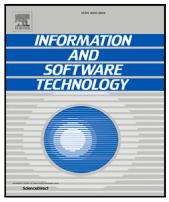
Information and Software Technology 146 (2022) 106853

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof



Detecting privacy requirements from User Stories with NLP transfer learning models☆

Francesco Casillo, Vincenzo Deufemia, Carmine Gravino *

Department of Computer Science, University of Salerno, Via Giovanni Paolo II, 132, Fisciano(SA), 84084, Italy

ARTICLE INFO

Keywords:
User Stories
Natural Language Processing
Deep learning
Transfer Learning

ABSTRACT

Context: To provide privacy-aware software systems, it is crucial to consider privacy from the very beginning of the development. However, developers do not have the expertise and the knowledge required to embed the legal and social requirements for data protection into software systems.
Objective: We present an approach to decrease privacy risks during agile software development by automatically detecting privacy-related information in the context of user story requirements, a prominent notation in agile Requirement Engineering (RE).
Methods: The proposed approach combines Natural Language Processing (NLP) and linguistic resources with deep learning algorithms to identify privacy aspects into User Stories. NLP technologies are used to extract information regarding the semantic and syntactic structure of the text. This information is then processed by a pre-trained convolutional neural network, which paved the way for the implementation of a Transfer Learning technique. We evaluate the proposed approach by performing an empirical study with a dataset of 1680 user stories.
Results: The experimental results show that deep learning algorithms allow to obtain better predictions than those achieved with conventional (shallow) machine learning methods. Moreover, the application of Transfer Learning allows to considerably improve the accuracy of the predictions, ca. 10%.
Conclusions: Our study contributes to encourage software engineering researchers in considering the opportunities to automate privacy detection in the early phase of design, by also exploiting transfer learning models.

1. Introduction

Requirements engineering (RE) is one of the most complex activity of software engineering. Misunderstandings and imperfections in the requirement documents can easily lead to design flaws and cause several problems [1,2]. Agile RE is based on face-to-face collaboration between customers and developers which helps to address several RE problems, but this does not exclude the presence of others. Among them, the detection of non-functional requirements (NFRs) by stakeholders is often a difficult activity due to several reasons [3]. To alleviate this problem, several solutions for the automatic detection of NFRs from text documents have been proposed [4–7]. For instance, Slankas et al. have proposed a tool-based approach, named *NFR Locator*, to extract sentences in unconstrained natural language documents, which are classified into one of the 14 defined NFR categories [7]. In general, these NFR detection tools provide only an overview of the identified NFRs. However, since stakeholders usually have expertise in few specific areas, they might have difficulties in defining all the features of a software application, increasing the risk of neglecting some of them [3].

Privacy is an essential NFR that needs special attention as business needs require data protection and safeguarding [8]. Even if privacy requirements frequently appear in software documentations, most of the time stakeholders ignore them. The difficulty of privacy requirement identification mainly depends from the quality of requirement specifications as shown in several studies (e.g., [9,10]).

In this paper we propose a deep learning approach to identify possible privacy requirements within User Stories (USs). The proposed solution aims to support practitioners, with poor privacy expertise, in the identification of NFRs related to privacy. Although a lot has been done in the field of privacy detection, to the best of our knowledge no study deals with the analysis of USs. Thus, we verify whether it is possible to exploit knowledge and tools proposed to address similar

☆ This work has been partially supported by the Italian Ministry of Education, University and Research (MIUR) under grant PRIN 2017 "EMPATHY: Empowering People in deAling with internet of THings ecosYstems" (Progetti di Rilevante Interesse Nazionale – Bando 2017, Grant 2017MX9T7H).

* Corresponding author.
E-mail addresses: fcasillo@unisa.it (F. Casillo), deufemia@unisa.it (V. Deufemia), gravino@unisa.it (C. Gravino).

https://doi.org/10.1016/j.infsof.2022.106853
Received 26 July 2021; Received in revised form 14 January 2022; Accepted 15 January 2022
Available online 30 January 2022

problems. With respect to conventional machine learning methods, the deep ones have unique advantages in feature extraction and semantic mining [11], and have achieved excellent results in text classification tasks [12–17]. Thus, from the analysis of user stories the deep learning models can infer individual privacy information and privacy rules, which can be used to recognize privacy-related entities for individual user stories. Then, the users can be reminded of the possibility of privacy leakage, based on the defined privacy rules.

The proposed approach combines the use of linguistic resources and Natural Language Processing (NLP) techniques to extract features useful not only to capture the semantic meaning and the syntactic structure of the text, but also to determine the presence or absence of privacy-related words. A further peculiarity of our approach is the use of Transfer Learning (TL), an emergent strategy where a system developed for a task is reused for a model on a different but related task [18–20]. Specifically, we use a pre-trained convolutional neural network (CNN) designed to identify personal, private disclosures from short texts [12] to extract features from user stories, which are combined with features obtained from a privacy dictionary to construct a US-privacy classifier.

To show the effectiveness of our approach, we present the results of an empirical study carried out by exploiting a dataset of 1680 user stories taken from [21]. In particular, we present a type of sanity check by formulating two research questions with the aim of verifying if a deep learning method (CNN) performs at least as conventional (shallow) machine learning methods, when exploiting NLP-based features (**RQ1**) or privacy word features (**RQ2**). The sanity check allows us to verify whether the further effort needed to apply CNN is payed back by an improvement in the prediction accuracy, and the possible contribution of PW features when applying shallow and deep learning methods.

The comparison between shallow and deep learning methods is often performed when evaluating text classification tools (e.g., [13–15]), mainly due to the possible noise in the data that can lead to substantial changes in the accuracy of decisions [13]. In particular, in some studies, shallow learning methods outperformed the deep ones in text classification tasks [15]. In our study, no clear result is obtained in the comparison when exploiting PW features (RQ2), thus confirming the importance of performing this kind of check. Differently, the results about RQ1 show that the deep learning method performs significantly better than the conventional machine learning methods, when exploiting NLP-based features.

After performing the two sanity checks, we investigate the proposed NLP-based Transfer Learning method by formulating a third research question (**RQ3**) aiming to compare its performances with those achieved using deep learning methods based on NLP-based features or privacy word (PW) features. The experimental results for RQ3 reveal an improvement of more than 10% (in terms of both Accuracy and F1-score [22]) compared to the individual CNNs.

Organization of the paper. Section 2 presents the research background on agile requirement engineering and how privacy is typically analyzed in this context. Section 3 describes the approach designed to identify privacy aspects in agile requirement specifications. Section 4 describes the design of the empirical study carried out to evaluate the approach. Section 5 reports on quantitative results and discusses the main findings. Section 6 concludes the paper and presents future research directions.

2. Related work

This section analyzes the different NLP techniques proposed in literature for US analysis. USs typically follow a structured format characterized by the *who*, the *what*, and the *why* of a requirement, becoming a standard de facto [23]. An example of user story defined by using the Cohn's model [24] is:

```
As a site member, I want to access to the Facebook
profiles of other members so that I can share my
experiences with them
```

Several frameworks and methodologies have been proposed for analyzing the quality of USs through their syntactic analysis, with the aim of making them more accurate and clear for customer's requirement definition [25–27]. Other relevant investigations concern with the transformation of USs into models and components useful for the next stages of the software development processes. In particular, software diagrams can be automatically generated from USs in order to provide a visual representation for project stakeholders, to identify conceptual entities, or to highlight potential problems in US definition [28–31]. These activities open the door to further automated analysis able to generate conceptual models [32], Use Case scenarios [33], and even Backlog Items [34]. USs can also be analyzed to automatically generate test cases [35] and create behavioral models that help testers who might be non-expert.

Other studies on USs focus on extracting attributes that can guide architecture design without relying on systematically and formally defined knowledge. For example, Gilson et al. show that USs might have a great impact on early stage decisions (because they might implicitly refer to quality attributes) allowing software architects to have an idea of the consequences of the possible design decisions [36]. In particular, they use machine learning (ML) techniques to classify if the USs refer to quality attributes and, if so, which ones they refer to.

Approaches dealing with other types of attributes are able to provide more detailed information, which improves the definition of customer requirements and facilitates decisions made during the software development process [3]. For example, Villamizar et al. define an approach for reviewing security-related aspects in agile requirement specifications with a focus on web applications [37]. Their results indicate significant differences when comparing the performance achieved by experts using their approach against other defect-based techniques. Similarly, Riaz et al. propose a ML-based tool that takes in input a set of natural language artifacts and automatically identifies (and classifies) security-relevant phrases according to predefined security objectives [38]. However, stakeholders may not be able to assess and define all aspects of a software application together with customers, increasing the risk of leaving out even high priority ones [3], as in the case of data privacy.

Many efforts have been devoted to privacy disclosure in the recent years, both to facilitate the work of analysts and developers [39,40] and to define a linguistic taxonomy of privacy for content analysis [41,42]. Many of the privacy detection approaches focus on the automatic recognition of sensitive personal information in unstructured text [12,16,43], which allows to develop several interesting tools, such as TABOO [17] and PrivacyBot [44]. On the other hand, many companies have particular needs with respect to personal data processing, and in the software design phase these needs may be set aside to make space for more functional requirements [45]. Therefore, the identification of privacy content can be considered crucial when building the architecture of a software system. However, to the best of our knowledge, nothing is proposed in the literature about the automatic identification of privacy content in the early stages of Agile software development. This work aims to fill this gap, by providing and evaluating an approach for detecting privacy information from USs.

