redefining confidentiality, integrity and availability in security," *J. Inf. Syst. Secur.*, vol. 10, no. 3, 2014.

[32] H. Rygge and A. Jøsang, "Threat poker: Solving security and privacy threats in agile software development," in *Secure IT Systems* (Lecture Notes in Computer Science), vol. 11252, 2018, pp. 468–483.

[33] M. Cohn, *User Stories Applied: For Agile Software Development*. Reading, MA, USA: Addison-Wesley, 2004.

[34] M. Galster, F. Gilson, and F. Georis, "What quality attributes can we find in product backlogs? A machine learning perspective," in *Software Architecture* (Lecture Notes in Computer Science), vol. 11681, 2019, pp. 88–96.

[35] F. Casillo, V. Deufemia, and C. Gravino, "Detecting privacy requirements from user stories with NLP transfer learning models," *Inf. Softw. Technol.*, vol. 146, Jun. 2022, Art. no. 106853.

[36] G. B. Herwanto, G. Quirchmayr, and A. M. Tjoa, "A named entity recognition based approach for privacy requirements engineering," in *Proc. IEEE 29th Int. Requirements Eng. Conf. Workshops (REW)*, Sep. 2021, pp. 406–411.

[37] G. B. Herwanto, G. Quirchmayr, and A. M. Tjoa, "From user stories to data flow diagrams for privacy awareness: A research preview," in *Proc. Int. Work. Conf. Requirements Eng., Found. Softw. Quality*. Cham, Switzerland: Springer, 2022, pp. 148–155.

[38] A. Gupta, G. Poels, and P. Bera, "Creation of multiple conceptual models from user stories—A natural language processing approach," in *Proc. Int. Conf. Conceptual Model.*, Salvador, Brazil, 2019, pp. 47–57.

[39] G. Lucassen, F. Dalpiaz, J. M. E. M. Van Der Werf, and S. Brinkkemper, "Visualizing user story requirements at multiple granularity levels via semantic relatedness," in *Conceptual Modelin* (Lecture Notes in Computer Science), 2016, pp. 463–478.

[40] H. J. Pandit, A. Polleres, B. Bos, R. Brennan, B. Bruegger, F. J. Ekaputra, J. D. Fernández, R. G. Hamed, E. Kiesling, M. Lizar, E. Schlehahn, S. Steyskal, and R. Wenning, "Creating a vocabulary for data privacy," in *Proc. Move Meaningful Internet Syst., OTM Conf.*, H. Panetto, C. Debruyne, M. Hepp, D. Lewis, C. A. Ardagna, and R. Meersman, Eds. Cham, Switzerland: Springer, 2019, pp. 714–730.

[41] A. Mikheev, M. Moens, and C. Grover, "Named entity recognition without gazetteers," in *Proc. 9th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 1999, pp. 1–8.

[42] F. Dalpiaz, "Requirements data sets (user stories)," Tech. Rep., 2018.

[43] X. Dai and H. Adel, "An analysis of simple data augmentation for named entity recognition," 2020, *arXiv:2010.11683*.

[44] D. M. Berry, "Evaluation of tools for hairy requirements and software engineering tasks," in *Proc. IEEE 25th Int. Requirements Eng. Conf. Workshops (REW)*, Sep. 2017, pp. 284–291.

