Decision Support for Security-Control Identification Using Machine Learning

Seifeddine Bettaieb[1], Seung Yeob Shin[1(✉)], Mehrdad Sabetzadeh[1], Lionel Briand[1], Grégory Nou[2], and Michael Garceau[2]

[1] SnT Centre, University of Luxembourg, Luxembourg City, Luxembourg
{seifeddine,shin,sabetzadeh,briand}@svv.lu
[2] BGL BNP Paribas, Luxembourg City, Luxembourg
gregory.nou@bgl.lu, mgarceau@cipherquest.com

Abstract. [**Context & Motivation**] In many domains such as healthcare and banking, IT systems need to fulfill various requirements related to security. The elaboration of security requirements for a given system is in part guided by the controls envisaged by the applicable security standards and best practices. [**Problem**] An important difficulty that analysts have to contend with during security requirements elaboration is sifting through a large number of security controls and determining which ones have a bearing on the security requirements for a given system. This challenge is often exacerbated by the scarce security expertise available in most organizations. [**Principal ideas/results**] In this paper, we develop automated decision support for the identification of security controls that are relevant to a specific system in a particular context. Our approach, which is based on machine learning, leverages historical data from security assessments performed over past systems in order to recommend security controls for a new system. We operationalize and empirically evaluate our approach using real historical data from the banking domain. Our results show that, when one excludes security controls that are rare in the historical data, our approach has an average recall of ≈95% and average precision of ≈67%. [**Contribution**] The high recall – indicating only a few relevant security controls are missed – combined with the reasonable level of precision – indicating that the effort required to confirm recommendations is not excessive – suggests that our approach is a useful aid to analysts for more efficiently identifying the relevant security controls, and also for decreasing the likelihood that important controls would be overlooked.

Keywords: Security requirements engineering · Security assessment · Machine learning

1 Introduction

Many IT systems, e.g., those used in the healthcare and finance sectors, need to meet a variety of security requirements in order to protect against attacks. The

E. Knauss and M. Goedicke (Eds.): REFSQ 2019, LNCS 11412, pp. 3–20, 2019.
https://doi.org/10.1007/978-3-030-15538-4_1

elaboration of these requirements is heavily influenced by the security controls prescribed by standards and best practices such as the ISO 27000 family of standards [14], NIST SP 800 guidelines [24], and OSA security patterns [25]. These controls define a wide range of technical and administrative measures for the avoidance, detection and mitigation of security risks [10]. An example security control from ISO 27002 is: "The integrity of information being made available on a publicly available system should be protected to prevent unauthorized modification." If an application has information assets with public access points, this control may be elaborated into detailed security requirements aimed at avoiding information tampering.

For a specific IT system in a particular context, only a subset of the controls in the security standards and best practices have a bearing on the security requirements. An important task that analysts need to do is therefore to decide which controls are relevant and need to be considered during requirements elaboration. Since the controls are numerous, performing this task entirely manually is not only cumbersome but also error-prone, noting that deciding whether a certain control is relevant often correlates with several contextual factors, e.g., the assets that are associated with a given system, the threats that the system is exposed to, and the vulnerabilities that the system leads to. Overlooking any of these factors can lead to wrong decisions about the security controls, and potentially serious consequences. This problem is made even more acute by the scarcity of expertise in security risk analysis in most organizations.

Our work in this paper is motivated by the need to provide automated decision support for identifying the security controls that are pertinent to a specific system. To this end, we observe that, in security-critical sectors, e.g., finance, security assessment is an increasingly systematic activity, where security assessment data is collected and recorded in a structured way [7]. Many system providers and security consulting firms now have detailed data models in place to keep track of the security-related properties of the systems that they analyze and the decisions they make regarding security. This raises the prospect that existing (historical) data about security assessments can be put to productive use for decision support. What we do in this paper is to examine the feasibility and effectiveness of this prospect in a real setting.

The starting point for our work was a year-long field study at a major international bank. Our study aimed at developing insights into industry practices for assessing IT security risks. The study focused specifically on early-stage security assessments during the system inception and requirements elaboration phases. This study led to a precise characterization of the historical data that we had at our disposal for building automated decision support. While the data model resulting from our field study inevitably has bespoke concepts that are specific to our study context, the majority of the concepts are general and aligned with widely used standards, particularly ISO 27001 and 27002. This helps provide confidence that our data model is representative of a wider set of security practices than our immediate study context.

With a data model for security assessments at hand, we explore the use of several *Machine Learning (ML)* algorithms for identifying the security controls that are most relevant to a given system and context. To this end, we define a set of features for learning from historical security assessment data. We empirically evaluate the accuracy of our approach using real data. Our results show that, when one excludes security controls that are rare, i.e., apply to too few systems in the historical data, our approach on average has a recall of ≈95% and precision of ≈67%. Since recall is high and the number of false positives is not excessive, as suggested by precision, we conclude that ML is a promising avenue for increasing the efficiency of identifying relevant security controls, and also reducing the likelihood that important controls would be missed. In situations where one has to deal with rarely used security controls, ML alone is not sufficient; this necessitates future investigations into how ML can be complemented with other techniques, e.g., guided manual reviews, expert rules and case-based reasoning, in order to provide comprehensive coverage of the security controls.

