Identifying security-related requirements in regulatory documents based on cross-project classification

Citation for the original published paper (version of record):
Mohamad, M., Steghöfer, J., Åström, A. et al (2022). Identifying security-related requirements in regulatory documents based on cross-project
classification. PROMISE 2022 - Proceedings of the 18th International Conference on Predictive Models and Data Analytics in Software Engineering, co-located with ESEC/FSE 2022: 82-91.
http://dx.doi.org/10.1145/3558489.3559074

(article starts on next page)

Identifying Security-Related Requirements in Regulatory Documents Based on Cross-Project Classification

Mazen Mohamad
Chalmers | University of Gothenburg
Gothenburg, Sweden
mazen.mohamad@gu.se

Jan-Philipp Steghöfer
Xitaso GmbH
Augsburg, Germany
jan-philipp.steghoefer@xitaso.com

Alexander Åström
Comentor AB
Gothenburg, Sweden
alexander.astrom@comentor.se

Riccardo Scandariato
Hamburg University of Technology
Germany, Hamburg
riccardo.scandariato@tuhh.de

ABSTRACT

Security is getting substantial focus in many industries, especially safety-critical ones. When new regulations and standards which can run to hundreds of pages are introduced, it is necessary to identify the requirements in those documents which have an impact on security. Additionally, it is necessary to revisit the requirements of existing systems and identify the security related ones. We investigate the feasibility of using a classifier for security-related requirements trained on requirement specifications available online. We base our investigation on 15 requirement documents, randomly selected and partially pre-labelled, with a total of 3,880 requirements. To validate the model, we run a cross-project prediction on the data where each specification constitutes a group. We also test the model on three different United Nations (UN) regulations from the automotive domain with different magnitudes of security relevance. Our results indicate the feasibility of training a model from a heterogeneous data set including specifications from multiple domains and in different styles. Additionally, we show the ability of such a classifier to identify security requirements in real-life regulations and discuss scenarios in which such a classification becomes useful to practitioners.

CCS CONCEPTS

• **Security and privacy → Software and application security**.

KEYWORDS

Security Requirements, Requirements Classification, Machine Learning, Automated Requirements Engineering

ACM Reference Format:
Mazen Mohamad, Jan-Philipp Steghöfer, Alexander Åström, and Riccardo Scandariato. 2022. Identifying Security-Related Requirements in Regulatory Documents Based on Cross-Project Classification. In *Proceedings of the 18th International Conference on Predictive Models and Data Analytics in Software Engineering (PROMISE '22), November 17, 2022, Singapore, Singapore.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3558489.3559074

PROMISE '22, November 17, 2022, Singapore, Singapore

ACM ISBN 978-1-4503-9860-2/22/11.
https://doi.org/10.1145/3558489.3559074

1 INTRODUCTION

Cybersecurity is gaining substantial focus in safety-critical industries, e.g., automotive. An important reason for that is the potential implication of security breaches on the safety of the users. The driving forces for this focus are regulatory authorities and standardization bodies which are frequently releasing documents that contain new security requirements. Engineers involved in creating safety- and security-critical products therefore need to digest new regulatory and standards documents frequently and identify requirements that have an impact on security. Since these documents can run to hundreds of pages, this is an error-prone and time-consuming process. In addition, the regulatory bodies around the world does not necessary align their work, which could lead to contradicting requirements in different markets. This, in turn, may force SW companies to branch their code base, and hence the need to continuously and closely monitor and maintain them. This too is an important aspect for companies to consider as early as possible.

In this paper, we investigate the usefulness of a classifier to identify security-related requirements in such regulatory documents. The classifier we use is trained on existing, freely available requirements specifications, therefore depending on freely available and easily accessible documents.

While there are many scenarios where such a classification would be useful, we want to highlight two: the introduction of new legislation or standards and the use of Requests for Proposals (RFPs) in the requirements engineering process.

In the first scenario, *new legislation* such as Europe's GDPR [7] or *new standards* such as the recently released automotive industry standard for cybersecurity ISO/SAE 21434 [26] or the new UN regulation for cybersecurity in the automotive industry [27] make it necessary to revisit the requirements of existing systems and identify those that have an impact on security. Gaining approval requires both understanding the concrete security requirements in the standards as well as which of the vehicles' requirements are security-related. Identifying both kinds of requirements can be cumbersome, time consuming, and requires an enormous amount of effort. If an automated approach provides sufficient accuracy, companies will be able to save significant effort, time, and cost.

The second scenario concerns requirements engineering processes that are based on a *request for proposal* [22] in which a customer provides the requirements. If an organisation bidding during the tender phase can easily classify security-related requirements, it will be easier to identify the impact on the architectural choices and the project's complexity early on. This in turn allows making a bid based on better information. The cost of the project will be clearer from the get-go.

In both cases, the cost impacts depend on the quality of the classification, the manual effort required to find suitable projects for learning, and how closely the classification results need to be checked and revised. In particular, finding existing specifications or regulations which have been labelled to identify security requirements can be very difficult and using publicly available information is desirable. At the same time, previous work shows that classification of security requirements [12] or non-functional requirements in general [14] is difficult across projects. This can be based on specific terminology in the training data or on the type of requirements specification, e.g., software requirements specification or backlog, used for training. Our first research question is thus:

RQ1: Which performance can a classifier for security-related requirements achieve if it is trained on data from other projects?

While applying such a classifier to existing requirements is relevant for both scenarios sketched above, we are also interested in classifying the requirements in standards and regulation documents. This is a novelty and much of the related work uses synthetic data, e.g., compiled from student projects (e.g., [5]). We therefore ask:

RQ2: Which performance can a classifier trained on requirements specification documents achieve in identifying security-related requirements in a regulation?

We answer these research questions by performing a cross-project prediction with fifteen different requirements specifications mined from the internet, including two non-technical requirement documents. Each of the fifteen documents forms one group. We then test the classifier with three different UN regulations.

Our results show that performance of a classifier trained on 14 of the specifications and applied to the remaining one can be very good, but that caveats apply with regard to the data quality of the training data. We also find the classifier helpful for practitioners in identifying security requirements in a regulation and point out the sections in these regulations which include security requirements.