[45] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[46] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*.
[47] D. van Landuyt, L. Pasquale, L. Sion, and W. Joosen, "Threat modeling at run time: The case for reflective and adaptive threat management (NIER track)," in *Proc. Int. Symp. Softw. Eng. Adapt. Self-Managing Syst. (SEAMS)*, May 2021, pp. 203–209.
[48] F. Gilson, M. Galster, and F. Georis, "Generating use case scenarios from user stories," in *Proc. IEEE/ACM Int. Conf. Softw. Syst. Processes (ICSSP)*, New York, NY, USA, Jun. 2020, pp. 31–40.
[49] A. Shostack, "Experiences threat modeling at Microsoft," in *Proc. CEUR Workshop*, vol. 413, 2008, pp. 1–11.
[50] S. W. Ambler, *The Object Primer: Agile Model-Driven Development With UML 2.0*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[51] D. Rosenberg and K. Scott, *Use Case Driven Object Modeling With UML*. Berlin, Germany: Springer, 1999.
[52] H. Alshareef, S. Stucki, and G. Schneider, "Transforming data flow diagrams for privacy compliance (long version)," 2020, *arXiv:2011.12028*.
[53] R. Meis, R. Wirtz, and M. Heisel, "A taxonomy of requirements for the privacy goal transparency," in *Trust, Privacy and Security in Digital Business*. Valencia, Spain: Springer, 2015, pp. 195–209.
[54] R. Meis and M. Heisel, "Understanding the privacy goal intervenability," in *Trust, Privacy and Security in Digital Business*, S. Katsikas, C. Lambrinoudakis, and S. Furnell, Eds. Cham, Switzerland: Springer, 2016, pp. 79–94.
[55] W. Hussain, M. Shahin, R. Hoda, J. Whittle, H. Perera, A. Nurwidyantoro, R. A. Shams, and G. Oliver, "How can human values be addressed in agile methods? A case study on SAFe," *IEEE Trans. Softw. Eng.*, vol. 48, no. 12, pp. 5158–5175, Dec. 2022.
[56] N. Mehdy, C. Kennington, and H. Mehrpouyan, "Privacy disclosures detection in natural-language text through linguistically-motivated artificial neural networks," in *Security and Privacy in New Computing Environments*, J. Li, Z. Liu, and H. Peng, Eds. Cham, Switzerland: Springer, 2019, pp. 152–177.
[57] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. EMNLP*, 2014, pp. 1532–1543.
[58] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
[59] D. Harel and B. Rumpe, "Meaningful modeling: What's the semantics of 'semantics?'" *Computer*, vol. 37, no. 10, pp. 64–72, Oct. 2004.
[60] D. M. Berry, "Empirical evaluation of tools for hairy requirements engineering tasks," *Empirical Softw. Eng.*, vol. 26, no. 6, pp. 1–77, Nov. 2021.
[61] G. B. Herwanto, G. Quirchmayr, and A. M. Tjoa, "PrivacyStory: Tool support for extracting privacy requirements from user stories," in *Proc. IEEE 30th Int. Requirements Eng. Conf. (RE)*, Aug. 2022, pp. 264–265.
[62] F. Gilson and C. Irwin, "From user stories to use case scenarios towards a generative approach," in *Proc. 25th Australas. Softw. Eng. Conf. (ASWEC)*, Nov. 2018, pp. 61–65.
[63] P. Checkland and J. Poulter, "Soft systems methodology," in *Systems Approaches to Making Change: A Practical Guide*, 2020, pp. 201–253.
[64] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Berlin, Germany: Springer, 2012.

GUNTUR BUDI HERWANTO received the bachelor's and master's degrees from Universitas Gadjah Mada, Indonesia. He is currently pursuing the Ph.D. degree with the Faculty of Computer Science, University of Vienna. He is a Lecturer with Universitas Gadjah Mada. He has presented at requirements engineering conferences and authored several peer-reviewed articles. His academic and professional interests include software development and natural language processing. He is deepening these areas and conducting research in the areas of privacy engineering automation and agile requirements engineering integration.

GERALD QUIRCHMAYR was born in Upper Austria. He received the degree in computer science and law. He is currently the Deputy Head of the Multimedia Information Systems Research Group, Faculty of Computer Science, University of Vienna, where he has held the position of a University Professor, since 2000. Prior to his current role, he has been affiliated with Johannes Kepler University Linz, the Karl-Franzens-University of Graz, and the University of Hamburg. Additionally, he has been an Adjunct Professor with the School of Information Technology and Mathematical Sciences, University of South Australia, since 2002. His academic journey predominantly revolves around the intersection of information systems in business and politics, emphasizing applications, security, and legal implications.

A. MIN TJOA received the Ph.D. degree in computer science from Johannes Kepler University Linz, Linz, Austria. He has been a Professor of Software Technology with the Vienna University of Technology, since 1994. With research expertise in data warehousing, business intelligence, cybersecurity, and environmental informatics. He has authored more than 200 peer-reviewed articles. He has held leadership roles in notable organizations, such as the Austrian Computer Society, the United Nations Commission on Science and Technology for Development, and the ASEAN European Academic University Network. In recognition of his work, he has been awarded the Honorary Doctorate from Czech Technical University in Prague and the Honorary Professorship from Hue University, Vietnam.