3. A methodology for privacy disclosure detection within user stories

The proposed technique aims at identifying privacy-related threats in agile requirement specifications. The approach considers USs and linguistic resources as input, and exploits NLP techniques to determine the presence or absence of privacy-related words in the USs. The latter are structured in a sentence as follows [37]:

As a site member, I want to access to the Facebook profiles of other members so that I can share my experiences with them

Fig. 1. The US words highlighted in red are contained in the *privacy dictionary* defined in [42].

Table 1
The privacy category of the words 'access' and 'share' [41].

Category name (number of words)	Description	Example dictionary words
OpenVisible (2)	*open and public access to people*	*port, display, accessible*

`As a` [*role*]`, I want to` [*feature*]`, so that` [*reason*].

Although this structure simplifies US's comprehension, the detection of privacy disclosures may be ineffective due to a wide variety of possible terms in USs. Therefore, more advanced approaches are needed to improve its effectiveness.

The proposed method leverages convolutional deep neural networks to identify short texts of USs having private disclosures. In particular, we first adopt a lexicon-based approach to identify the words having entity-level privacy disclosures, by using the matches between USs and a privacy dictionary as machine learning features. This method can give high precision, but low recall since it relies only on the count of sensitive words in a document, without considering the context in which these words are used. To improve recall, we also exploit NLP tools to derive linguistic features, such as syntactic dependencies and entity relations, which keep the sentence level context into consideration.

The proposed deep neural network model combines together multiple channels to perform the disclosure/non-disclosure classification task. Each channel refers to different representations of the same candidate user story.

To deal with the paucity of curated data in the field, we propose the use of transfer learning, which allows to utilize knowledge acquired for one task to solve related ones. In particular, we exploit a pre-trained CNN that exploits NLP-based features to detect privacy disclosures in Reddit users' posts and comments. This neural network is trained on 10K disclosure and non-disclosure sentences.

In what follows we provide details of the features used for privacy disclosure detection (Sections 3.1 and 3.2), and the architectures of the considered deep neural network models (Sections 3.3 and 3.4).

3.1. Lexicon-based privacy disclosure features

These features are extracted from the text of the USs by using linguistic resources, i.e., dictionaries, containing individual words or phrases that are assigned to one or more linguistic categories. By using a privacy dictionary it is possible to count the occurrences of each dictionary word within a US text, incrementing the relevant categories to which the words belong [41,42]. The final result consists of values for each linguistic privacy category, represented as a percentage of the total words in the text.

We use the privacy dictionary proposed by Vasalou et al. [42], which constructed and validated eight dictionary categories on empirical material from a wide range of privacy-sensitive contexts. Experimental results have shown that the identified categories allow to effectively detect privacy language patterns within a given text. Figs. 1 highlights the two US words contained in the privacy dictionary defined in [41,42], whereas Table 1 reports the information on the privacy category *OpenVisible* they belong to.

3.2. NLP-based features for privacy disclosure

These linguistic features are obtained from the text of the USs by extrapolating entities, the parts of speech, and the dependencies between them, since the aim is to understand the text from its meaning and to capture those features that may affect the classification of USs as related to privacy disclosure.

Table 2
Parts-of-speech and dependencies extracted from a user story.

Text	Part of speech	Dependency
As	SCONJ	prep
a	DET	det
site	NOUN	compound
member	NOUN	pobj
I	PRON	nsubj
want	VERB	ROOT
to	PART	aux
access	VERB	xcomp
to	ADP	prep
the	DET	det
Facebook	PROPN	compound
profiles	NOUN	pobj
of	ADP	prep
other	ADJ	amod
members	NOUN	pobj
so	SCONJ	mark
that	SCONJ	mark
I	PRON	nsubj
can	VERB	aux
share	VERB	advcl
my	DET	poss
experiences	NOUN	dobj
with	ADP	prep
them	PRON	pobj

In what follows, we describe how these NLP-based features have been obtained for the US shown in Fig. 1 by using the NLP spaCy toolkit.[1] First, the text is pre-processed by removing punctuation and insignificant words, leaving only lexical items (*tokenization*). The result of this process for the considered US is: ['As', 'a', 'site', 'member', 'I', 'want', 'to', 'access', 'to', 'the', 'Facebook', 'profiles', 'of', 'other', 'members', 'so', 'that', 'I', 'can', 'share', 'my', 'experiences', 'with', 'them']

The Dependency Parser (DP) Toolkit[2] from spaCy has been used to extract information on syntactic relations and parts of speech (POS), which enable the data to be enriched with such information on syntactic and semantic structure. Table 2 reports the POSs and dependencies extracted from the user story of Fig. 1. These features help the model to understand the common sequence of tokens and the occurrence of dependency tags [46].

The Named Entity Recognizer (NER)[3] of spaCy has been used to assign labels to contiguous tokens. The default model provided by the library identifies various entities, such as companies, locations, organizations, and products, and new entities can be added to the system by updating the model with new data.

3.3. Deep neural network models

After doing all the necessary pre-processing steps, the data is then fed into a multi-input deep neural network to learn the hidden patterns and features to distinguish between texts having disclosure and non-disclosure occurrences. In particular, we constructed two deep

[1] https://spacy.io/
[2] https://spacy.io/usage/linguistic-features#dependency-parse
[3] https://spacy.io/usage/linguistic-features#named-entities

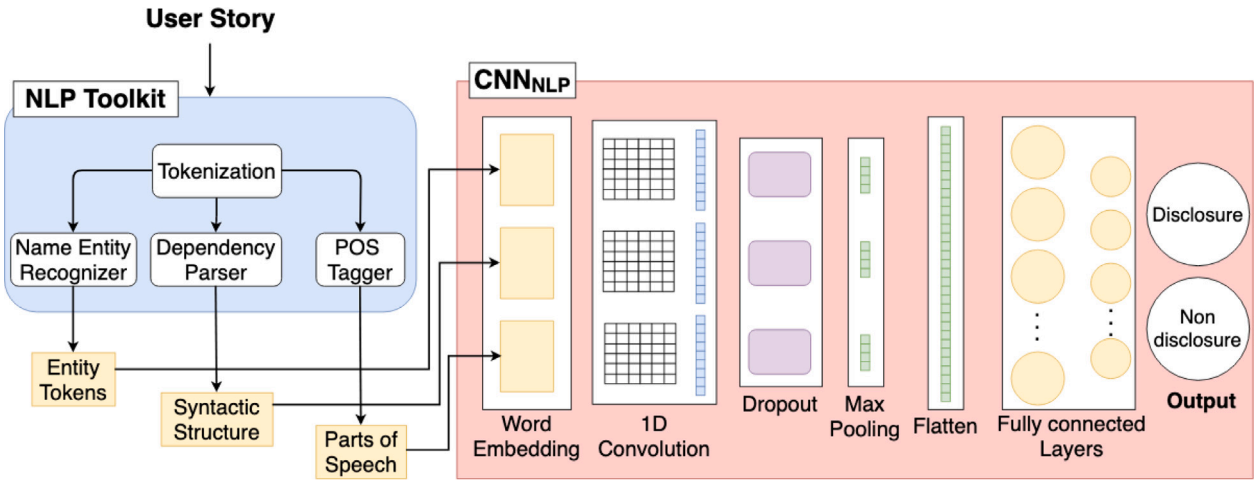


Fig. 2. The CNN_{NLP} architecture.

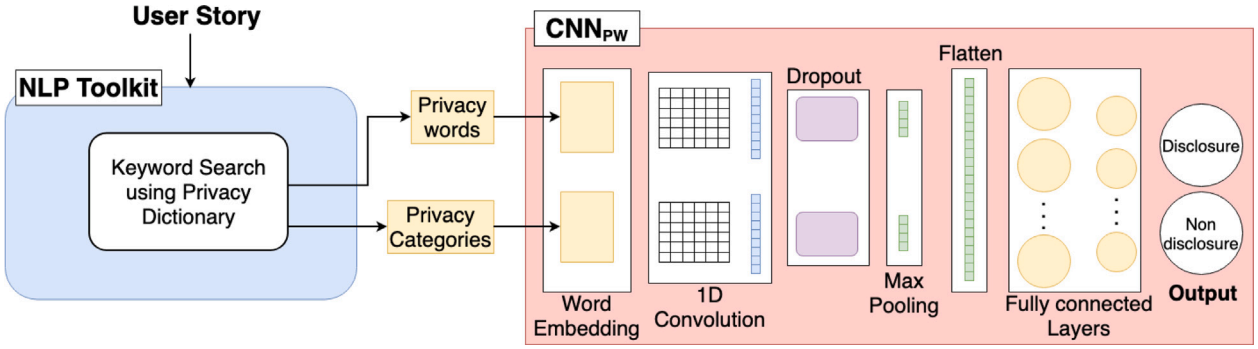


Fig. 3. The CNN_{PW} architecture.

convolutional neural networks, one based on the NLP-based features introduced in Section 3.2 (see Fig. 2), the other on the lexicon-based features of Section 3.1 (see Fig. 3). The first takes lexical (word tokens) features through one input, syntactical features (dependency parse tree information) through another input following a merging of those feature vectors. Later these vectors additionally get merged with supplemental (auxiliary) inputs before going through a further multi-layer perceptron stage. At the end of the deep neural network, a single neuron is used to provide the probability toward each of the above mentioned classes. The latter performs similar operations fed the features obtained from the privacy dictionary.