2 RELATED WORK

In this section, we review related work by looking at studies for classifying requirements in general, and focus on the studies which specifically tackle security requirements classification.

Automatic classification and categorization of requirements has been explored in multiple studies. Researchers have suggested approaches for classifying functional requirements in order to analyse customers' requirements [31] and to classify requirements documents into content topics in order to assist reviewers from certification authorities in finding inconsistencies [21]. Winkler and Vogelsang [30] studied the classification of contents of requirements specifications into classes of requirements or information.

Table 1 summarizes the related work focusing on classifying security requirements. For each of the reviewed papers, we specify what was classified, e.g., security/non-security or functional/non-functional requirements. We also look at the number of requirements used to train the classifier, the source of the data used in the study, and whether or not the study uses cross-project prediction.

Knauss et al. [12] approach the problem that security issues are ignored early in software projects by creating an early indication of security issues based on security requirements. The approach uses a classifier for security-related requirements. The model is evaluated by using three industrial requirement specifications. The paper also tests transferability of a model trained on two specifications to the third. The results showed an F_1measure below 50% in these cases, while a cross validation on the whole data set gave an average F_1-measure of 84%. We use the results from this study as our baseline, as we use a dataset that includes the projects from Knauss et al. and apply the same idea of classifying across projects.

Kutranovic et al. [14] also study the classification of requirements into functional and non-functional. The researchers study a multi-class classification of the non-functional requirements. One of these classes is security. The data set was from the RE17 conference challenge including 66 security requirements. The results of security requirements classification show an F_1measure of 88%. A similar study is done by Cleland-Huang et at. [5]. The data was collected from 15 projects with a total of 684 requirements, of which 326 are non-functional. The results show an average recall of around 70%. However, the score for security requirements was below that average.

Casamayro et al. [3] used a semi-supervised learning approach to identify non-functional requirements. The researchers used the TERA-PROMISE repository which consists of requirements taken from 15 projects. The repository consists of 625 requirements in total. The classifier achieved a precision and recall of around 80%.

Abad et al. [1] suggest a method for improving the automated classification of functional and non-functional requirements by pre-processing the requirements. The researchers used the same dataset used by Casamayro et al. [3]. The classification performance varied significantly among different classification algorithms. Four of the six algorithms achieved precision and recall values under 30%, whereas one achieved 97% and 100% recall and precision respectively.

Hey et al. [9] propose an approach which fine-tunes Bidirectional Encoder Representations from Transformers (BERT), which is a pre-trained deep-learning language model to classify requirements. The researchers use the PROMISE NFR dataset and apply the model on unseen projects, which we consider to be similar to the cross-project approach we use in this study. The performance of the model proved to achieve considerably better results than related approaches with an F_1 score of 83% for security. The used dataset, however, only included 66 security requirements.

Munaiah et al. [20] use the Common Weakness Enumeration (CWE) data set to train a classifier for security requirements based on only positive samples. The study test the approach on the three projects (same used by Knauss et al. [12]) and achieved an average precision, recall and F_1score of 67.35%, 70.48% and 67.68% respectively. The study concludes that using a one-class classifier trained on solely positive samples outperforms binary classifiers trained with samples from both the negative and positive classes.

Table 1: Studies with related work that deal with classification of security requirements

Study	Classification	Requirements	Data source	Cross-project
Abad et al. [1]	FR / NFR with subcategories for NFR	625	TERA-PROMISE	No
Casamayro et.al [3]	FR / NFR with subcategories for NFR	625	TERA-PROMISE	No
Knauss et. al [12]	security / non-security	510	3 projects	Yes
Kutranovic et al. [14]	FR / NFR with subcategories for NFR	625	RE17 conference data	No
Cleland-Huang et al. [5]	NFR multiclass	684	30 student projects	Yes
Hey et al. [9]	NFR multiclass	625	TERA-PROMISE	Yes*
Munaiah et al. [20]	security / non-security	NA**	Common Weakness Enumeration (CWE)	Yes*
Tong Li et al. [17]	security / non-security	625/510***	15 / 3 projects	Yes
This study	Security / non-security	3880	15 heterogenous projects	Yes

* Study tests the classifier on projects not included in the training set but do not apply a cross-project approach.
** The study does not use requirements but rather the Common Weakness Enumeration (CWE) data set.
*** Study uses two different data sets. The same used by Cleland-Huang et al. and Knauss et al.

Tong Li et al. [17] propose an ontology-based learning approach to build a classifier for security requirements. The study investigates the ability of the classifier to be non-domain-specific. The results are compared to those achieved by Knauss et al. [12] and show an increase in the F_1measure by 0.2.

In our study, we use a heterogeneous dataset with significantly more requirements than the related work (i.e., approximately as many security requirements as the total number of requirements used in the related studies), as our aim is to study the possibility of cross-project training and prediction. We also investigate the applicability of a *simple* classifier in detecting security requirements in real-life regulatory documents which we believe is not sufficiently investigated in related literature.

As we can see from Table 1, two studies include a cross-project validation of the requirements classifiers. The results of the cross-project validation reported by [12] shows that in all the examined cases, either the precision, the recall of both had value less than 50%, which indicates the inability to do cross-project classification. In our study, we got different results, which we believe is due to the number of requirements (more than five times more than in Knauss et al.), and the number of different projects (five times more than in Knauss et al.).

3 METHOD

In this section, we describe the method we used to conduct this study. We start by explaining how we identified the projects from which we collected data in 3.1, then we describe how we defined security requirements in 3.2, and the labelling process in 3.3. Later, we present our experimental setup for each of our research questions in 3.4 and 3.5.

3.1 Identification of Projects

We identified fifteen different requirement specifications from commercial projects, student projects, domain specific guidelines, industrial projects, and research projects with a total of 3.880 requirements to use in this study. Ten of the data points were collected by performing a Google search. Our goal was to use data that is as easily accessible and as heterogeneous as possible. Hence, we did not restrict the search to a specific domain or a specific type of requirement specifications but only file type, as we were only interested in file types from which we can easily extract the data. The search string we used was:

```
requirements specification filetype:xls OR xlsx OR pdf
```

We went through the results and excluded files from which extraction of data was not possible, e.g., due to access restrictions or non-English text. We then selected the first ten files that fit our criteria for inclusion in our data set.