The rest of the paper is organized as follows: Sect. 2 provides background and compares with related work. Section 3 summarizes the outcomes of our field study on security assessment. Section 4 presents our ML-based approach for recommending relevant security controls. Sections 5 and 6 report on our evaluation. Section 7 discusses threats to validity. Section 8 concludes the paper.

2 Background and Related Work

This section discusses the industry standards and the existing research strands related to our work.

2.1 Information Security Standards

Our collaborating partner has its IT security practices grounded in the ISO 27000 family of information security standards [14]. This commonly used series of standards provides a systematic approach for handling information security. Among these standards, ISO 27001 and 27002 relate most closely to our work in this paper. ISO 27001 specifies a set of requirements for developing and maintaining an information security management system. The standard further envisages requirements for the assessment and control of the security risks posed by security breaches in IT systems. ISO 27002 complements ISO 27001 by providing guidelines for selecting, implementing, and managing controls for security risks. The standard has a total of 128 security controls. These controls span 11 security categories, e.g., security policy, asset management, and access control. When elaborating the security requirements for a system, one has to identify the controls that are relevant to the system at hand. As noted earlier, performing this task without automated assistance is both tedious and prone to errors. Our work in this paper takes aim at providing suitable automated support for the above task.

2.2 Security Requirements Engineering

Security requirements have been widely studied for IT systems, e.g., [6,12,16, 19,22,30,32]. The most closely related research threads to our work are those concerned with early-stage security risk analysis. Two notable techniques to this end are STRIDE and DREAD, both originating from Microsoft [20]. These techniques have been used and improved by many corporations over the years [22]. STRIDE is a method for classifying security threats, whereas DREAD is a method to rate, compare and prioritize the severity of the risks presented by each of the threats classified using STRIDE. Our work is complementary to STRIDE and DREAD, first in that we focus on risk mitigation as opposed to risk classification and triage, and second in that we take an automation angle rather than dealing exclusively with manual security analysis.

Some prior research attempts to assist security engineers through capturing domain expertise in a reusable form. For example, Schmitt and Liggesmeyer [29] propose a model for structuring security knowledge as a way to improve the efficiency of specifying and analyzing security requirements. Sindre and Opdahl [31] develop a systematic approach for security requirements elicitation based on use cases, with a focus on reusable methodological guidelines. In contrast to the above work, we explore how historical data from past security assessments can be mined and reused within a corporate context for building automated decision support. We further demonstrate the effectiveness of our approach through systematic empirical means by applying the approach to an industrial case study.

2.3 Applications of Machine Learning in Requirements Engineering

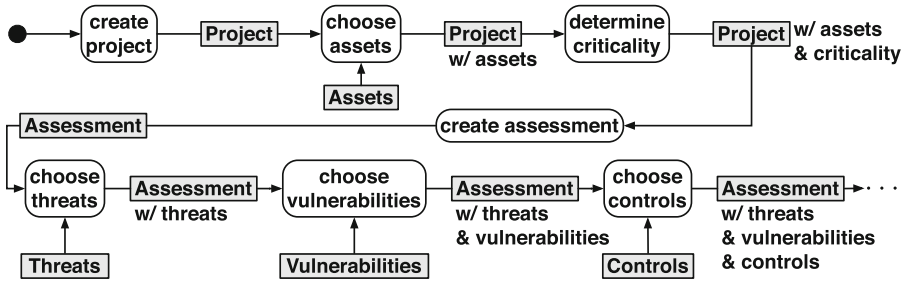
ML has generated a lot of traction in Requirements Engineering for supporting a variety of tasks, e.g., extracting user-story information [28], identifying non-functional requirements [3], and requirements classification [17]. To the best of our knowledge, we are the first to attempt applying ML for generating automated recommendations for security controls using historical data.

3 Field Study on Security Assessment

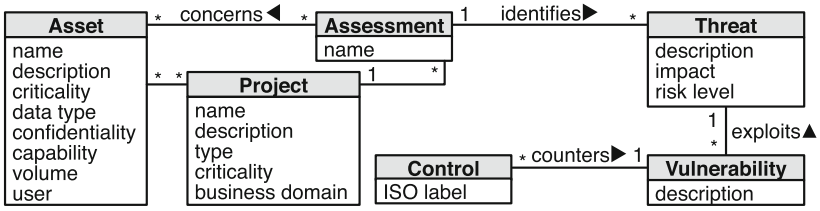
This section describes the results of a field study conducted with the goal of building insights into how IT security assessments are done in practice. We started our study with meetings with IT security specialists at our collaborating partner (a bank). Subsequently, the first author spent approximately a year onsite at the partner's headquarters, learning about the details of the security assessment process followed there and the data model that underlies this process.

The security assessment process at our partner is a customized procedure shaped around the guidelines of the ISO 27000 standards [14]. A central goal of this process is to derive, for a given system, a set of ISO-specified controls that need to be elaborated further into security requirements.