3.4. A transfer learning methodology for privacy disclosure detection

The previous deep neural networks require a specific dataset to train the model from scratch to the specific classification task. Unfortunately, the datasets of user stories available in literature contain few hundreds of examples. For this reason, it could be possible that the models are not able to adequately learn how to classify a US. To deal with this problem we introduce a neural network model that exploits transfer learning for the disclosure/non-disclosure classification task.

Transfer learning is an approach in which the knowledge learned from a large-scale dataset to solve a particular task is reused (transferred) and applied to solve a different but related task [18]. In particular, transfer learning allows to use pre-trained shallow/deep learning models by fine-tuning them on a relatively small labeled dataset from the downstream task.

In the proposed model, the NLP-based features described in Section 3.2 are processed by a pre-trained convolutional neural network whose aim is to identify short texts that have personal, private disclosures [12]. In particular, the neural network identifies whether the unstructured text given as input contains private disclosures by analyzing the semantic and syntactic structure of the text through the extraction of the characteristics described above, i.e., entities, dependencies, and parts of speech. This network has been trained for privacy disclosure classification on ten thousand Reddit users' posts and comments.

Fig. 4 shows the architecture of the deep neural network, named PD_{TL}, obtained by applying transfer learning. Taking advantage of the flexibility of the tools provided by Keras,[4] the pre-trained neural network proposed in [12] has been truncated after the Flatten layer. The latter is concatenated with the Flatten layer of another neural network that processes the lexicon-based privacy features. As a consequence, the resulting neural network processes the information concerning the semantic and syntactic structure, enriching this analysis with the information derived from the privacy dictionary.

4. Empirical study design

In this section we present the design of the empirical study we have performed. In particular, we first provide the research questions and the motivations behind their formulation. Then, data employed for the analysis is described, followed by the presentation of the validation methods. In the last part of the section, the evaluation criteria we adopted for assessing the predictions achieved with the built machine learning models and threats to validity discussion are presented. The data and scripts to train the models and reproduce the results may be found online at https://tinyurl.com/US-privacy.

[4] https://keras.io/

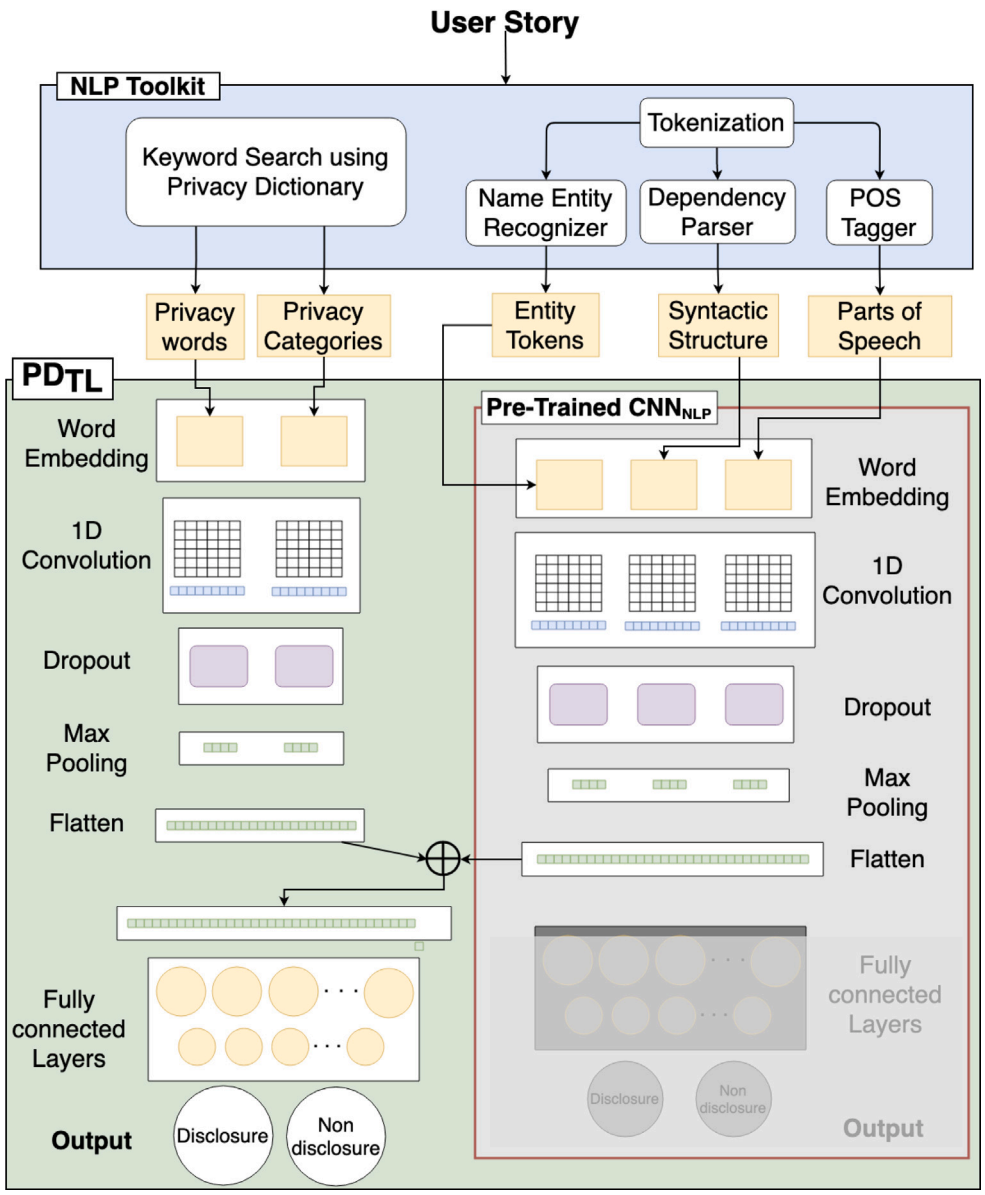


Fig. 4. The PD_{TL} architecture.

4.1. Research questions

The aim of our investigation is to assess the application of advanced methods and technologies to detect privacy content from USs. As mentioned in the introduction, we have first performed a sort of sanity check to verify:

(a) if a deep learning method (CNN_{NLP}) performs at least as shallow machine learning methods, when NLP-based features are exploited;
(b) if a deep learning method (CNN_{PW}) performs at least as shallow machine learning methods, when privacy word (PW) features are exploited.

Then, starting from the consideration that user story datasets are difficult to obtain, especially those containing sensible information, we have investigated the use of Transfer Learning (TL) which allows developers to analyze the similarities between different tasks and to exploit a neural network used for one task in a given domain and apply it to another domain.

To conduct this research study, we have formulated three research questions:

RQ1 Is CNN_{NLP} accurate at least as conventional machine learning methods to detect privacy content when using NLP-based features?
RQ2 Is CNN_{PW} accurate at least as conventional machine learning methods to detect privacy content when using PW features?
RQ3 Are predictions obtained with PD_{TL} better than those achieved with CNN_{NLP} and CNN_{PW}?

To answer RQ1 we have considered a convolutional network that is trained on different features extracted through different NLP techniques to predict if USs contain privacy information. The considered neural networks are powerful and flexible models that have the ability to detect complex patterns even with limited training data. These models have showed high performance in several domains, including natural language processing [47]. It is therefore reasonable to assume that they are also effective in this context.

As for conventional machine learning methods, we have considered: Logistic Regression (LR), Support Vector Machine (SVM), Gaussian

Naive Bayes (GNB), k-Nearest Neighbors (kNN), Random Forest (RF), and Decision Tree (DT). In the following, we name the models using NLP-based features as: LR_{NLP}, SVM_{NLP}, GNB_{NLP}, kNN_{NLP}, RF_{NLP}, and DT_{NLP}. The choice of using approaches like LR and SVM is not accidental: they are often used in the literature for solving relevant problems in software engineering. Moreover, they are particularly suitable when dealing with binary classification tasks.

Similarly, for addressing RQ2 we built and compare models obtained with CNN, LR, SVM, GNB, kNN, RF, and DT using PWs as features. In the following, we name these models as CNN_{PW}, LR_{PW}, SVM_{PW}, GN_{BPW}, kNN_{PW}, RF_{PW}, and DT_{PW}.

To address RQ3, we have considered the CNN, named PD_{TL}, defined in Section 3.4. In particular, the expectation is that the model resulting from transfer learning can provide better predictions than CNN_{NLP} and CNN_{PW}, as CNN_{NLP} is trained on few data containing privacy information and does not exploit PW features, while CNN_{PW} analyzes USs on a smaller set of features than PD_{TL}.

4.2. Data collection

The proposed model for the detection of privacy disclosures in USs needs data on which it is trained in order to make predictions. In particular, the data it needs should consist of a set of USs, each enriched by a label indicating whether that US has privacy disclosures, and by as many features as possible that contribute to the assertion of privacy relations. Datasets of this type, or similar, have not been found either on the Web or in the literature. Therefore, there was a need to build such a dataset, starting from a set of USs from which to extrapolate the characteristics that the model needs to make reliable predictions. To this end, a search was carried out to identify a large set of USs: this led to the discovery of 22 publicly available datasets, each containing more than 50 USs [21]. The method used to obtain these datasets is described in detail in [48].

Table 3 reports details and statistics about the considered datasets. Each row provides a brief description of the project, the number of USs, the number of privacy terms contained in the USs, and statistics about the NLP features used by the proposed approaches. In particular, each US was processed through the different NLP techniques in order to extrapolate the useful features to the subsequently defined models. The last four columns of the table indicate the percentages of USs containing: both Privacy Words and Disclosures (*PW&Di*), only Privacy Words (*PW*), only Disclosures (*Di*), none of the above (*None*). Note that the first author of the paper was in charge of manually classifying the privacy information, while the other two cross-checked the data. Table 4 shows four USs together with the extracted NLP features.

