Noname manuscript No.
(will be inserted by the editor)

Using Machine Learning to Assist with the Selection of Security Controls During Security Assessment

Seifeddine Bettaieb · Seung Yeob Shin · Mehrdad Sabetzadeh · Lionel C. Briand · Grégory Nou · Michael Garceau

Received: date / Accepted: date

Abstract **[Context]** 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. 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. **[Objective]** In this article, we develop automated decision support for the identification of security controls that are relevant to a specific system in a particular context. **[Method and Results]** 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 $\approx 94\%$ and average precision of $\approx 63\%$. We further examine through a survey the perceptions of security analysts about the usefulness of the classification models derived from historical data. **[Conclusions]** 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 –

S. Bettaieb · S. Y. Shin
SnT Centre, University of Luxembourg, Luxembourg
E-mail: {seifeddine.bettaieb,seungyeob.shin}@uni.lu

M. Sabetzadeh · L. C. Briand
SnT Centre, University of Luxembourg, Luxembourg
University of Ottawa, Canada
E-mail: {msabetza,lbriand}@uottawa.ca

M. Garceau · A. Meyers
BGL BNP Paribas, Luxembourg
E-mail: mgarceau@cipherquest.com, antoine.meyers@bnpparibas.com

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. Further, our survey results suggest that the generated classification models help provide a documented and explicit rationale for choosing the applicable security controls.

Keywords Security Requirements Engineering, Security Assessment, Automated Decision Support, 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 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 (ISO and IEC 2018), NIST SP 800 guidelines (NIST 2012), and OSA security patterns (OSA 2018). These controls define a wide range of technical and administrative measures for the avoidance, detection and mitigation of security risks (Furnell 2008). 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 aiming to avoid 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 article 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 (Dowd et al. 2006). 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 article 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 to develop 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 $\approx 94\%$ and precision of $\approx 63\%$. 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.

To gain insight into how useful our approach is in practice, we conduct a survey involving six security experts from our collaborating bank. The results of this survey suggest that the explicit classification models we derive from historical data are largely consistent with the implicit and, at the moment, undocumented reasoning performed by the experts. In this sense, the derived classification models not only help harmonize the decision making process about security controls but also provide a useful aid for training new staff who are yet to be familiarized with the bank's security assessment process.

This article is an extension of a previous conference paper (Bettaieb et al. 2019) published at the 25th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2019). The article offers important extensions over the previous conference paper by: (1) providing a more thorough discussion of background, related work, and our field study on security assessment, (2) improving the empirical evaluation of our approach by considering additional historical data and additional alternatives for configuring the learning process, and (3) examining the opinions of industry experts about the practical usefulness of our approach.

The rest of the article is organized as follows: Section 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, including our expert survey. Section 7 discusses threats to validity. Section 8 concludes the article.

2 Background and Related Work

This section presents background on industry standards related to security controls and machine learning algorithms used in this article. We further discuss and compare with different strands of related research in the area of security requirements engineering, security-control identification, and applications of machine learning.

2.1 Background

2.1.1 Information Security Standards

Information security standards, e.g., ISO 27000 (ISO and IEC 2018), ISO 31000 (ISO 2018), and NIST SP 800-30 (NIST 2012), provide a set of guidelines and best practices to help organizations build reliable, systematic processes for ensuring the secure handling of sensitive data. To mitigate the risks posed by the inevitable presence of security breaches, organizations typically tailor information security standards through specific frameworks and methodologies, e.g., Method for an Optimised aNAlysis of Risks (MONARC) (CASES 2018), Operationally Critical Threat, Asset and Vulnerability Evaluation (OCTAVE) (Caralli et al. 2007), MEthod for Harmonized Analysis of RIsk (MEHARI) (CLUSIF 2018), VECTOR matrix (Cyber Threat Institute 2019), and Control Objectives for Information and Related Technologies (COBIT) (ISACA 2018). These frameworks and methodologies are primarily meant at helping organizations identify and address security risks in a precise but manual manner. Automated decision support for managing security risks remains an under-explored topic.

Our collaborating partner has its IT security practices grounded in the ISO 27000 family of information security standards (ISO and IEC 2018). 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 article. ISO 27001 specifies a set of requirements for developing and maintaining an Information Security Management System (ISMS). The standard further envisages requirements for the assessment and control of the security risks posed by security breaches in IT applications. 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 performing a security risk assessment, one has to identify the security controls that are relevant to the enforcement of the ISMS requirements. As noted earlier, performing this task without automated assistance is both tedious and prone to errors. Our work in this article takes aim at providing suitable automated support for the above task.