In Fig. 1(a), we show an overview of the security assessment process gleaned from our field study, and in Fig. 1(b) – a (simplified) version of the underlying



(a) Security assessment process



(b) Data model

Fig. 1. Main outcomes of our field study; note that the outcomes have been scoped to the analytical task pursued in this paper (i.e., decision support for security controls).

data model. While we present the security assessment process in a sequential manner, in practice, the process is iterative. This means that before they are finalized, the choices and the decisions made during assessment may undergo multiple rounds of improvement based on the findings at the different steps of the process. The data model of Fig. 1(b) is populated incrementally as the assessment workflow unfolds, with each step of the workflow adding new information. As we describe in Sect. 4, we use this data model as the basis for defining features for learning from past security assessment records.

As shown in Fig. 1(a), security assessment starts with the "create project" step. A new project represents a system-to-be that is at the inception and requirements gathering stage. In this step, the basic information about a project is specified, e.g., project description and business domain. Next and in the "choose assets" step, the analysts define and link the assets relevant to a given project. In general, an asset can be defined as a resource with economic value that is held or controlled by an individual, corporation, or country [13]. The step is followed by the "determine criticality" step where the analysts, in collaboration with the business stakeholders, decide about project criticality. The more critical a project is, the more is the need to evaluate potential threats and vulnerabilities systematically. To evaluate the criticality of a project, the analysts fill out a security questionnaire comprised of 12 multiple-choice questions. Each question covers a possible aspect of exposure, e.g., the level of exposure to external attacks. Once the questionnaire has been completed, the

analysts exercise expert judgment to decide the project criticality level and update the project information accordingly.

The "create assessment" step captures various contextual information about the assets that have been linked to a project. The (data) type of an asset is determined by the content that the asset stores, processes, or transfers. The classification of asset types at our partner is based on their in-house domain expertise and the guidelines of the national data protection authority. The confidentiality of an asset is determined by how sensitive its content is. This attribute is a value on an (ordinal) scale ranging from public to secret. The criticality of an asset is a quantitative score indicating risk exposure. This score determines whether the potential risk posed by an asset is significant enough to warrant additional security analysis. The score is derived from the following asset attributes through a combination of expert judgment and rules: (1) the capability attribute, capturing the output channels to which the content of an asset can be sent, (2) the volume attribute, capturing the volume of data that an individual transaction can read, write, or delete from an asset, and (3) the user attribute, estimating in a logarithmic scale the number of users that can access an asset. We note that for an individual project, our partner may conduct multiple assessments from different perspectives and involving different groups of analysts. In this paper, when we refer to an assessment, we mean the collection of *all* assessment activities performed over a given project. Consequently, the assessment information collected per project is the union of the outcomes of all the assessment activities performed.

Once the contextual information for the assets in a project has been specified, the analysts move on to the identification of threats and vulnerabilities, and subsequently, the security controls. A threat refers to anything that has the potential to cause serious harm to a system, e.g., unauthorized disclosure of confidential information [13]. Threats are identified in the "choose threats" step of the process of Fig. 1(a). In this step, the analysts carefully examine a threat catalog consisting of 22 threat items and decide which ones are applicable. If a threat is deemed applicable to a project, the analysts qualify the threat more precisely within the context of that project. Specifically, for each applicable threat, the analysts provide a description, choose an appropriate risk level, and determine whether the threat impacts confidentiality, integrity, availability, or traceability. Next, in the "choose vulnerabilities" step, the analysts decide about the applicable vulnerabilities. A vulnerability represents a weakness that can be exploited by a threat, leading to the risk of asset damage or exposure [13]. An example vulnerability would be "oversight in defining access control rules to shared information". At our partner, vulnerabilities are identified using a predefined catalog with 156 entries. This catalog encompasses all the vulnerabilities known to the partner in its application domain.

Finally, in the "choose controls" step, the analysts select the appropriate security controls for the project being assessed. The source for the security controls at our partner is the ISO 27002 standard [13]. We thus refer to these controls as ISO controls. The catalog of ISO controls used by our partner constitutes 134 entries. In the remainder of this paper, we develop and evaluate automated

decision support for choosing ISO controls. In particular, assuming that the assessment process steps prior to choosing the controls have been already performed for a project, we propose an approach based on ML for recommending the ISO controls relevant to the project.

Before we move on to presenting our approach, it is important to note that, for simplicity and succinctness, we have scoped the security assessment process and data model in Fig. 1 to what is needed for recommending ISO controls. In particular, we have left out of this figure and our explanation thereof a number of activities, e.g., risk mitigation and residual risk analysis, which take place after ISO-control selection.

4 Approach

Our approach for recommending ISO controls is based on ML. In this section, we present the main principles and considerations behind the approach.

4.1 Source Data for Building a Classification Model

To build a classification model, we utilize the database of historical assessment records at our collaborating partner. This database covers all the systems assessed by the partner in the past nine years. From this database, we extract various attributes. Our attributes, which are based on the data model of Fig. 1(b), are discussed next.

4.2 Machine Learning Features

We engineered our features for learning through a joint endeavor with the IT security specialists at our partner. Table 1 presents our feature set alongside our intuition as to why each feature may be a useful indicator for the relevance of ISO controls. Essentially, we chose a feature for inclusion in the set if we deemed the feature to be characterizing an important aspect of security assessment. For instance and as shown in Table 1, the criticality attribute of a project is used as a feature. In contrast, the name attribute of a project is not, since the name has no impact on the identification of ISO controls. The ISO controls (not shown in the table) are treated as class attributes. We build one classifier per ISO control. The class attribute for each ISO control is thus a binary value indicating whether or not the control is relevant to a given project.