This dataset was manually analyzed to verify if it was heterogeneous enough, i.e., if it included enough instances of each type of USs. The types of USs are the result of the assumption explained in the previous section. In particular, the types identified are: USs containing privacy words and disclosures, USs containing only privacy words, USs containing only disclosures, USs that do not contain neither privacy words nor disclosures. Fig. 5 shows the percentages of USs for each type for the considered dataset. Special attention was paid to types that contained only one of the two properties implying the presence of privacy content. If they did not have a large number of instances, there was a risk that the model would fail to differentiate correctly between the various types of USs, thus compromising the validity of the prediction.

The independent variables identified are the features extracted through NLP, thus entities, dependencies, parts of speech, privacy words, and privacy categories, while the dependent variables are Accuracy and F1-score. The choice of the latter variables is explained in the following section.

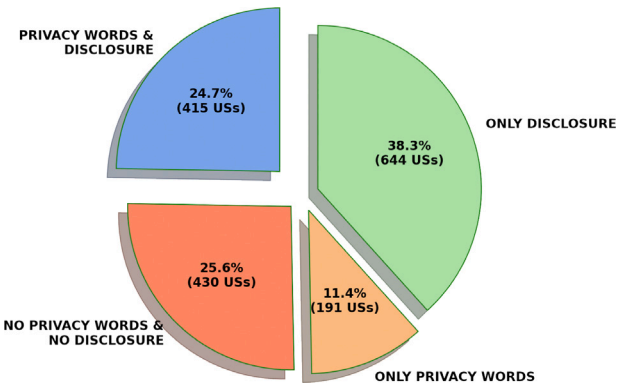


Fig. 5. Partitions of the dataset for each type of US.

4.3. Evaluation criteria

To evaluate the accuracy of the predictions, we used four popular evaluation metrics for classification task [49] – *Accuracy*, *Precision*, *Recall*, and *F1-score*. Accuracy is the most intuitive performance measure, and it is the ratio of the correctly predicted observations, i.e., *true positive* + *true negative*, to the total observations. Precision is calculated as *true positive* / (*true positive* + *false positive*) and indicates correctness of the responses provided by a technique. Recall measures the completeness of the responses and is calculated as *true positive* / (*true positive* + *false negative*). F1-score is defined as the harmonic mean of precision and recall and indicates balance between those.

These types of evaluation metrics were firstly considered because in a binary classification task accuracy, precision and recall are equally important. Furthermore, these metrics allowed a comparison between the models implemented in this work and the pre-trained model evaluated on the same metrics.

The objective is to try to observe how precise the models are while identifying aspects of privacy, and to try to understand what are the limitations of their ability to extract as much as possible such aspects from the test dataset.

Furthermore, it was verified that the predictions obtained using the different models came from the same population in order to assess whether the differences observed by applying the chosen evaluation criterion (i.e., Accuracy and F1-score) were legitimate or due to coincidence [50]. Note that, non-parametric techniques are usually preferred [51] to parametric methods when comparing machine learning and deep learning models mainly because they make fewer assumptions about the data. Thus, we decided to employ the McNemar test to compare the performance of two models [52,53]. In particular, given the predictions of two models, A and B, and the truth labels, a contingency table is calculated, which examines the number of instances of the following: (i) Both classifiers were correct; (ii) Both classifiers were incorrect; (iii) A was correct and B was incorrect; (iv) B was correct and A was incorrect. This makes it possible to estimate the probability that A is better than B at least as many times as observed in the experiment [52]. For comparing the performance of multiple machine learning and deep learning classifiers for the research questions in this thesis, the following null hypothesis was made:

Hn_0: All models are equally accurate in identifying aspects of privacy.

McNemar's test allowed to test the null hypothesis by comparing each pair of models under the same null hypothesis. As usual we considered a *p-value* of 0.05 as a "significance" threshold, i.e., p values lower than 0.05 are then assumed to be "significant", implying that the results obtained are hardly due to chance, allowing the null hypothesis to be rejected [52]. Thus, for the comparisons in which the null hypothesis was successfully rejected, it was determined whether one classifier was significantly better than the other classifiers.

Table 3
Properties of the datasets used for the research.

Dataset	Description	Size	Privacy terms	%PW&Di	%PW	%Di	%None
1	Online platform for delivering transparent information on US governmental spending	98	118	0.224	0.194	0.388	0.194
2	Electronic land management system for the Loudoun County, Virginia	58	107	0.328	0.000	0.638	0.340
3	An online platform to support waste recycling	51	86	0.176	0.137	0.137	0.549
4	Website for create a transparent overview of governmental expenses	53	85	0.566	0.151	0.170	0.113
5	Platform for obtaining insights from data	66	69	0.742	0.091	0.106	0.061
6	First version of the Scrum Alliance Website	97	115	0.175	0.031	0.670	0.124
7	New version of the NSF website: redesign and content discovery	73	115	0.041	0.000	0.740	0.219
8	App for camp administrators and parents	55	56	0.273	0.182	0.164	0.382
9	First version of the PlanningPoker.com website	53	53	0.170	0.057	0.623	0.151
10	Platform to find, share and publish data online	67	63	0.552	0.134	0.104	0.209
11	Management information system for Duke University	68	132	0.206	0.191	0.206	0.397
12	Simplified toolbox to enable fast and easy development with Hadoop	64	67	0.109	0.219	0.219	0.453
13	Research data management portal for the university of Oxford, Reading and Southampton	102	119	0.186	0.186	0.245	0.382
14	Personal interactive assistant for independent living and active aging	138	126	0.036	0.065	0.413	0.486
15	Conference registration and management platform	69	106	0.116	0.430	0.739	0.101
16	Software for machine-actionable data management plans	83	115	0.578	0.181	0.229	0.012
17	Web-based archiving information system	57	72	0.123	0.070	0.211	0.592
18	Institutional data repository for the University of Bath	53	89	0.660	0.038	0.226	0.075
19	Repository for different types of digital content	100	88	0.050	0.120	0.220	0.610
20	Software for archivists	100	117	0.250	0.130	0.430	0.190
21	Digital content management system for Cornell University	115	173	0.252	0.157	0.391	0.200
22	Citizen science platform that allows anyone to help in research tasks	60	82	0.050	0.067	0.400	0.483

Table 4
Overview of the dataset used for the empirical study.

#	User Story	Entities	Dependencies	Parts of Speech	Privacy Categories	Privacy Words	Disclosure?
0	As a Data user, I want to have the 12-19-2017 deletions processed.	['As', 'a', 'Data', 'user', 'PERSON', 'want', 'to', 'have', 'the', '12', '19', '2017', 'deletions', 'processed']	['prep', 'det', 'compound', 'pobj', 'nsubj', 'ROOT', 'aux', 'xcomp', 'det', 'nummod', 'nummod', 'nummod', 'dobj', 'acl']	['SCONJ', 'DET', 'PROPN', 'NOUN', 'PRON', 'VERB', 'PART', 'AUX', 'DET', 'NUM', 'NUM', 'NUM', 'NOUN', 'VERB']	[['PrivateSecret', 1]]	['data']	0
1	As a UI designer, I want to redesign the Resources page, so that it matches the new Broker design styles.	['As', 'a', 'HEALTH', 'HEALTH', 'PERSON', 'want', 'to', 'redesign', 'the', 'Resources', 'page', 'so', 'that', 'it', 'matches', 'the', 'new', 'PRODUCT', 'design', 'styles']	['prep', 'det', 'compound', 'pobj', 'nsubj', 'ROOT', 'aux', 'xcomp', 'det', 'compound', 'dobj', 'mark', 'mark', 'nsubj', 'advcl', 'det', 'amod', 'compound', 'compound', 'dobj']	['SCONJ', 'DET', 'PROPN', 'NOUN', 'PRON', 'VERB', 'PART', 'VERB', 'DET', 'PROPN', 'NOUN', 'SCONJ', 'SCONJ', 'PRON', 'VERB', 'DET', 'ADJ', 'PROPN', 'NOUN', 'NOUN']	none	none	0
2	As a UI designer, I want to report to the Agencies about user testing, so that they are aware of their contributions to making Broker a better UX.	['As', 'a', 'HEALTH', 'HEALTH', 'PERSON', 'want', 'to', 'report', 'to', 'the', 'ORG', 'about', 'user', 'testing', 'so', 'that', 'PERSON', 'are', 'aware', 'of', 'their', 'contributions', 'to', 'making', 'PRODUCT', 'a', 'better', 'UX']	['prep', 'det', 'compound', 'pobj', 'nsubj', 'ROOT', 'aux', 'xcomp', 'prep', 'det', 'pobj', 'prep', 'compound', 'pobj', 'mark', 'mark', 'nsubj', 'advcl', 'acomp', 'prep', 'poss', 'pobj', 'prep', 'pcomp', 'nsubj', 'det', 'amod', 'ccomp']	['SCONJ', 'DET', 'PROPN', 'NOUN', 'PRON', 'VERB', 'PART', 'VERB', 'ADP', 'DET', 'PROPN', 'ADP', 'NOUN', 'NOUN', 'SCONJ', 'SCONJ', 'PRON', 'AUX', 'ADJ', 'ADP', 'DET', 'NOUN', 'ADP', 'VERB', 'PROPN', 'DET', 'ADJ', 'PROPN']	[['OpenVisible', 1]]	['report']	1
3	As a UI designer, I want to move on to round 2 of DABS or FABS landing page edits, so that I can get approvals from leadership.	['As', 'a', 'HEALTH', 'HEALTH', 'PERSON', 'want', 'to', 'move', 'on', 'to', 'round', 'CARDINAL', 'of', 'DABS', 'or', 'FABS', 'landing', 'page', 'edits', 'so', 'that', 'PERSON', 'can', 'get', 'approvals', 'from', 'leadership']	['prep', 'det', 'compound', 'pobj', 'nsubj', 'ROOT', 'aux', 'xcomp', 'prt', 'aux', 'advcl', 'nummod', 'prep', 'pobj', 'cc', 'compound', 'compound', 'nsubj', 'conj', 'mark', 'mark', 'nsubj', 'aux', 'advcl', 'dobj', 'prep', 'pobj']	['SCONJ', 'DET', 'PROPN', 'NOUN', 'PRON', 'VERB', 'PART', 'VERB', 'ADV', 'PART', 'VERB', 'NUM', 'ADP', 'NOUN', 'CCONJ', 'NOUN', 'NOUN', 'NOUN', 'NOUN', 'SCONJ', 'SCONJ', 'PRON', 'VERB', 'AUX', 'NOUN', 'ADP', 'NOUN']	none	none	1