Additionally we added three industrial requirements specification previously used by Knauss et. al [12], and two non-technical requirement documents.

Table 2 shows information about the requirement specifications from which we collected our data. As the table indicates, the documents are from different projects and contain different numbers of requirements. Some of the documents stem from industrial and commercial projects, domain specific guidelines, one has been created by researchers, and yet others have been created by students. There is also a difference in the type of specification: we included System Requirement Specifications (SRS), Requests for Proposals (RFP), non-technical requirement documents (NTR), and Backlog Items (BL) for certain products. The requirements in different documents are written on different levels, e.g., domain level, user level, and system level. Hence, there are different levels of details in the requirements, which means their length (number of words) varies significantly. Figure 1 shows the minimum, maximum, and average number of words per requirement for each of the 15 projects. As can be seen from the figure, all projects have very short requirements (less than 10 words). However when it comes to the average, the range is between 11 and 47 words. The difference is even more pronounced when we look at the longest requirements which vary between 39 and 181 words with one outlier that has 313 words. Additionally, the different documents contain requirements written in different styles. Some are written as user stories, others as instructions to potential vendors, or in a conversational language style.

3.2 Operational Definition of Security Requirements

We used the following criteria to define security requirements:

Table 2: Publicly available requirements specifications included in the study.

Spec	Project	Project type	Spec. type	Number of requirements	Security requirements	Pre-labeled
1	Mobile application for restaurants	Student project	SRS	98	18	Yes
2	University inventory management	Student project	SRS	27	11	Yes
3	Financial management system	Commercial project	RFP	180	100	Partially
4	Research data management system	Commercial project	SRS	92	9	Partially
5	Software development platform	Research project	BL	567	104	Partially
6	Financial management system	Commercial project	RFP	171	41	No
7	Financial management system	Commercial project	RFP	995	179	Partially
8	Customer relation management system	Commercial project	RFP	127	38	Yes
9	Electronic document management system	Commercial project	BL	253	34	No
10	Human resources management system	Commercial project	RFP	493	52	Yes
11	Global Platform Sepc. (GP)	Industrial project	SRS	177	63	Yes
12	Customer Premise Network (CPN)	Industrial project	SRS	210	41	Yes
13	Common Electronic Purse (ePurse)	Industrial project	SRS	124	83	Yes
14	Automated Driving Systems 2.0	Guideline	NTR	281	5	No
15	Critical Information Infrastructure Security Protection	Regulation	NTR	75	61	No
Total				**3,880**	**839**	

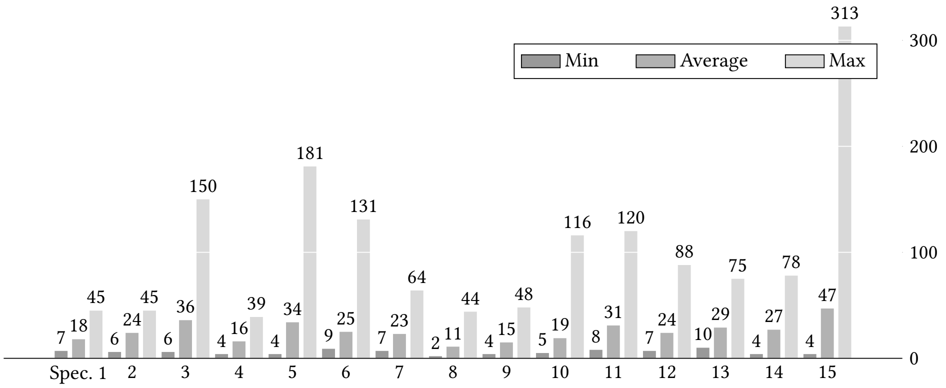

Figure 1: Number of words per requirement per spec.

- Requirements that address security properties of a system. This includes, but is not limited to, requirements for access control, data confidentiality, availability, data integrity, and logging.
- Requirements for managing security, e.g., vulnerability, threat, and asset analysis.
- Process-related requirements which include security, e.g., training employees on secure programming.
- Functional requirements with security constraints. An example is shown in the requirement below which limits access to the payment function to a certain group of authorized users: *"The system shall provide the ability for an authorized user to reissue a payment according to configurable business rules e.g., supervisor approval"* (Spec. 7)
- Requirements for securely implementing a feature or function, e.g., *"Ability to securely integrate with Microsoft Excel"* (Spec. 8)
- Requirements for integrating security products, e.g., the Lightweight Directory Access Protocol (LDAP).
- Requirements to comply with a certain security regulation, standard, or best practice, e.g., ISO 27001 [11].

3.3 Construction of the Ground Truth

As we were planning to use a supervised learning algorithm which requires labelled data, we labelled the requirements of the training data according to our operational definition. We differentiate two classes for the labelling: *(i)* security requirements are the *positive class*; *(ii)* all other requirements are the *negative class*.

As shown in the last column of Table 2, the requirements documents we gathered fall into three categories: **Not labelled:** For this data, we performed a manual labelling using the operational definition described in Section 3.2. This was done by the first author and double checked by an external collaborator, except for projects 14 and 15 which were labelled by an automotive cybersecurity domain expert and practitioner.

Pre-labeled: meaning that the requirements were already tagged as security / not security in the source specification documents. The first author revised the labels in these specifications for both security and functional requirements, as we identified issues with incorrect labels. The revision resulted in moving many requirements from the negative class to the positive class.

Partially labeled: This refers to specification documents which did not have specific labels for security, but rather a label to indicate non-functional or quality requirements. The first author went over the quality requirements to identify the security ones. The first author also went over the functional ones to identify additional security requirements. In particular specification 7 contained a number of such cases.