4.3 Dealing with Imbalance

An important issue we have to take account of in our approach is imbalance in our security assessment data. In particular, we observe that the absence of ISO controls is much more prevalent than their presence across the projects. This imbalance is caused by the relatively infrequent use of several ISO controls. When a class – in our context, a particular ISO control being applicable – is

Table 1. Our features for machine learning.

Feature	(D) Definition and (I) Intuition
Project type	(D) The type of a project (total of 3 types: usual business, large scale and integration project). (I) Each project type implies a different scale and a specific process for handling risks
Project criticality	(D) The criticality of a project (total of 3 levels: very critical, critical, and non-critical). (I) The more critical a project, the more stringent are the security controls
Business domain	(D) The area of business under which a project falls (total of 48 domains, e.g., web banking, wealth management). (I) The feature relates to how severe the consequences of a breach are. For example, a breach may have more severe implications in wealth management than in certain other domains due to the involvement of vital client information
Business domain category	(D) The category for a group of business domains (total of 4 categories, e.g., the HR category, which encompasses all the human-resource-related business domains). (I) The feature provides an extra layer of abstraction for distinguishing different business domains
Security answers (A1..A12)	(D) The answers provided by the analysts to the questions on a static security questionnaire (total of 12 questions). An example question is: "What is the project's level of exposure to external attacks?" All answers are on a three-point scale: low, significant, very high. (I) The answers serve as an indicator for the seriousness of potential security breaches
Number of assets	(D) The number of assets linked to a project. (I) Increasing the number of assets may lead to an increased attack surface, thus warranting more rigorous controls
Number of critical assets	(D) The number of critical assets in a project. (I) Critical assets are specific entities with major importance. If these assets are compromised, the effects are more serious than those for regular assets. Critical assets may necessitate more security controls
Number of assets per category (C1..C9)	(D) The number of assets in an asset category (total of 9 categories, e.g., mobile application or database). (I) Each asset category has a different impact on the security controls in a project. For example, a client database being compromised would typically have more serious consequences than, say, a mobile application being inaccessible
Number of users	(D) The maximum number of users who can access the data of a project. (I) The potential risk of data exposure is correlated to the number of users accessing the data
Data type	(D) The most sensitive type of data in an asset (total of 4 types, e.g., personal data). (I) The more sensitive the data, the more impact a breach would have
Capability	(D) The capability of extracting data (total of 3 modes: screen, print, and electronic). Screen means that a user can view the data on a screen. Print means that a user can print the data on paper. Electronic means that a user can store the data onto an electronic device. (I) Data exposure risks increase as one goes from screen to print to electronic data extraction. The security controls required may thus be impacted by the extraction capability
Volume	(D) The volume of data that can be read, written, or deleted by one data transaction (total of 3 types: record-by-record, percentage-per-day, and unlimited). Record-by-record means that a user can access only one record at a time. Percentage-per-day means that a user can access a certain percentage of the dataset in one day. Unlimited means that a user has unlimited access. (I) The risk of data exposure correlates with volume. Volume may thus have an influence on the security controls
Confidentiality	(D) The maximum confidentiality level of the assets in a project (total of 4 levels: public, restricted, confidential, secret). (I) The higher the confidentiality level, the more severe are the consequences of a breach. The security controls may thus be influenced by the level of confidentiality
Threats (T1..T22)	(D) The presence or absence of a threat (total of 22 threats). (I) Threats exploits vulnerabilities. The presence of a threat has a direct influence on the security assessment decisions, including those related to the security controls
Threat impact (S1..S4)	(D) Impact scores based on all the threats in a project. Separate scores are computed for confidentiality (S1), integrity (S2), availability (S3), and traceability (S4). (I) The scores relate to the impact of security breaches and thus may influence the controls
Risk (R1..R22)	(D) Estimated risk of each threat on a scale of 1–8 (negligible to extremely high). (I) The risk posed by a threat influences security decisions, including those about the security controls
Vulnerability (V1..V154)	(D) The presence or absence of a vulnerability (total of 154 vulnerabilities). (I) Security controls counter vulnerabilities, and are naturally affected by which vulnerabilities apply

rare, ML classification models have a tendency to predict the more prevalent classes [1]. In our context, this means that, unless steps are taken to counter imbalance for rarely used ISO controls, any classification model that we build may invariably find the rare ISO controls inapplicable. To tackle imbalance, we examine two commonly used methods, namely synthetic minority over-sampling technique (SMOTE) [4] and cost-sensitive learning (CSL) [8].

4.4 Choice of Classification Algorithm

We elect to use interpretable ML techniques to provide analysts not only with security control recommendations, but also the rationale behind how the security controls were selected. An interpretable model would explain how and why a specific decision was made concerning a particular security control. For instance, the model would indicate that a particular ISO control is selected mostly because a certain combination of threats and vulnerabilities is present. We note that, in this paper, we do not attempt to validate the resulting ML models with domain experts. Nevertheless, scoping our work to interpretable ML is important, because experts are unlikely to accept decisions for which they are not provided an explanation.

5 Case Study

We evaluate our approach through an industrial case study from the banking domain. The case study is a follow-on to our field study of Sect. 3 and was conducted with the same industry partner.