4.4. Validation method

In order to define the degree of accuracy or effectiveness of a machine learning model, one or more evaluations are carried out on the errors that are obtained in the predictions. In that case, after training, an error estimate is made for the model, called *residual evaluation*. However, this estimate only gives an idea of how well the model does on the data used to train it, as it is possible for the model to

be inadequate or in excess of the data. Thus, the problem with this evaluation technique is that it does not give an indication of how well the learning model will generalize to an independent or invisible dataset, i.e., on data it has not already seen. To this end, we have applied a k-fold cross-validation, dividing the original dataset into training and validation sets k times, considering $k = 5$. Furthermore, we run this 5-cross-validation 40 times. First, the cardinality of the sets defined by the assumption that determines whether the US is related to privacy content was analyzed. These cardinalities are shown in Fig. 5.

Using cross-validation, the sets defined for training consisted of 664 instances, where 50% are USs containing both privacy words and disclosures, the remaining 50% being divided between the other three types as described below. The test set was defined as 166 instances, where 83 consisted of USs containing both disclosures and privacy words.

4.5. Threats to validity

This part discusses the main threats to validity, explaining their possible effect and how they have been mitigated. Threats to the validity of this work derive mainly from the correctness of the tools used, the assumption regarding privacy content, and the generalizability and repeatability of the presented results.

Construct Validity It is about making sure that the measurement method corresponds to the construct being measured and is about the adequacy of the observations and inferences made based on the measurements performed during the study. In the context of using deep learning techniques for privacy content detection, methods offered by the Scikit-learn library have been used. In particular, the `f1_score`[5] method for measuring F1-score, and the `accuracy_score`[6] method for measuring Accuracy were used from Scikit-learn. Relying on results from a single tool can pose a threat to validity especially in the case of deep-learning. However, here it was decided to choose Accuracy and F1-score as they were used in previous studies investigating an approach to detect privacy disclosures and this allowed the results obtained here to be compared with those obtained in those studies. In addition, since the aim is to compare these deep learning techniques with classical machine learning models, the choice of these methods was almost obligatory, as the evaluation methods offered by Keras are not compatible with the Scikit-Learn models. On the other hand, the metrics offered by the latter library are based on results and predictions and, therefore, are also suitable for neural networks built using Keras.

Internal Validity It refers to the validity of the research results. It is mainly concerned with validating the control of extraneous variables and external influences that may impact the result. In the context of this work, exploring the applicability of transfer-learning for the detection of privacy aspects, it was assumed that the models used are compatible with each other, as they are both produced using the same technology, i.e., Keras. It would be interesting to observe how two models or neural networks developed through different APIs (e.g., Keras and Pytorch[7]) can be combined in a transfer-learning experiment. Another threat to internal validity could be causality. However, it is assumed that the necessary conditions for causality are quite fulfilled as statistically significant correlations have been found between measures obtained via different methods, reinforcing the idea that these correlations derive from fairly robust causal relationships.

External Validity It relates to the generalizability and repeatability of the produced results. The approach proposed in this work is based on Python. However, the statistical models used are replicable in other programming languages, so it is assumed that this method is programming language agnostic and therefore can be repeated for any other programming language given the availability of suitable frameworks. To promote the replication and construction of this work, as said above, we made available all tools, scripts and data.

Conclusion Validity It is a measure of how reasonable a research or experimental conclusion is. Although the number of observations made on statistical tests is not large, all the required hypotheses have been proved, therefore, the relationship between the data and the result is considered reasonable.

5. Results and discussion

In this section we present and discuss the results of the empirical study for each research question addressed.

5.1. RQ1: Is CNN_{NLP} accurate at least as conventional machine learning methods to detect privacy content when using NLP-based features?

We present the results achieved with the models built to verify if a deep learning method (CNN) exploiting NLP-based features performs at least as conventional (shallow) machine learning methods (our first sanity check).

As described in the previous Section 4.2, the built models are trained and tested with the same number of positive and negative samples. In particular, the positive samples are those with both Disclosures and Privacy Words, so for each fold 332 positive and 332 negative samples are selected for the training phase, while for the testing phase 83 positive and 83 negative samples are selected.

Each of the folds identified for the training was used for building CNN_{NLP} (i.e., the model obtained with NLP-based CNN), LR_{NLP}, SVM_{NLP}, GNB_{NLP}, kNN_{NLP}, RF_{NLP}, and DT_{NLP} (i.e., the models obtained with conventional machine learning methods by exploiting Scikit-Learn).

The aggregated results achieved in terms of the employed evaluation criteria are reported in Table 5, while Figs. 6 and 7 show the results of all runs graphically. By analyzing the accuracy and F1-score values reported in Table 5, we can observe that the results of CNN_{NLP} are better than those obtained by the others. Indeed, accuracy and F1-score values are 0.720 and 0.713 for CNN_{NLP}, respectively, while the other machine learning methods have obtained values less than 0.7. The worst results have been obtained with SVM_{NLP}.

From Fig. 6 we can observe that CNN_{NLP} is characterized by better Accuracy values for all runs, except for four cases. It is also interesting to note that for all methods we have a regular trend about the Accuracy values for all the runs, with a variation of up to 10%. For a few cases we have variations around 20% in the case of CNN_{NLP}. This is probably due to the syntactic structure of the USs selected for training phase. In particular, a greater number of positive and negative samples with a similar syntactic structure hampers the model to learn the presence of privacy aspects.

From Fig. 7 we can observe that for LR_{NLP}, KNN_{NLP}, DT_{NLP}, and RFC_{NLP} we have a regular trend about the F1-score values for all the runs (with a variation of up to 10%). Differently, for CNN_{NLP}, GNB_{NLP}, and SVM_{NLP} we can note some runs characterized by a variation around 20%. Only in seven runs CNN_{NLP} is characterized by F1-score values below those of other approaches.

As designed we have also verified whether the differences in the performances are statistically significant. To this end, we have performed the McNemar test to test the null hypothesis: "there are no differences in the accuracy of the models being compared". In particular, we have compared the predictions achieved with CNN_{NLP} with those achieved with each shallow machine learning based model (i.e., those obtained with LR, SVM, GNB, kNN, RFC, and DT). For all the performed comparisons, we have obtained a *p-value* <0.001, allowing the rejection of the null hypothesis, i.e., there is significant differences

[5] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html#sklearn-metrics-f1-score

[6] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

[7] https://pytorch.org/

Table 5
Results achieved with each model to answer RQ1 in terms of accuracy and F1-score.

Model	Accuracy	F1-Score
CNN_{NLP}	**0.720**	**0.713**
LR_{NLP}	0.617	0.605
SVM_{NLP}	0.519	0.084
GNB_{NLP}	0.510	0.612
kNN_{NLP}	0.557	0.519
RFC_{NLP}	0.662	0.669
DT_{NLP}	0.609	0.611

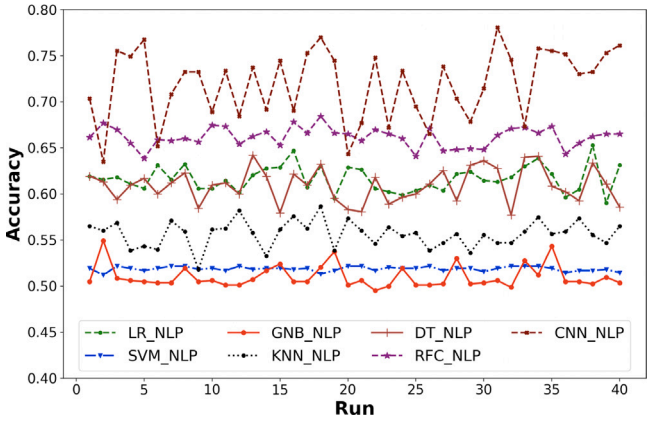


Fig. 6. Accuracy values of all the runs (to answer RQ1).