At the end, to assure the quality of the overall labelling, we created a subset of the data by randomly selecting 10% of the requirements from ten different projects including all projects labelled by the first author. Two senior researchers (the second and fourth authors) labeled this subset of the data without having prior knowledge of the original labels. The new labels were then compared to the original ones and they matched in approximately 90% of the cases. We also calculated inter-rater agreement coefficient (Cohen's kappa [6]) between each of the senior raters and the original one. The kappa values we obtained were 0.65 and 0.74 with the second and fourth authors respectively. These values indicate a substantial inter-rater agreement as per Landis et al. [16]. When checking the instances in which the raters disagreed, we found that they mostly were in Project 7. We accounted for that when we planned our experimentation as discussed in Section 3.4.

3.4 Experimental Set-up for RQ1

3.4.1 Cross-project prediction. In this experiment, we iteratively select one project and use its requirements as the testing set. The requirements from all other projects are put together as the training set, i.e., used to train the classifier. In each iteration, we compute the performance indicators. We repeat the process for each of the 15 projects in our data set. As mentioned in Section 3.3, we had some labelling disagreement in Specification 7 among the raters. We hence repeated the experiment after completely removing the data from project 7. The results indicate a minor drop in the precision and recall (0.7 % and 0.3 % respectively). Hence we decided to keep the data from Specification 7 to avoid cherry picking the data and hence biasing the results.

3.4.2 Selection of Classification Algorithm. Our problem is a binary classification to predict whether a textual requirement belongs to the positive class (security-related requirement), or the negative class (non-security related requirement). In early experimentation, we tested three of the most commonly used algorithms for text classification: Random Forest (RF) [10], Linear Support Vector Machine (Linear SVM) [24], and Naive Bayes (NB) [19]. Random Forest showed a performance advantage in this classification problem and, therefore, we committed to RF in this study. All results we report in the remainder of this paper refer to this classifier.

3.4.3 Data Pre-Processing. To prepare data for the classifier, we pre-processed the requirements specifications following standard procedures for machine learning. We used the machine learning library scikit-learn [23] for the entire study.

Remove noise: We removed punctuation and special characters by defining regular expressions using the *re* library in python, and applying them on the data.

Extract words: Since the requirements are written in natural language, the words are separated with spaces. Hence, we considered every sequence of characters between two spaces to be a word.

Ignore case: Since case sensitivity is not important in our problem, we converted all words to lower case.

Remove stop words: Stop words are the most commonly used terms in a language. These words appear in most sentences and thus do not contribute to classifying the requirement. We used the pre-defined *English* stop words list in scikit-learn.

Stemming: To ensure that words that stem to the same root, e.g., "authorization" and "authorize", are treated the same way, we used PorterStemmer in scikit-learn.

3.4.4 Feature extraction. To select a feature extraction method, we tested two different word embedding techniques, TF-IDF (Term Frequency-Inverted Document Frequency) and Word2Vec. We found TF-IDF to perform better, and we believe the reason to be the relatively small number of data points in the training set, which aligns with the findings of Cahyani and Patasik [2].

Hence, in this study we use the Bag of Words (BOW) representation of features, where each word of the textual corpus is considered a feature and TF-IDF which is a statistical measure that reflects how important a specific term is in a given corpus [18]. Term frequency gives weight to a term that appears frequently in a given document (requirement in our case), inverted document frequency gives weight to terms that are rare in the documents, and the final score is the multiplication of the two values. In our case, we expect the terms relevant to the positive class to be rare among all the requirements as the positive class is significantly smaller than the negative one. Hence, we expect TF-IDF to have good performance.

3.4.5 Up-sampling. Since we had significantly more non-security requirements than security ones in the data set, we used an up-sampling technique to achieve balanced classes in the training set for the machine learning model. After splitting the data into training and test sets, we applied the Synthetic Minority Oversampling (SMOTE) technique [4] on our training set. Rather than replicating the minority observations, SMOTE creates synthetic records based on the existing minority records.

3.5 Experimental Set-up for RQ2

To investigate **RQ2**, we hypothesised that the classifier can perform well in predicting security requirements and statements in a regulation. To test that, we constructed a test set of three regulatory documents recently entered into force, which gives them high relevance in the automotive domain at the time of writing.

Table 3 shows the data used for testing the classifier for **RQ2**. The three documents shown in the table are United Nations (UN) regulations and they are all taken from the automotive domain. The regulations vary in the magnitude of security relevance. UN-R155 includes 30 security requirements which is about 33% of the total number of requirements, whereas UN-R156 has 9% and UN-R157 includes less than 5%. Two of the regulations have 12 sections whereas the third has 13, and we have considered all the sections for testing even the ones concerning the scope and the definitions. The reason is that we wanted to see whether the classifier is able to identify definitions related to security as it does for actual requirements. In total, the regulations include 37 sections and 362 requirements and statements, out of which 47 are security related.

Table 3: Test data for RQ2

ID	Regulation	Number of sections	Number of requirements	Security requirements
UN-R155	UN Regulation No. 155 - Uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system [27]	12	91	30
UN-R156	UN Regulation No. 156 - Uniform provisions concerning the approval of vehicles with regards to software update and software updates management system [28]	12	97	9
UN-R157	UN Regulation No. 157 - Uniform provisions concerning the approval of vehicles with regard to Automated Lane Keeping Systems [29]	13	174	8
Total		**37**	**362**	**47**

Table 4: RQ1: Cross-project prediction results

				f-measures		
Spec.	Accuracy	Precision	Recall	f_1	$f_{1/2}$	f_2
1	94.9	88.2	83.3	85.7	87.2	84.3
2	96.3	100	90.1	95.2	98.0	92.6
3	84.4	92.9	78.0	84.8	89.4	80.6
4	97.8	88.9	88.9	88.9	88.9	88.9
5	93.1	84.2	76.9	80.4	82.6	78.3
6	91.8	86.5	78.0	82.1	84.7	79.6
7	94.9	81.4	92.7	86.7	83.4	90.2
8	94.5	100.0	81.6	89.9	95.7	84.7
9	96.4	93.1	79.4	85.7	90.0	81.8
10	95.7	94.3	63.5	75.9	85.9	67.9
11	76.3	63.0	81.0	70.8	65.9	76.6
12	82.4	53.7	70.7	61.1	56.4	66.5
13	75.8	94.9	67.5	78.9	87.8	71.6
14	95.9	29.4	100.0	45.5	34.2	67.6
15	84.0	93.0	86.9	89.8	91.7	88.0
Avg.	88.2	84.3	77	78.1	81.2	76.6

To mitigate the risk of bias, the labelling of the regulations was done by a different person (the third author) than the one who labelled the rest of the data set. This person is a security practitioner with over 10 years of experience in the automotive domain. We used the same pre-processed dataset and algorithm used in RQ1 to train the classifier in this experiment. We then looked at the aggregation of the results based on the structures of the regulations.