5.1 Research Questions

Our case study aims to answer the following research questions (RQs):

RQ1 (classification): *Which classification algorithm is the most accurate at recommending security controls?* The accuracy of our approach is partly driven by the selected ML algorithm. In RQ1, we examine standard classification algorithms based on the existing best practices in the literature [23], and compare the accuracy of the resulting classifiers.

RQ2 (features): *Which features are the most influential for recommending security controls?* Features used in constructing an ML-based classifier typically have different degrees of importance toward the classifier's decision making. In RQ2, we evaluate the importance of the features in Table 1.

RQ3 (usefulness): *What is the overall utility of our approach?* For our approach to be useful in practice, the decision support must propose sufficiently accurate security controls in practical time. RQ3 measures the accuracy of our security recommendation system at the level of projects alongside the execution time of the main steps of our approach.

5.2 Implementation

Our recommendation system is built using the Weka framework [11]. Weka supports a broad spectrum of ML techniques. We ran our experiments on a computer equipped with an Intel i7 CPU with 16 GB of memory.

5.3 Case Study Data

Our raw data is a database of 274 assessment projects conducted over a span of nine years, from 2009 until present. Of these assessment projects, we excluded 47 because they either were not carried through to completion, or were built for testing and training purposes. This leaves us with 227 assessment projects for evaluating our approach.

Among the controls introduced by ISO 27002, some never or too rarely appear in our data. Based on our ML expertise and feedback from security engineers, we excluded the ISO controls that had been used less than 5 times within the selected 227 assessment projects. The applicability of such ISO controls cannot be predicted meaningfully using ML. In summary, our experimental dataset provides values for all the features in Table 1 and 77 ISO controls across 227 assessment projects.

5.4 Experimental Setup

To answer the RQs in Sect. 5.1, we performed three experiments, EXPI, EXPII and EXPIII, as described below.

EXPI. This experiment answers RQ1. We select the following interpretable ML algorithms as candidates for building our recommendation system: Naive Bayes [15], Logistic Regression [18], J48 [27], CART [2], JRip [5], and PART [9]. EXPI compares the accuracy of these six alternatives using the features of Table 1.

We start EXPI with hyper-parameter optimization (HPO) for the six alternatives considered. In doing so, we also account for the data imbalance problem described in Sect. 4.3. As noted in this earlier section, we consider two techniques for handling imbalance: SMOTE and CSL. SMOTE resolves imbalance by adding new artificial (synthetic) minority samples to the dataset. CSL mitigates the bias of the classifier toward the majority class by assigning a larger penalty to either false positives or false negatives. In our context, we levy a larger penalty on false negatives, i.e., ISO controls that apply to a project but are erroneously classified as not relevant. The proportional prevalence of the majority versus the minority (rare) class in our experimental dataset rounds up to 12 to 1. Specifically, our dataset contains 17952 instances of the majority and 1548 instances of the minority class. We use the same ratio for CSL by setting the cost values of false negatives and false positives to 12 and 1, respectively. Note that the cost values of true positives and true negatives are zero.

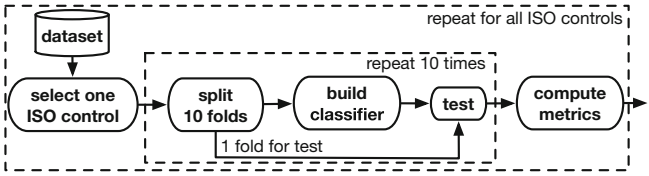


Fig. 2. 10-fold validation for all ISO-control classifiers.

For HPO, we use a step-wise grid search algorithm [21] that starts with a first coarse grid search and then refines the areas of good accuracy with additional finer-grained grid searches. For example, to find an optimal value of a real-type hyper-parameter, at the first search iteration, $i = 1$, we vary the parameter value within the valid range of the parameter by $s_i = 0.1$ step width. After finding the best parameter value, b_i, at the first search iteration, we adjust the step width, s_{i+1}, by $s_i \times 0.1$ (e.g., 0.01 at the second iteration) and adjust the search range for the parameter to $[b_i - s_i, b_i + s_i]$ for the next iteration. We continue the iterations until the difference between the best accuracy values found at the ith and $i - 1$th iterations are less than 0.01. Note that our HPO searches all the possible values in the valid ranges of the integer- and enum-type parameters at the first iteration, and then uses the best-found values at the subsequent iterations for tuning real-type parameters.

Following HPO, we measure through cross validation the accuracy of the alternative ML algorithms for predicting ISO controls. The cross validation process is illustrated in Fig. 2. The "repeat 10 times" block in the figure applies standard 10-fold cross validation [23] to the classifier built for an individual ISO control. This is repeated for all the ISO controls through the "repeat for all ISO controls" block. At the end, the "compute metrics" step calculates the EXPI accuracy metrics described in Sect. 5.5.

EXPII. This experiment answers RQ2. We evaluate the importance of the features in Table 1 based on the best-found configuration in RQ1. For each of the ISO-control classifiers, we rank the features using a standard metric for feature evaluation, as we discuss in Sect. 5.5. We then identify and aggregate the most influential features across all the ISO controls.