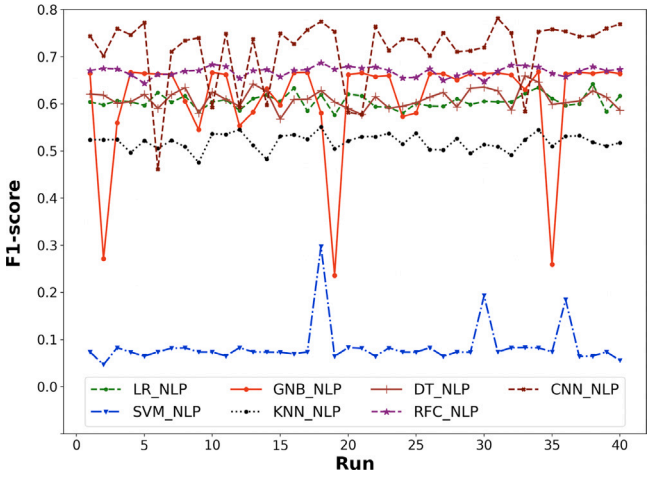


Fig. 7. F1-score values of all the runs (to answer RQ1).

between the predictions achieved with CNN_{NLP} and those achieved with the employed shallow machine learning based model. We can also conclude that the further effort needed to apply CNN is payed back by a significant improvement in the prediction accuracy.

> Thus, we can positively answer our first research question because a deep learning method (CNN) has provided better predictions than conventional (shallow) machine learning methods.

5.2. RQ2 is CNN_{PW} accurate at least as conventional machine learning methods to detect privacy content when using PW features?

This section is devoted to the presentation of results achieved by models built to answer our second research question RQ2, i.e., if a deep learning method (CNN) performs at least as shallow machine learning methods, when PW features are exploited (our second sanity check).

Table 6
Results achieved with each model to answer RQ2, in terms of accuracy and F1-score.

Model	Accuracy	F1-Score
CNN_{PW}	0.805	0.823
LR_{PW}	0.801	0.819
SVM_{PW}	**0.828**	**0.848**
GNB_{PW}	0.584	0.343
kNN_{PW}	**0.810**	**0.825**
RF_{PW}	**0.829**	**0.851**
DT_{PW}	0.805	0.819

Similarly to RQ1 analysis, the built models are trained and tested with the same number of positive and negative samples (see above). Thus, each of the folds identified for the training was used for the models built with CNN and LR, SVM, GNB, kNN, RF, and DT, by exploiting Scikit-Learn and PW features.

The aggregated results achieved in terms of employed evaluation criteria are reported in Table 6, while Figs. 8 and 9 show the results of all runs graphically. By analyzing the Accuracy and F1-score values for the built models shown in Table 6, we can observe that the values are from 0.80 to 0.85, which can be considered good results, except for GNB_{PW} which is characterized by worse performance with respect to the other employed machine learning methods.

From Figs. 8 and 9 we can observe that for all methods we have a regular trend for all the runs (with a variation of up to 10%), except for CNN_{PW} where for three runs the F1-score values are less than those of the other runs of about 20%, and for just one run the Accuracy value is less than those of the other runs of about 14%. The best result in terms of accuracy and F1-score has been obtained by using RF_{PW}. SVM_{PW} and kNN_{PW} have also provided better predictions than CNN_{PW}. They are reported in bold in Table 6.

As designed we have also verified whether the differences in the performances are statistically significant, by performing the McNemar test. For all the performed comparisons (i.e., CNN_{PW} vs LR_{PW}, CNN_{PW} vs SVM_{PW}, CNN_{PW} vs GNB_{PW}, CNN_{PW} vs kNN_{PW}, CNN_{PW} vs RF_{PW}, CNN_{PW} vs DT_{PW}), we obtained a *p-value* <0.001, allowing the rejection of the null hypothesis, i.e., there is significant differences between the predictions achieved using the two considered models. Thus, CNN_{PW} performs better than three conventional machine learning methods (i.e., LR_{PW}, DT_{PW}, and GNB_{PW}) and worse than the other three (i.e., SVM_{PW}, kNN_{PW}, and RF_{PW}), when PW features are exploited.

> Thus, we cannot positively answer our second research question, i.e., the deep learning methods is not accurate at least as all the considered conventional machine learning methods to detect privacy content when using PW features.

We can conclude that the second sanity check has been particularly useful because it highlights something unexpected, i.e., a deep learning method is not accurate at least as a conventional machine learning method. But it can happen as shown in previous similar works (e.g., [15]).

Just for completeness, we want to observe that the prediction models built with the shallow machine learning methods exploiting PW features are better than those obtained with the same shallow machine learning methods but exploiting NLP-based features (see Tables 5 and 6). The results of McNemar test have also revealed that these differences are statistically significant. Thus, the shallow machine learning methods improved their performances when trained with a not so large set of features.

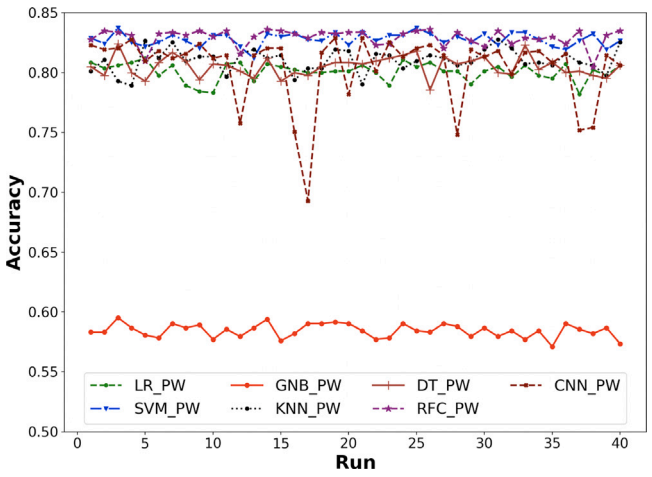


Fig. 8. Accuracy values of all the runs (to answer RQ2).

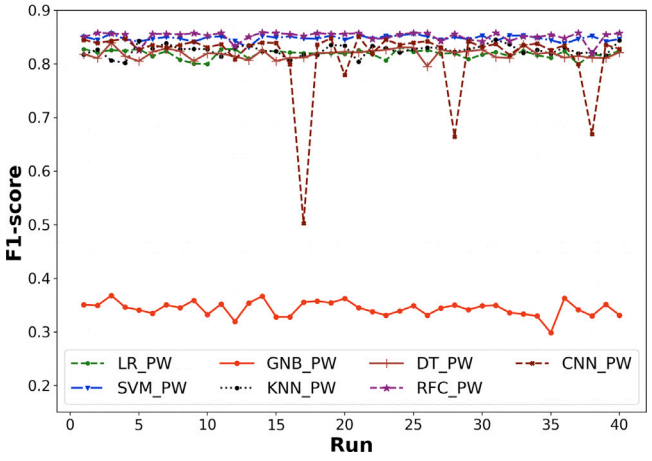


Fig. 9. F1-score values of all the runs (to answer RQ2).

5.3. RQ3: Are predictions obtained with PD_{TL} better than those achieved with CNN_{NLP} and CNN_{PW}?

The main goal of our investigation is the attempt to apply the technique of Transfer Learning, which consists in using the knowledge of a model in solving a specific task and combining it with another model for solving a different task, expanding the set of features used for prediction.

Similarly to RQ1 and RQ2 analyses, the comparisons between PD_{TL}, CNN_{NLP}, and CNN_{PW} have been performed in terms of Accuracy and F1-score, whereas the McNemar's statistical test has been used to verify the significance of the achieved results.

The aggregated results achieved in terms of employed evaluation criteria are reported in Table 7, while Figs. 10 and 11 show the results of all runs graphically. We can note that the model resulting from the application of the Transfer Learning (PD_{TL}) has provided better F1-score and Accuracy values (i.e., values greater than 0.90) than those achieved with the models based on deep learning analyzed previously (i.e., CNN_{NLP} and CNN_{PW}). In particular, PD_{TL} surpasses CNN_{PW} and CNN_{NLP} of more than 10% both in terms of Accuracy and F1-score. Furthermore, the results of the McNemar test have revealed that the differences are statistically significant (p-value <0.001 for both the comparisons). Furthermore, as clearly shown in Figs. 10 and 11 PD_{TL} has provided better results for all the runs except one, and the distribution of the values is characterized by less variation with respect to the ones of CNN_{NLP} and CNN_{PW}.

Table 7
Results achieved with each model to answer RQ3, in terms of accuracy and F1-score.

Model	Accuracy	F1-Score
CNN_{NLP}	0.720	0.713
CNN_{PW}	0.805	0.823
PD_{TL}	**0.937**	**0.937**

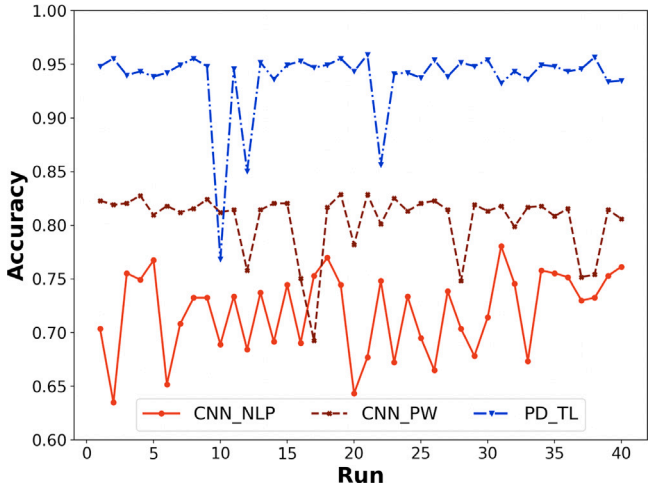


Fig. 10. Accuracy values of all the runs (to answer RQ3).