4 RESULTS

In this section, we present the results we got from running the experiments to answer our research questions. We analyze these results, and provide our answers to the research questions.

4.1 Data Analysis for RQ1

Our results indicate that a classifier trained on other projects is useful for an initial classification of a large set of requirements.

The results of the cross-project prediction are shown in Table 4. To evaluate the performance of the prediction model, it is insufficient to look only at accuracy, since the negative class (i.e., non-security-related requirements) dominates with 78% of total requirements. Hence, we focus on precision and recall. Additionally, we report different f-measures. Each row in the table corresponds to the case where the specification in the first column is used as the testing set. The averages of the projects are shown in the last row.

Table 5: RQ1: Comparison of our results with the ones from Knauss et al. [12] on the specifications used in that paper.

Spec.	Precision	Recall	f_1
11	+12	+25	+17.8
12	+27.7	-14.3	+21.1
13	+10.1	+26.5	+32.9
∅	+16,6	+12.4	+23.9

These results are significantly better than what the related work reports for cross-project prediction, as can be seen in Table 5, where we compare our results to those reported by Knauss et al. [12]. In that paper, the researchers found that when they trained a Bayesian classifier on two different projects, it fared poorly when applied to a third project. We hypothesise that a minimal number of specifications is necessary for the classifier to become general enough, as in our study we used requirements from 14 projects to train the model compared to two used by Knauss et al. If the classifier has been trained on a sufficient number of variations of a term, the likelihood that it will pick up a certain phrasing increases, thus improving recall. An example from our data is the term "sensitive data" in Specification 10 which is not used in any other specification. *We thus recommend to select training data that is as heterogeneous as possible.* In an industrial setting, this would require taking into consideration training the model with requirements specifications taken from different parts of the organisation, written by different analysts and practitioners.

If a classifier is robust w.r.t. different ways of phrasing requirements, it will be easier to train a cross-project classifier. We thus investigated the impact of heterogeneity of phrasing on the classifier. For this purpose, we regard the features the classifier uses to make a decision about the requirement. In our case, a feature is a word stem relevant to the detection of a security requirement and derived from the training data. We thus expect the most relevant features to contain stems such as "access" or "encrypt".

The results in Table 6 support this expectation. The most important features for the full data are based on generic security terms such as "security" and "authorize". These terms are indeed strong

Table 6: The 25 top-ranked terms used by the classifier when trained on different data sets and their importance.

Rank	Full data		Spec. 3		Spec. 7	
1	secur	23	author	23.4	secur	30.3
2	author	22.7	secur	20.8	authent	11.2
3	authent	10.3	authent	10.6	log	5.4
4	log	4	log	4.4	access	4.9
5	password	3.6	access	4	encrypt	4.2
6	vulner	3.2	protect	3	protect	3.5
7	access	3	vulner	2.7	author	3
8	psam	1.9	password	2.5	vulner	2.8
9	protect	1.8	encrypt	2	password	2.6
10	encrypt	1.8	login	1.5	login	1.8
11	signatur	1	sign	1	signatur	1.5
12	idm	0.7	**psam**	0.9	sign	1.2
13	host	0.5	mac	0.9	mac	0.8
14	messag	0.5	role	0.7	**psam**	0.8
15	login	0.5	audit	0.7	solut	0.7
16	sign	0.5	signatu	0.7	host	0.7
17	audit	0.4	sensit	0.6	onli	0.6
18	select	0.4	credenti	0.6	select	0.6
19	trail	0.4	**shall**	0.5	credenti	0.6
20	applic	0.3	**use**	0.4	devic	0.5
21	role	0.3	abil	0.4	**right**	0.5
22	privaci	0.3	**right**	0.4	role	0.5
23	provid	0.3	onli	0.4	**cng**	0.4
24	record	0.3	trail	0.3	audit	0.4
25	mac	0.3	**vendor**	0.3	**manag**	0.4

indicators for a security-related requirement. Other, more specific terms such as "password"or "login" score significantly lower and are less differentiating.

Table 6 shows some aberrations when considering the specific specifications (these are shown in bold in the table). For Specification 3, e.g., the term "shall" is ranked on position 19. This is due to the way the requirements are written ("the system shall"). Another interesting term is "user". In particular requirements that are written as user stories might contain this term in all requirements, not only the security-related ones. Finally, the term "vendor" appears in some specifications, in particular RFPs and is not exclusive to security-related requirements there. Additionally, there is the term "psam" which appears multiple times (22) in specification 13. The term refers to a security feature, hence the requirements in which it appears are security-related.

We also looked at the relationship between the achieved precision and recall values in the cross-project validation with respect to the total number of requirements and the number of security requirements in each specification. These are shown in Figures 2 and 3 respectively. As the trend lines show, the precision value tends to slightly decrease with higher number of requirements and increases with higher number of security requirements, whereas the recall value remains stable in both cases to a large extent. An example is specification 14 which has 281 requirements and only 5 security ones. The precision value of the cross-project validation for that project is only 29%, which is by far the lowest value. However, the recall value of the same project is 100%.