EXPIII. This experiment answers RQ3 by examining how much useful assistance one can expect from ML for identifying the ISO controls relevant to a given assessment project. Specifically, EXPIII performs the leave-one-out validation process shown in Fig. 3. The "leave one project out" step takes one project out from the dataset. The remaining dataset is then utilized for training the classifiers of all the ISO controls. Subsequently, the withheld project is used for testing the trained classifiers, as shown in "repeat for all ISO controls" block of the figure. This is repeated for all the projects in the dataset, as indicated by the "repeat for all projects" block. At the end of the process, we compute

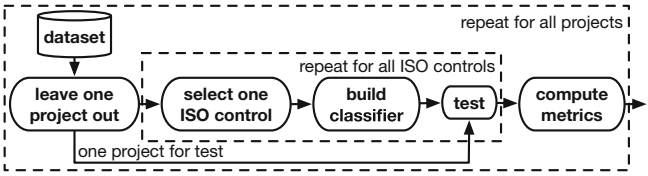


Fig. 3. Leave-one-out validation for all projects.

the EXPIII accuracy metrics described in Sect. 5.5. We note that these accuracy metrics are only indicative of in-vivo usefulness; the metrics have to be considered in their application context for a more definitive evaluation. Doing so requires user studies and is beyond the scope of this current paper.

5.5 Metrics

In EXPI, for a given ISO control c, we define the precision and recall metrics as follows: (1) precision $P^c = TP/(TP + FP)$ and (2) recall $R^c = TP/(TP + FN)$, where TP, FP, and FN are the sum of the true positives, false positives, and false negatives, respectively, across the 10 folds of cross validation for ISO control c. A true positive is a project to which c is relevant and is correctly predicted as such; a false positive is a project to which c is not relevant but is incorrectly predicted to have c as a control; a false negative is a project to which c is relevant but is incorrectly predicted to not have c as a control. These metrics are used for comparing the accuracy of different ML algorithms.

In practice, the decision as to whether an ISO control is applicable should be made as simple as possible to minimize the effort needed from the analysts. The most critical factor here is recall, since the presence of false negatives implies that important ISO controls may be missed. A recall that is too low would thus undermine the usefulness of the approach, meaning that the analysts would be better off doing the selection of the relevant controls entirely manually. To allow the analysts to focus only on the recommended controls, we prioritize recall over precision.

In EXPII, we use the gain ratio metric [26]. This metric, which is commonly used for ranking ML features, is a modification of the information gain metric, aimed at reducing bias on multi-valued features.

In EXPIII, we define precision and recall around a project. This is in contrast to EXPI, where these notions were defined around an ISO control. Let p be the project withheld from the set of all projects in a given round of leave-one-out validation. We define (1) precision P^p as $TP/(TP + FP)$ and (2) recall R^p as $TP/(TP + FN)$, where TP is the number of relevant ISO controls correctly predicted as such for project p, FP is the number of ISO controls that are not relevant to project p but are incorrectly predicted as being relevant, and FN is the number of relevant ISO controls incorrectly predicted as not being relevant to project p. These precision and recall metrics are used for measuring overall accuracy at a project level.

6 Results

In this section, we answer the RQs of Sect. 5.1 based on the results of our case study.

6.1 RQ1

Table 2 shows the results of EXPI, described in Sect. 5.4. Specifically, the table reports the average precision and recall – average P^c and R^c, defined in Sect. 5.5, across all ISO controls – of the six alternative ML classification algorithms considered.

As we argued previously, in our application context, recall has priority over precision. The results of Table 2 thus clearly suggest that J48, which yields an average recall of 94.95% and average precision of 65.90%, is the best choice among the ML classification algorithm considered. For all classification algorithms, including J48, handling imbalance via CSL leads to substantially more accurate classification, when compared to doing so via SMOTE. When J48 is applied alongside CSL with a cost ratio of 12 to 1 for false negatives versus false positives (see Sect. 5.4), the optimal hyper-parameters are as follows: *pruning confidence = 0.001* and *minimal number of instances per leaf = 7*.

Table 2. Comparison of the average precision and recall of different ML classification algorithms with optimized hyper-parameters.

Algorithm	CSL		SMOTE	
	P^c (avg.)	R^c (avg.)	P^c (avg.)	R^c (avg.)
J48	65.90	**94.95**	77.11	78.15
CART	55.11	92.42	76.03	64.20
JRip	64.32	91.35	74.68	79.49
PART	69.19	92.89	73.63	76.74
Logistic regression	64.32	51.54	68.77	58.91
Naive Bayes	33.35	61.59	17.02	50.93

The answer to **RQ1** *is that* J48 combined with CSL leads to the most accurate classification. Using this combination, we obtained an average recall of 94.95% and average precision of 65.90% in our case study.

We answer RQ2 and RQ3 using J48, CSL, and the best hyper-parameter values mentioned above.

6.2 RQ2

As explained in EXPII of Sect. 5.4, we use gain ratio for estimating the importance of our features (Table 1). Based on the gain-ratio scores of the features across all the ISO-control classifiers, we make the following observations:

1. There are 12 vulnerabilities that have a zero gain ratio in all the classifiers. A subsequent investigation revealed that the vulnerabilities in question are not present in any of the past projects. We excluded these vulnerabilities from the dataset. The impact of this exclusion on precision and recall is negligible.