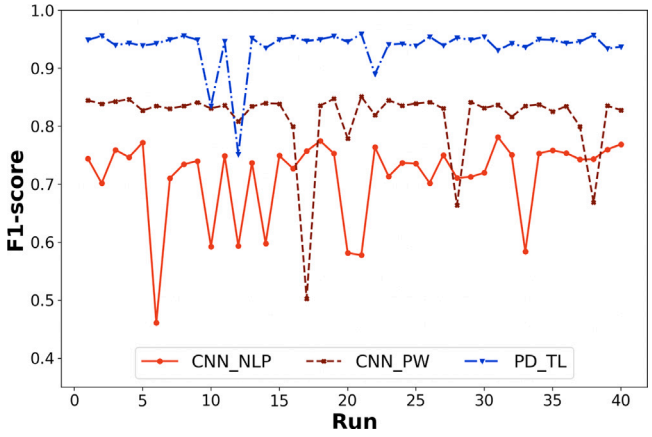


Fig. 11. F1-score values of all the runs (to answer RQ3).

Based on the obtained results, it is therefore possible to state not only that Transfer Learning is feasible but that it is better than using deep learning models alone for privacy content analysis.

> Thus, we can positively answer our third research question, i.e., predictions obtained with PD_{TL} are better than those achieved with CNN_{NLP} and CNN_{PW}.

5.4. Findings and suggestions for researchers and practitioners

The analysis carried out to answer our research questions allows us to highlight implications for researchers and practitioners about the applicability of our findings. We organize the discussion according to the achieved contributions.

On the use of a tool to predict privacy content. We have provided an approach and tool to automatically predict privacy content from

user stories (problem never addressed before), which exploit a combination of NLP and transfer learning strategies. This should encourage software engineering researchers and in particular practitioners in considering the opportunities of automating privacy content detection.

☞ *Implication 1. Practitioners have the possibility to exploit an approach and tool that allow to reduce the effort (and cost) to identify privacy requirements in the early phase of design. User studies involving practitioners should be performed with the aim of promoting the suggested approach and tool.*

On the use of deep learning methods. As expected the experimental results show that the use of NLP-based CNNs can contribute to improve predictions about privacy requirements with respect to the use of conventional (shallow) machine learning methods. However, the analysis has also revealed that the strategy for training the models are crucial. In particular, RQ2 analysis has not highlighted a clear advantage in using deep learning methods with respect to conventional (shallow) machine learning methods. Other studies achieved a similar result (e.g., [15]).

☞ *Implication 2. Researchers should invest some effort in conducting empirical studies considering different datasets aiming at identifying strategies for training NLP-based prediction models in the context of agile for privacy requirement detection.*

On the use of privacy words. Our analysis has clearly shown that the use of privacy words allowed us to significantly improve the predictions of some employed shallow machine learning methods (if we compare RQ2 results against RQ1 results). In particular, RF_{PW}, SVM_{PW}, and kNN_{PW} have also provided better predictions than CNN_{NLP}. Thus, even cheaper methods can provide good predictions when exploiting data of the specific domain under investigation.

☞ *Implication 3. The research community should invest some effort in investigating the impact of the specific domain data on the use of cheaper methods aiming at verifying their effectiveness with respect to more expensive methods.*

On the use of Transfer Learning. The main result of our analysis is about the use of Transfer Learning that has allowed us to improve the performance of the built NLP-based CNN prediction models of about 10% in terms of ts. This is a further confirmation of the benefit of using this emergent strategy, which allows to reuse a system developed for a task to build a model for a different but related task [18–20].

☞ *Implication 4. Researchers should apply Transfer Learning for training NLP-based prediction models aiming at improving their effectiveness in detecting privacy as well as security requirements in the agile context.*

6. Conclusions and future work

Interest in machine learning techniques based on natural language processing has been growing in recent years, including in the field of software engineering. Most of the existing attempts are focused on the generation of models and components useful in the different phases of software engineering from customer-specified requirements. On the other hand, few attempts to capture non-functional requirements have been documented in the literature, yet they contribute quite a bit in the evaluation of software quality.

The results of our empirical study have revealed that deep learning methods can be used for the detection of non-functional requirements from customer requirements. In particular, it was found that deep learning models can be used for the identification of privacy disclosures in user stories, even with near-optimal performance. Furthermore, the search for recent deep learning techniques has led to the exploration of Transfer Learning, and therefore the possibility of its application in this context has been evaluated. The experiment on the application of Transfer Learning has demonstrated the feasibility of practicing this technique in the context of privacy content detection in user stories.

As for future research directions, there are reasons to extend this work to a broader scope, including other NFRs, or experimenting with such techniques for other similar tasks. Of course, future developments could also focus on improving the strategies employed in the work. For instance, an update of the privacy dictionary used or the production of newer, more elaborate taxonomies could help in this regard. Further research might involve the adoption of other NLP techniques for feature extraction, expanding the set on which the various models are trained and tested. Eventually, it would be interesting to analyze the application of Transfer Learning between models of different nature, both technological and methodological, in order to better understand in which contexts and circumstances this technique leads to significant improvements.

CRediT authorship contribution statement

Francesco Casillo: Methodology, Software, Validation, Writing – original draft. **Vincenzo Deufemia:** Conceptualization, Formal analysis, Writing – review & editing, Supervision. **Carmine Gravino:** Conceptualization, Formal analysis, Writing – review & editing, Supervision.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.infsof.2022.106853.

References

[1] I. Sommerville, P. Sawyer, Requirements Engineering: A Good Practice Guide, Wiley, New York, NY, USA, 1997.
[2] K. Pohl, Requirements Engineering: Fundamentals, Principles, And Techniques, first ed., Springer Publishing Company, Incorporated, 2010.
[3] D.M. Fernández, S. Wagner, M. Kalinowski, M. Felderer, P. Mafra, A. Vetrò, T. Conte, M.-T. Christiansson, D. Greer, C. Lassenius, T. Männistö, M. Nayabi, M. Oivo, B. Penzenstadler, D. Pfahl, R. Prikladnicki, G. Ruhe, A. Schekelmann, S. Sen, R. Spinola, A. Tuzcu, J.L. de la Vara, R. Wieringa, Naming the pain in requirements engineering - Contemporary problems, causes, and effects in practice, Empir. Softw. Eng. (2017) 2298–2338.
[4] F. Paetsch, A. Eberlein, F. Maurer, Requirements engineering and agile software development, in: Proceedings Of 12th IEEE International Workshops On Enabling Technologies (WETICE 2003), Infrastructure For Collaborative Enterprises, 9–11 June 2003, Linz, Austria, IEEE Computer Society, 2003, pp. 308–313, http://dx.doi.org/10.1109/ENABL.2003.1231428.
[5] Z. Kurtanović, W. Maalej, Automatically classifying functional and non-functional requirements using supervised machine learning, in: Proceedings Of IEEE 25th International Requirements Engineering Conference, RE, 2017, pp. 490–495, http://dx.doi.org/10.1109/RE.2017.82.
[6] Q.L. Nguyen, Non-functional requirements analysis modeling for software product lines, in: Proceedings Of ICSE Workshop On Modeling In Software Engineering, MiSE 2009, Vancouver, BC, Canada, May 17–18, 2009, IEEE Computer Society, 2009, pp. 56–61, http://dx.doi.org/10.1109/MISE.2009.5069898.
[7] J. Slankas, L. Williams, Automated extraction of non-functional requirements in available documentation, in: Proceedings Of 1st International Workshop On Natural Language Analysis In Software Engineering, NaturaLiSE, 2013, pp. 9–16, http://dx.doi.org/10.1109/NAturaLiSE.2013.6611715.
[8] P. Anthonysamy, A. Rashid, R. Chitchyan, Privacy requirements: Present future, in: Proceedings Of IEEE/ACM 39th International Conference On Software Engineering: Software Engineering In Society Track, ICSE-SEIS, 2017, pp. 13–22, http://dx.doi.org/10.1109/ICSE-SEIS.2017.3.
[9] L. Cao, B. Ramesh, Agile requirements engineering practices: An empirical study, IEEE Softw. (2008) 60–67.
[10] F. Paetsch, A. Eberlein, F. Maurer, Requirements engineering and agile software development, in: Proceedings Of IEEE International Workshops On Enabling Technologies: Infrastructure For Collaborative Enterprises, 2003, pp. 308–313, http://dx.doi.org/10.1109/ENABL.2003.1231428.
[11] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[12] N. Mehdy, C. Kennington, H. Mehrpouyan, Privacy disclosures detection in natural-language text through linguistically-motivated artificial neural networks, in: Security And Privacy In New Computing Environments, 2019, pp. 152–177, http://dx.doi.org/10.1007/978-3-030-21373-2_14.

[13] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P.S. Yu, L. He, A survey on text classification: From shallow to deep learning, 2020, CoRR abs/2008.00364, arXiv:2008.00364.

[14] J. Haneczok, J. Piskorski, Shallow and deep learning for event relatedness classification, Inf. Process. Manage. 57 (6) (2020) 102371.

[15] M. Oleynik, A. Kugic, Z. Kasáč, M. Kreuzthaler, Evaluating shallow and deep learning strategies for the 2018 n2c2 shared task on clinical text classification, J. Am. Med. Inf. Assoc. 26 (11) (2019) 1247–1254.