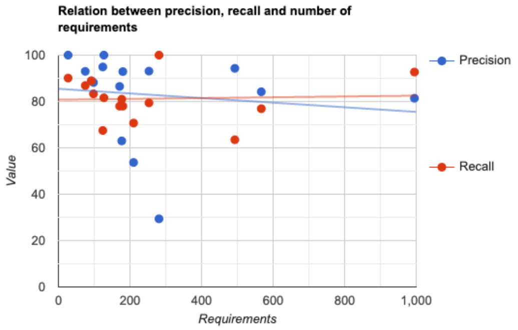

Figure 2: Precision and recall with respect to the number of requirements in each specification

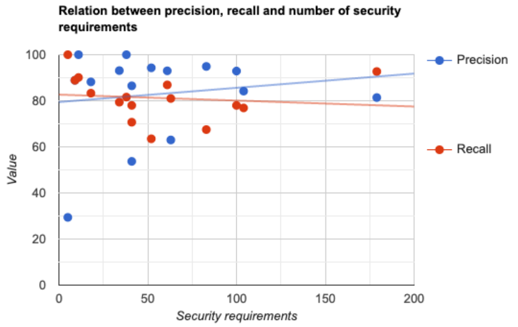

Figure 3: Precision and recall with respect to the number of security requirements in each specification

Table 7: RQ2: Classification of individual security-related *requirements* in regulatory documents

				f-measures		
Regulation	**Accuracy**	**Precision**	**Recall**	f_1	$f_{1/2}$	f_2
UN-R155	83.5	74.2	77	75.4	74,7	76,4
UN-R156	99	100	98	94	99,6	98,4
UN-R157	97.1	80	50	61.5	71,4	54
Avg.	93.2	84.7	72	77	76,6	76,3

4.2 Data Analysis for RQ2

Our results for the second research question indicate that a classifier trained on various requirements documents is useful to predict security requirements in regulatory documents. Table 7 shows the results of the classification of individual security related requirements in the three regulatory documents used for validation. We report the accuracy, precision, recall, and f-measures achieved by the classifier for each regulation.

To get a better insight of the results, we aggregated them based on the sections of the documents. Table 8 shows the section-based aggregated results of the three regulations. The first column refers to the section in the regulation, the second shows if the section includes at least one security requirements, the third indicates if the classifier was able to predict at least one security requirement in the section. Columns four and five show the total number of requirements in the section and the security related ones respectively. We report the precision, recall, and f-measures for identifying relevant sections in the regulations in Table 9. These results show higher precision compared to the original results except for UN-R157. This is due to a false positive prediction of Section 5 whose 35 requirements do not include a security-related one. The classifier erroneously identified *"protection against unintentional manual deactivation"* as security-related while our human experts marked that requirement as a safety concern. For the recall values, the aggregated results are better than those on individual requirements except for UN-R155, were there are two sections (3 and 8) including one requirement each which refer to security (out of six and four respectively), and the classifier was unable to accurately predict them.

5 DISCUSSION

In this section, we discuss the results of this study and how to interpret them, as well as provide our insights when it comes to the selection of training data in Section 5.1. We also discuss the limitations of the automated classification of requirements in Section 5.2.

5.1 Interpreting the Results

Our goal for the first research question (**RQ1**) was to determine which performance a classifier for security-related requirements can achieve when trained on data from other projects. The results in Table 4 indicate an average precision above 84% and 12 out of 15 project got a higher precision value than 84%. The recall, however, varies between 63.5% and 100% with an average score of 77%.

We believe that the performance we were able to achieve will be sufficient to support engineers in the initial labelling of a large set of requirements, in particular if the training data is heterogeneous enough and contains a large enough number of security-related requirements. All of our findings indicate that the significant factors for classification performance are heterogeneity and number of requirements in the training data.

Suitable training data should include use of relevant security mechanisms and standards. If the classifier is trained with training data that uses password authentication but the requirements to be labelled do not, the risk of false negatives increases and recall declines. Since we have selected the first ten requirement specifications that adhered to our criteria from a Google search, we have presented the classifier with a worst-case scenario.

High-quality labelling of the training data also has a major impact on classification quality. Hence, an extra effort to revise the labelling needs to be weighed against the benefits. Especially in cases where the unlabelled data to which the classifier should be applied is large, investing time in labelling a comparatively small set of training data might be significantly faster than a full manual labelling.

To revise the labelling, two of the authors of this study spent approximately 20 hours on the 3880 requirements included in the 15 specifications. Overall, we estimate that the effort of preparing the classifier, collecting publicly available data, cleaning up the data, relabelling it, and running the classification took approximately one entire work week for one person. When compared to the effort that can be spent on manually labelling reasonably large specifications, we believe this to be a worthwhile investment.

For the second research question (**RQ2**), we believe there are usage scenarios for the classifier on regulatory documents for which the achieved results can be sufficient. A classifier for security requirements would not replace the need to read the whole regulation. However, it can help prioritizing the documents which need direct attention from the security team by indicating their security relevance. If a company wants to enter a new market, e.g., it will need to identify relevant market-specific regulations and security-requirements within those regulations, as well as the already applied regulations to analyse any potential contradictions. This indicate also the need to classify the legacy documents. Quickly identify the most important documents can help stem the sheer amount of information the organisation needs to process.

Additionally, the classifier can help to prioritize the sections in the documents that need attention. This becomes more helpful when the documents are very lengthy. In our experiment, the results indicate that the classifier was able to identify the sections that are most relevant and contain the most security requirements. This can be an important input for engineers to focus their attention. A more detailed review of the entire document can then happen at a later point in time to address unidentified security-related requirements.

In general, the classifier performed well with high accuracy when classifying the individual requirements as shown in Table 7. Coupled with the even better results when identifying relevant sections in the documents as shown in Table 9, our results show that an organisation can benefit by being able to quickly digest a large number of documents and identify which (parts of them) are security-related with a reasonable level of confidence.

5.2 Limitations

We have seen limitations of the automated classification. In particular, short requirements that name specific security mechanisms, standards or unconventional technology will be difficult to classify automatically. Likewise, architectural elements with names that contain security-related terms or requirements that mention security-related terms without referring to cyber-security have a negative impact on precision and recall.

5.2.1 *Security mechanisms.*

The list of important features for the classifier depends on the security mechanisms in place. The term "password", e.g., is included in all three lists. However, a system does not necessarily have to be password protected. There would be differences in the list of terms if, e.g., a public-key infrastructure would be used. While there are no such issues in the specifications we used, we do observe that specific terminology, e.g., names of protocols are used without describing that they are security-related. Specification 10, e.g., contains the following two requirements: "*The solution should support LDAPv3*" and "*The solution should support Kerberos*". The fact that LDAP and Kerberos are used for authentication and are therefore security-related is not visible from these descriptions. Such requirements can negatively affect recall since they might not be picked up by a classifier who was not trained on requirements that used these terms in the context of requirements that contained other security-related terms. This issue does not, however, apply to regulatory documents as it is very uncommon that they mention any specific technologies.