2. With the above 12 vulnerabilities removed, we observed that different ISO-control classifiers use different but overlapping subsets of features. This indicates that the decision about the relevance of different ISO controls is influenced by different factors. The feature subsets were picked automatically by J48's internal feature selection mechanism as implemented in Weka (this mechanism is also based on gain ratio).

In light of the second observation above, we answer RQ2 by measuring the overall importance of the features across all the classifiers. To do so, we first aggregated the top five most important features based on the rankings obtained from the different classifiers. We then computed the importance of a set F of features of the same type (e.g., vulnerability features: V1 to V154 in Table 1) as the percentage of the number of classifiers having some feature of F in their top five most important features. Table 3 shows the results. For example, all (100%) of the classifiers have some vulnerability in their top five most important features. The domain experts in our study stated that the results of Table 3 were consistent with their intuition about the most important factors in determining the relevance of ISO controls.

Table 3. Most important features for ISO-control classification.

Vulnerability	Risk	# of assets per category	Threat impact	Threat	Security answer	# of critical assets
100%	62.80%	16.00%	15.38%	12.80%	2.50%	1.20%

The answer to **RQ2** *is that* overall and in descending order of magnitude, vulnerabilities, risks, the number of assets per category, threat impacts, threats, security answers, and the number of critical assets are the most influential feature groups. This finding is consistent with the intuition of the security specialists in our case study.

6.3 RQ3

Figure 4 summarizes through a boxplot the results of EXPIII, described in Sect. 5.4. Specifically, the boxplot shows the distributions of precision, P^p, and recall, R^p, as defined in Sect. 5.5. On average, our approach has a recall of 94.85% and precision of 67.38% when tasked with identifying the ISO controls relevant to a given project. The high recall suggests that the analysts can focus most of their attention on the recommended

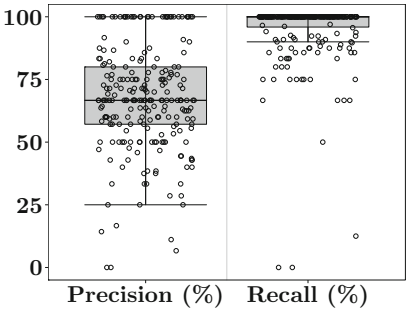


Fig. 4. Precision and recall distributions resulting from leave-one-out validation.

ISO controls, since the recommendations most likely contain all the relevant controls. The precision is reasonable too: On average, our approach recommends 9.4 ISO controls – both true and false positives – for a project. Of these, one can expect an average of 6.3 recommendations to be correct and 3.1 to be incorrect. The domain experts in our study confirmed that, given the small number of recommended ISO controls, they can vet the validity of the recommendations efficiently.

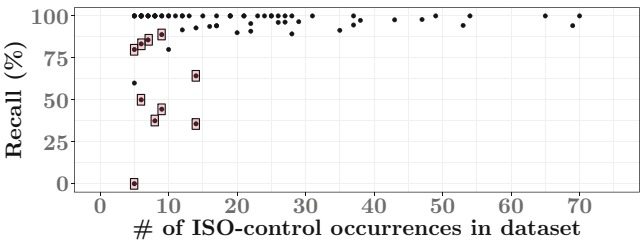


Fig. 5. Recall values of the ISO classifiers.

From Fig. 4, we further observe that the recall (R^p) for eight out of the total of 227 projects in our dataset is below 75%. Upon a follow-up investigation, we determined that the root cause for low recall in these projects is that the majority (and in two cases, all) of the ISO controls relevant to these projects have low prevalence in the dataset. In Fig. 5, we plot the recall of each ISO-control classifier (R^c) against the prevalence of the respective ISO control in the dataset. The ten datapoints encircled by □ represent the ISO controls that bring about low recall in the above-mentioned eight projects. A complementary insight from Fig. 4 is that recall is highly stable for those ISO controls that occur ≥ 15 in our dataset. As noted previously, handling less frequent ISO controls requires complementary techniques and is the subject of future work.

With regard to execution time, we make the following remarks: Generating J48 classification models for all the ISO controls subject to our experiments took 215 s in total; this gives an average training time of 2.8 s per ISO control. With the classifiers built, issuing recommendations for a given project takes an average of 1.7 s. These results suggest that our approach is scalable.

The answer to **RQ3** *is that*, based on our case study results, our approach shows promise in terms of usefulness. In particular, our approach has a high recall (94.85%) and acceptable precision (67.38%) in identifying the ISO controls relevant to a security assessment project. Further, the execution times for training and classification are small. This suggests that our approach will scale to larger datasets.

7 Threats to Validity

The validity considerations most relevant to our work are construct and external validity, as we discuss below.

Construct Validity: Our evaluation metrics are scoped to the security controls for which there are at least five occurrences in the historical data. Below this threshold, applying ML is unlikely to be meaningful. Our evaluation examines whether ML is a suitable technique for our analytical purpose only when ML is applicable. Other techniques – not explored in this paper – are required for dealing with the security controls to which ML cannot be meaningfully applied.