[16] G. Xu, C. Qi, H. Yu, S. Xu, C. Zhao, J. Yuan, Detecting sensitive information of unstructured text using convolutional neural network, in: Proceedings Of International Conference On Cyber-Enabled Distributed Computing And Knowledge Discovery, CyberC, 2019, pp. 474–479, http://dx.doi.org/10.1109/CyberC.2019.00087.

[17] J. Neerbeky, I. Assentz, P. Dolog, Taboo: Detecting unstructured sensitive information using recursive neural networks, in: Proceedings Of IEEE 33rd International Conference On Data Engineering, ICDE, 2017, pp. 1399–1400, http://dx.doi.org/10.1109/ICDE.2017.195.

[18] L. Torrey, J. Shavlik, Transfer learning, in: Handbook Of Research On Machine Learning Applications And Trends: Algorithms, Methods, And Techniques, IGI global, 2010, pp. 242–264.

[19] E. Kocaguneli, T. Menzies, E. Mendes, Transfer learning in effort estimation, Empir. Softw. Eng. 20 (3) (2015) 813–843.

[20] R. Krishna, T. Menzies, Bellwethers: A baseline method for transfer learning, IEEE Trans. Software Eng. 45 (11) (2019) 1081–1105.

[21] F. Dalpiaz, Requirements data sets (user stories), 2018, http://dx.doi.org/10.17632/7zbk8zsd8y.1, https://data.mendeley.com/datasets/7zbk8zsd8y/.

[22] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley, 1999.

[23] G. Lucassen, F. Dalpiaz, J.M. van der Werf, S. Brinkkemper, The use and effectiveness of user stories in practice, in: M. Daneva, O. Pastor (Eds.), Requirements Engineering: Foundation For Software Quality, Springer International Publishing, Cham, 2016, pp. 205–222.

[24] M. Cohn, User Stories Applied: For Agile Software Development, Addison Wesley, 2004.

[25] S. Jiménez, R. Juárez-Ramírez, A quality framework for evaluating grammatical structure of user stories to improve external quality, in: Proceedings Of 7th International Conference In Software Engineering Research And Innovation, CONISOFT, 2019, pp. 147–153, http://dx.doi.org/10.1109/CONISOFT.2019.00029.

[26] G. Lucassen, F. Dalpiaz, J.M. van der Werf, S. Brinkkemper, Forging high-quality user stories: Towards a discipline for agile requirements, in: Proceedings Of IEEE 23rd International Requirements Engineering Conference, RE, 2015, pp. 126–135, http://dx.doi.org/10.1109/RE.2015.7320415.

[27] P. Heck, A. Zaidman, A quality framework for agile requirements: A practitioner's perspective, 2014, arXiv:1406.4692.

[28] W.B.A. Karaa, Z.B. Azzouz, A. Singh, N. Dey, A.S. Ashour, H.B. Ghézala, Automatic builder of class diagram (ABCD): an application of UML generation from functional requirements, Softw. Pract. Exp. 46 (11) (2016) 1443–1458.

[29] M. Elallaoui, K. Nafil, R. Touahni, Automatic transformation of user stories into UML use case diagrams using NLP techniques, Procedia Comput. Sci. 130 (2018) 42–49, The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops.

[30] S. Nasiri, Y. Rhazali, M. Lahmer, N. Chenfour, Towards a generation of class diagram from user stories in agile methods, Procedia Comput. Sci. 170 (2020) 831–837, The 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.

[31] G. Lucassen, M. Robeer, F. Dalpiaz, J.M.E.M. van der Werf, S. Brinkkemper, Extracting conceptual models from user stories with visual narrator, Requir. Eng. 22 (3) (2017) 339–358.

[32] M. Robeer, G. Lucassen, J.M. van der Werf, F. Dalpiaz, S. Brinkkemper, Automated extraction of conceptual models from user stories via NLP, in: Proceedings Of IEEE 24th International Requirements Engineering Conference, RE, 2016, pp. 196–205, http://dx.doi.org/10.1109/RE.2016.40.

[33] F. Gilson, C. Irwin, From user stories to use case scenarios towards a generative approach, in: Proceedings Of 25th Australasian Software Engineering Conference, ASWEC, 2018, pp. 61–65, http://dx.doi.org/10.1109/ASWEC.2018.00016.

[34] L. Müter, T. Deoskar, M. Mathijssen, S. Brinkkemper, F. Dalpiaz, Refinement of user stories into backlog items: Linguistic structure and action verbs, in: E. Knauss, M. Goedicke (Eds.), Requirements Engineering: Foundation For Software Quality, Springer International Publishing, Cham, 2019, pp. 109–116.

[35] P. Rane, Automatic Generation of Test Cases for Agile using Natural Language Processing (Ph.D. thesis), Virginia Tech, 2017.

[36] F. Gilson, M. Galster, F. Georis, Extracting quality attributes from user stories for early architecture decision making, in: Proceedings Of IEEE International Conference On Software Architecture Companion, ICSA-C, 2019, pp. 129–136, http://dx.doi.org/10.1109/ICSA-C.2019.00031.

[37] H. Villamizar, A. Anderlin Neto, M. Kalinowski, A. Garcia, D. Méndez, An approach for reviewing security-related aspects in agile requirements specifications of web applications, in: Proceedings Of IEEE 27th International Requirements Engineering Conference, RE, 2019, pp. 86–97, http://dx.doi.org/10.1109/RE.2019.00020.

[38] M. Riaz, J. King, J. Slankas, L. Williams, Hidden in plain sight: Automatically identifying security requirements from natural language artifacts, in: Proceedings Of IEEE 22nd International Requirements Engineering Conference, RE, 2014, pp. 183–192, http://dx.doi.org/10.1109/RE.2014.6912260.

[39] K. Barker, M. Askari, M. Banerjee, K. Ghazinour, B. Mackas, M. Majedi, S. Pun, A. Williams, A data privacy taxonomy, in: Proceedings Of The 26th British National Conference On Databases: Dataspace: The Final Frontier, BNCOD 26, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 42–54, http://dx.doi.org/10.1007/978-3-642-02843-4_7.

[40] S. De Capitani Di Vimercati, S. Foresti, G. Livraga, P. Samarati, Data privacy: Definitions and techniques, Internat. J. Uncertain. Fuzziness Knowledge-Based Systems 20 (06) (2012) 793–817.

[41] A.J. Gill, A. Vasalou, C. Papoutsi, A.N. Joinson, Privacy dictionary: A linguistic taxonomy of privacy for content analysis, in: Proceedings Of The SIGCHI Conference On Human Factors In Computing Systems, ACM, New York, NY, USA, 2011, pp. 3227–3236, http://dx.doi.org/10.1145/1978942.1979421.

[42] A. Vasalou, A. Gill, F. Mazanderani, C. Papoutsi, A. Joinson, Privacy dictionary: A new resource for the automated content analysis of privacy, J. Am. Soc. Inf. Sci. Technol. 62 (11) (2011) 2095–2105, http://dx.doi.org/10.1002/asi.21610.

[43] P. Silva, C. Gonçalves, C. Godinho, N. Antunes, M. Curado, Using NLP and machine learning to detect data privacy violations, in: Proceedings Of IEEE Conference On Computer Communications Workshops, 2020, pp. 972–977, http://dx.doi.org/10.1109/INFOCOMWKSHPS50562.2020.9162683.

[44] W.B. Tesfay, J. Serna, K. Rannenberg, PrivacyBot: Detecting privacy sensitive information in unstructured texts, in: Proceedings Of Sixth International Conference On Social Networks Analysis, Management And Security, SNAMS, 2019, pp. 53–60, http://dx.doi.org/10.1109/SNAMS.2019.8931855.

[45] S. Sheth, G. Kaiser, W. Maalej, Us and them: A study of privacy requirements across North America, Asia, and Europe, in: Proceedings Of The 36th International Conference On Software Engineering, ICSE 2014, Association for Computing Machinery, New York, NY, USA, 2014, pp. 859–870, http://dx.doi.org/10.1145/2568225.2568244.

[46] D.A. Evans, C. Zhai, Noun phrase analysis in unrestricted text for information retrieval, in: Proceedings Of 34th Annual Meeting Of The Association For Computational Linguistics, ACL, Santa Cruz, California, USA, 1996, pp. 17–24, http://dx.doi.org/10.3115/981863.981866, URL: https://aclanthology.org/P96-1003.

[47] T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation, 2015, ArXiv abs/1508.04025, arXiv:1508.04025.

[48] F. Dalpiaz, I. Van Der Schalk, S. Brinkkemper, F.B. Aydemir, G. Lucassen, Detecting terminological ambiguity in user stories: Tool and experimentation, Inf. Softw. Technol. 110 (2019) 3–16.

[49] D.M.W. Powers, Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation, J. Mach. Learn. Technol. 2 (2011) 37–63.

[50] R.L. Wasserstein, N.A. Lazar, The ASA statement on p-values: Context, process, and purpose, Amer. Statist. 70 (2) (2016) 129–133.

[51] A. Fernández, S. García, J. Luengo, E. Bernadó, F. Herrera, Genetics-based machine learning for rule induction: State of the art, taxonomy, and comparative study, IEEE Trans. Evol. Comput. 14 (2011) 913–941.

[52] S. Salzberg, On comparing classifiers: Pitfalls to avoid and a recommended approach, Data Min. Knowl. Discov. 1 (3) (1997) 317–328.

[53] N. Japkowicz, M. Shah, Evaluating Learning Algorithms: A Classification Perspective, Cambridge University Press, 2011, http://dx.doi.org/10.1017/CBO9780511921803.