A similar problem is the use of security standards by name or reference without a description. An example is the requirement

Table 8: RQ2: Aggregation of the predictions results of UN-R155, UN-R156, and UN-R157 based on the regulation's sections. Classification mismatches are italicised.

	UN-R155				**UN-R156**				**UN-R157**			
	Security		**# Requirements**		**Security**		**# Requirements**		**Security**		**# Requirements**	
Section	**reference**	**Pred.**	**Total**	**security**	**reference**	**Pred.**	**Total**	**security**	**reference**	**Pred.**	**Total**	**security**
1	Yes	Yes	4	2	No	No	1	0	No	No	1	0
2	Yes	Yes	13	1	No	No	11	0	No	No	28	0
3	Yes	No	6	1	No	No	7	0	No	No	4	0
4	No	No	7	0	No	No	7	0	No	No	9	0
5	Yes	Yes	14	6	No	No	4	0	*Yes*	*No*	50	1
6	Yes	Yes	13	2	No	No	12	0	*No*	*Yes*	37	0
7	Yes	Yes	22	18	Yes	Yes	43	9	No	No	7	0
8	*Yes*	*No*	4	1	No	No	4	0	Yes	Yes	17	6
9	No	No	3	0	No	No	4	0	Yes	Yes	9	1
10	No	No	2	0	No	No	2	0	No	No	5	0
11	No	No	1	0	No	No	1	0	No	No	3	0
12	No	No	1	0	No	No	1	0	No	No	2	0
13									No	No	2	0

Table 9: RQ2: Classification of security-related *sections* in standards UN-R155, UN-R156, and UN-R157.

Standard	**Precision**	**Recall**	$\mathbf{f}_1$	$\mathbf{f}_{1/2}$	$\mathbf{f}_2$
UN-R155	100	71.4	83.3	92.6	75.7
UN-R156	100	100	100	100	100
UN-R157	66.7	66.7	66.7	66.7	66.7

"*Hosting Service Provider will possess an ISO 27002 Certificate of Conformance or equivalent certifications*" from specification 3 and the explicit mention of HIPAA (a privacy standard for patient data) in Specifications 3 and 6. The use of security-related terms not explicitly tied to security is also present in Specification 3 in the requirement "*Solutions will have a fraud detection function*".

5.2.2 *Misclassification in the regulations.*

We consider the recall value to be the most important for the industrial usage scenarios of the classifier. Hence, we looked closely at the instances where the predictions results were false negatives. In UN-R155 there were a total of 7 false negatives. One main reason for these misclassifications is referring to the security issues as "risks" in the regulations. By examining our training set, the security issue are normally referred to using the term "vulnerability". Another issue is the use of the term "cybersecurity" in the regulation, whereas in the training set, the used term is "security" in 13 of the 15 specifications. This is an issue as "security" is one of the most important features for the classifier as indicated in Table 6.

In UN-R156 there was only one false negative instance, which is in Section 2 of the regulation. It refers to the definition of data integrity and describes it as errors or changes in the data, which was not considered by the classifier as security-related.

In UN-R157, there were eight security-related requirements and statements and the classifier was able to identify four of them, providing the worst recall value among the regulations. Three of the four false negatives were mainly on requirements about the availability and integrity of data, whereas the fourth was about taking measures against tempering of the system. Neither the term "manipulation" nor "temper" nor are among the most important features considered by the classifier.

To summarize, we have observed that the main reasons for the false negatives in the regulatory documents are the lack of data of the same type in the training set, and the implicit reference to security. We believe that in the latter cases, the classification is borderline and might even be hard for human experts to make.

5.2.3 *Other misclassifications.*

When analysing the results of the classification, we see several other misclassifications that have a negative impact on precision and recall. The naming of applications or modules in the system influences precision, e.g., when a part of the system is called "the secure analytical environment" as in Specification 3. Every time a requirement refers to that module, there is a chance that the classifier will tag this as a security requirement because of the term "secure". Likewise, when requirements refer to security other than cyber-security (e.g., in Specification 10: "*The system should be able to manage and contain data on different employee categories: Internal: Statutory SNE Interim Trainee ; External: Consultants Security Personnel*"), the requirement might be misclassified as a security-related requirement.

Other misclassifications can be observed in cases where requirements describe a UI part such as a login page and how a user interacts with it or include a sequence of events that includes a security-related action (e.g., "*After the user login he/she may submit a request*" in Specification 2). This affects precision since such requirements might be falsely classified as security-related.

5.2.4 *Addressing the Limitations.*

A possible extension of our approach is to use online learning to incorporate manually curated data from the project under investigation. This way, the initial learning on freely available data can be complemented with parts of the project's actual requirement specification that have already been labelled. There are a number of random forest variants for online learning (see, e.g., [15, 25]) that can be used for this purpose. Alternatively, a Bayesian classifier such as the one used by Knauss et al. [12] can be used. Such an approach would continuously improve predictive performance while the engineers revise the labelling suggested by an initial classification.

6 THREATS TO VALIDITY

In terms of **internal validity**, we consider the data labelling and the selection of the algorithm. All data was labelled by two persons. One person has previous experience working with security and requirements engineering, and the second is a practitioner with many years of experience in the security domain. However, there is a risk that judgement was subjective. To mitigate this risk, we performed the quality assurance step described in Section 3.3.

The *selection of the RF algorithm* was based on a preliminary run which compared three algorithms. It is possible that another algorithm had performed better. Similarly, we tested two feature extraction models, not including recent innovations such as BERT. However, our purpose was not to find an optimal approach, but rather to investigate if cross-project prediction was feasible.