External Validity: Generalizability is an important concern for any single case study, including the one in this paper. While the historical information we draw on for learning is aligned with commonly used ISO standards and is thus representative of a broader set of security assessment practices in industry, additional case studies are essential for examining whether our approach remains effective in other application contexts. In particular, the nature and source of security controls in other contexts and how accurately the pertinence of these controls can be determined through automation requires further investigation.

8 Conclusion

In this paper, we proposed an approach based on machine learning for assisting analysts with the task of deciding what security controls are relevant to a given system and context. This task is an important prerequisite for the proper elaboration of security requirements in the early stages of development. We evaluated our approach using real security assessment data from the banking domain. The results suggest that our approach provides effective decision support for security controls whose application is not too rare in the existing data. For these controls, our approach yielded an average recall of ≈95% and average precision of ≈67%. As far as we are aware, we are the first to have applied machine learning for supporting the selection of security controls.

In the future, we would like to study whether complementary techniques such as case-based reasoning can be utilized for handling security controls with too few occurrences in the existing data. Another important future direction is to provide decision support for the identification of threats and vulnerabilities. Broadening our approach to cover these aspects requires going beyond the structured assessment information that is stored according to a pre-defined schema. In particular, we will need to additionally consider and extract security-related information from textual development artifacts, e.g., system and asset descriptions. Finally, we would like to conduct a qualitative evaluation of the interpretable machine-learning models in our current case study, and further perform new case studies to investigate the usefulness of our approach in domains other than banking.

Acknowledgments. Financial support for this work was provided by the Alphonse Weicker Foundation.

References

1. Batista, G.E., et al.: A study of the behavior of several methods for balancingmachine learning training data. ACM SIGKDD Explor. Newslett. **6**, 20–29 (2004)
2. Breiman, L., et al.: Classification and Regression Trees. Wadsworth International Group, Belmont (1984)
3. Casamayor, A., et al.: Identification of non-functional requirements in textual specifications: a semi-supervised learning approach. IST **52**(4), 436–445 (2010)
4. Chawla, N.V., et al.: SMOTE: synthetic minority over-sampling technique. JAIR **16**, 321–357 (2002)
5. Cohen, W.W.: Fast effective rule induction. In: ICML 1995 (1995)
6. Dalpiaz, F., Paja, E., Giorgini, P.: Security Requirements Engineering: Designing Secure Socio-Technical Systems. MIT Press, Cambridge (2016)
7. Dowd, M., et al.: The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities. Pearson Education, London (2006)
8. Elkan, C.: The foundations of cost-sensitive learning. In: IJCAI 2001 (2001)
9. Frank, E., Witten, I.H.: Generating accurate rule sets without global optimization. In: ICML 1998 (1998)
10. Furnell, S.: End-user security culture: a lesson that will never be learnt? Comput. Fraud Secur. **2008**, 6–9 (2008)
11. Hall, M., et al.: The WEKA data mining software: an update. ACM SIGKDD Explor. Newslett. **11**, 10–18 (2009)
12. Ionita, D., Wieringa, R.: Web-based collaborative security requirements elicitation. In: REFSQ Workshops (2016)
13. ISO/IEC 27002:2005 Code of Practice for Information Security Controls. ISO Standard (2005)
14. ISO/IEC 27000:2018 Information Security Management Systems. ISO Standard (2018)
15. John, G.H., Langley, P.: Estimating continuous distributions in Bayesian classifiers. In: UAI 1995 (1995)
16. Jufri, M.T., et al.: Risk-assessment based academic information system security policy using octave allegro and ISO 27002. In: ICIC 2017 (2017)
17. Kurtanović, Z., Maalej, W.: Mining user rationale from software reviews. In: RE 2017 (2017)
18. le Cessie, S., van Houwelingen, J.C.: Ridge estimators in logistic regression. Appl. Stat. **41**(1), 191–201 (1992)
19. Li, T.: Identifying security requirements based on linguistic analysis and machine learning. In: APSEC 2017 (2017)
20. Meier, J.D., et al.: Improving web application security: threats and countermeasures. Technical report, Microsoft (2012)
21. Mitchell, T.M.: Machine learning and data mining. Commun. ACM **42**(11), 30 (1999)
22. Myagmar, S., et al.: Threat modeling as a basis for security requirements. In: SREIS 2005 (2005)
23. Nasrabadi, N.M.: Pattern recognition and machine learning. J. Electron. Imaging **16**(4), 049901 (2007)

24. NIST Special Publication 800–30: Guide for Conducting Risk Assessments. NIST Standard (2012)
25. OSA: Open Security Architecture. http://www.opensecurityarchitecture.org. Accessed Sep 2018
26. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
27. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, Burlington (1993)
28. Rodeghero, P., et al.: Detecting user story information in developer-client conversations to generate extractive summaries. In: ICSE 2017 (2017)
29. Schmitt, C., Liggesmeyer, P.: A model for structuring and reusing security requirements sources and security requirements. In: REFSQ Workshops (2015)
30. Sihwi, S.W., et al.: An expert system for risk assessment of information system security based on ISO 27002. In: ICKEA 2016 (2016)
31. Sindre, G., Opdahl, A.L.: Eliciting security requirements with misuse cases. REJ **10**, 34–44 (2005)
32. Türpe, S.: The trouble with security requirements. In: RE 2017 (2017)