In terms of **external validity**, we consider *overfitting* and *imbalanced data sets*. *Overfitting* is a common problem in machine learning [8]. It causes the classifier to perform very good during training and optimization, but very poorly when applied. This happens when the model learns too many details from the training data and fails to perform well when presented with new data. We believe that our experimentation setup of splitting the training and testing data based on different projects taken from different sources indicates that the results we got were not due to overfitting, and that the model will perform in accordance to these results when applied to other data. We used oversampling to tackle the issue of *imbalanced data sets*. This might have increased the likelihood of overfitting according to Sotiris et al. [13]. To reduce this risk, we used the synthetic minority oversampling technique rather than random oversampling and applied it after splitting the data set into training and testing data.

7 CONCLUSION

We have shown that it is feasible to train a classifier for security-related requirements on publicly available data and achieve a satisfactory classification performance when applying it to a new specification. Our results furthermore indicate the feasibility of using such a classifier in a real-life context to predict security-relevant requirements in regulatory documents. This has the potential to significantly reduce the effort of digesting new regulations and standards. Instead of spending significant time on manual classification of requirements in these documents, we show that it is feasible to spend relatively little time on preparing high-quality training data for use in a classifier to support the manual labour of the engineers.

In our future work, we will investigate the possibility to extend our approach towards online learning to seamlessly incorporate parts of the requirement specifications that have already been labelled and vetted. This will allow an organisation to build up a tailored classifier that can be used repeatedly and that adapts to new language being used in new and upcoming standards.

REFERENCES

[1] Z. S. H. Abad, O. Karras, P. Ghazi, M. Glinz, G. Ruhe, and K. Schneider. 2017. What Works Better? A Study of Classifying Requirements. In *RE'17*. 496–501.

[2] Denis Eka Cahyani and Irene Patasik. 2021. Performance comparison of TF-IDF and Word2Vec models for emotion text classification. *Bulletin of Electrical Engineering and Informatics* 10, 5 (2021), 2780–2788.

[3] Agustin Casamayor, Daniela Godoy, and Marcelo Campo. 2010. Identification of non-functional requirements in textual specifications: A semi-supervised learning approach. *IST* 52, 4 (2010), 436 – 445.

[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[5] Jane Cleland-Huang, Raffaella Settimi, Xuchang Zou, and Peter Solc. 2007. Automated classification of non-functional requirements. *Requirements Engineering* 12, 2 (2007), 103–120.

[6] Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20, 1 (1960), 37–46.

[7] European Union. 2016. General Data Protection Regulation (GDPR) 2016/679. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679

[8] Douglas M Hawkins. 2004. The problem of overfitting. *Journal of chemical information and computer sciences* 44, 1 (2004), 1–12.

[9] Tobias Hey, Jan Keim, Anne Koziolek, and Walter F Tichy. 2020. NoRBERT: Transfer learning for requirements classification. In *RE'20*. IEEE, 169–179.

[10] Tin Kam Ho. 1995. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, Vol. 1. IEEE, 278–282.

[11] International Organization for Standardization. 2013. ISO/IEC 27001 INFORMATION SECURITY MANAGEMENT, 1st Edition. https://www.iso.org/isoiec-27001-information-security.html

[12] Eric Knauss, Siv Houmb, Kurt Schneider, Shareeful Islam, and Jan Jürjens. 2011. Supporting requirements engineers in recognising security issues. In *REFSQ*. Springer, 4–18.

[13] Sotiris Kotsiantis, D Kanellopoulos, and P Pintelas. 2005. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering* 30 (11 2005), 25–36.

[14] Zijad Kurtanović and Walid Maalej. 2017. Automatically classifying functional and non-functional requirements using supervised machine learning. In *RE'17*. IEEE, 490–495.

[15] Balaji Lakshminarayanan, Daniel M Roy, and Yee Whye Teh. 2014. Mondrian forests: Efficient online random forests. In *Advances in Neural Information Processing Systems*. 3140–3148.

[16] J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33, 1 (1977), 159–174.

[17] Tong Li and Zhishuai Chen. 2020. An ontology-based learning approach for automatically classifying security requirements. *Journal of Systems and Software* 165 (2020), 110566. https://doi.org/10.1016/j.jss.2020.110566

[18] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge University Press.

[19] Melvin Earl Maron. 1961. Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM)* 8, 3 (1961), 404–417.

[20] Nuthan Munaiah, Andrew Meneely, and Pradeep K. Murukannaiah. 2017. A Domain-Independent Model for Identifying Security Requirements. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*. 506–511. https://doi.org/10.1109/RE.2017.79

[21] Daniel Ott and Frank Houdek. 2014. Automatic Requirement Classification: Tackling Inconsistencies Between Requirements and Regulations. *International Journal of Semantic Computing* 8, 01 (2014), 47–65.

[22] Barbara Paech, Robert Heinrich, Gabriele Zorn-Pauli, Andreas Jung, and Siamak Tadjiky. 2012. Answering a request for proposal–challenges and proposed solutions. In *REFSQ*. Springer, 16–29.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[24] John Platt. 1999. Sequential minimal optimization: A fast algorithm for training support vector machines. In *Advances in Kernel Methods-Support Vector Learning*.

[25] Amir Saffari, Christian Leistner, Jakob Santner, Martin Godec, and Horst Bischof. 2009. On-line random forests. In *ICCV Workshops*. IEEE, 1393–1400.

[26] Technical Committee ISO/TC 22/SC 32 Electrical and electronic components and general system aspects. 2021. ISO/SAE 21434, Road vehicles — Cybersecurity engineering.

[27] ECE/TRANS/WP.29 United Nations. 2021. UN Regulation No. 155 - Uniform provisions concerning the approval of vehicles with regard to cyber security and cyber security management system.

[28] ECE/TRANS/WP.29 United Nations. 2021. UN Regulation No. 156 - Uniform provisions concerning the approval of vehicles with regards to software update and software updates management system.

[29] ECE/TRANS/WP.29 United Nations. 2021. UN Regulation No. 157 - Uniform provisions concerning the approval of vehicles with regard to Automated Lane Keeping Systems.

[30] J. Winkler and A. Vogelsang. 2016. Automatic Classification of Requirements Based on Convolutional Neural Networks. In *RE'16 Workshops*. 39–45.

[31] QL Xu, RJ Jiao, X Yang, MG Helander, HM Khalid, and O Anders. 2007. Customer requirement analysis based on an analytical Kano model. In *IEEM*. IEEE, 1287–1291.