2.1.2 Machine Learning for Decision Support

ML-based techniques have been widely used in software engineering for developing decision support systems (Casamayor et al. 2010; Kurtanović and Maalej 2017; Rodeghero et al. 2017). In our context, we use ML-based techniques to recommend

ISO-specified security controls that are relevant to a given new IT project. To do so, we rely on *supervised* learning; the labeled data here is the historical data from past risk assessments. An IT application is typically associated with multiple assets, e.g., networks, which pose different potential risks, e.g., untrusted networks. Our decision support, therefore, needs to consider the possibility of multiple security controls being recommended for a given application. Ascribing security controls to a given application should thus be viewed as a multilabel classification problem (Zhang and Zhou 2014).

Multilabel classification is performed by either (1) *problem transformation*, which is to convert multilabel datasets into binary or multiclass datasets aiming to fit the input dataset to standard binary or multiclass ML algorithms, or (2) *algorithm adaptation*, which is to modify binary or multiclass ML algorithms to directly support multilabel datasets (Zhang and Zhou 2014). In our work, we employ the former, i.e., problem transformation, since this technique has been successfully used in numerous applications (Boutell et al. 2004; Park and Fürnkranz 2007; Read et al. 2009; Tsoumakas and Vlahavas 2007) and is further independent from individual binary and multiclass ML algorithms thus allowing us to compare several ML algorithms.

Problem transformation techniques can be categorized into *binary relevance* (Boutell et al. 2004), *classifier chain* (Read et al. 2009), *calibrated label ranking* (Park and Fürnkranz 2007), and *random k-labelsets* (Tsoumakas and Vlahavas 2007). In our work, individual security controls are selected based on contextual factors and independently of other security controls. We thus elect to base our decision support system on the binary relevance method which decomposes a multilabel classification problem into a set of independent binary classification problems, with each binary classification problem corresponding to one specific label, in our context, one security control.

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 security controls is much more prevalent than their presence across the projects. This imbalance is caused by the relatively infrequent use of several security controls. When a class – in our context, a particular security control being applicable – is rare, ML classification models have a tendency to predict the more prevalent classes (Batista et al. 2004). In our context, this means that, unless steps are taken to counter imbalance for rarely used security controls, any classification model that we build may invariably find the rare security controls inapplicable. To tackle imbalance, we examine three commonly used methods. These are: (1) Synthetic Minority Over-sampling TEchnique (SMOTE) (Chawla et al. 2002), (2) SMOTE combined with Wilson's Edited Nearest Neighbor (ENN) (Wilson 1972), and (3) Cost-Sensitive Learning (CSL) (Elkan 2001).

SMOTE (Chawla et al. 2002) modifies a class distribution directly through oversampling. The main idea behind SMOTE is synthesizing new artificial minority samples by interpolating between existing minority samples. Compared to a baseline oversampling technique which creates samples by replication, SMOTE provides better performance in dealing with the imbalance problem since SMOTE causes a classifier to build larger decision regions containing nearby minority samples than regions determined by using the replication method (Chawla et al. 2002).

ENN (Wilson 1972) is an undersampling technique which creates a subset of the original dataset by eliminating samples. ENN removes any sample whose label is

inconsistent with the labels of at least two of its three nearest neighbors. ENN is often used in combination with an oversampling technique such as SMOTE to mitigate overfitting (Batista et al. 2004). This avoids synthetic minority samples from too deeply invading the majority class regions. We apply ENN to an over-sampled dataset produced by SMOTE as a data cleaning method.

CSL (Elkan 2001) accounts for the cost of misclassification during the construction of a decision model. When using CSL to deal with imbalance, the cost of misclassifying a minority sample needs to be higher than the cost of misclassifying a majority sample. This is because minority (positive) samples are usually more important to correctly identify than majority samples when a dataset is imbalanced.

2.2 Related Work

2.2.1 Security Requirements Engineering

Security requirements have been widely studied for IT applications, e.g., (Dalpiaz et al. 2016; Haley et al. 2008; Ionita and Wieringa 2016; Jufri et al. 2017; Li 2017; Meier et al. 2003; Myagmar et al. 2005; Schmitt and Liggesmeyer 2015; Sihwi et al. 2016; Sindre and Opdahl 2005; Türpe 2017; Yu et al. 2015). 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 (Meier et al. 2003). These techniques have been used and improved by many corporations over the years (Myagmar et al. 2005). 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 (2015) propose a model for structuring security knowledge as a way to improve the efficiency of specifying and analyzing security requirements. Sindre and Opdahl (2005) develop a systematic approach for security requirements elicitation based on use cases, with a focus on reusable methodological guidelines. Haley et al. (2008) present a framework for eliciting and analyzing security requirements. The framework is based on building a system context and representing security requirements as constraints. A problem-oriented notation is used to specify a system context, which is then validated by ensuring that the security constraints remain satisfied. Yu et al. (2015) explore the automated analysis of security requirements by using a uniform representation of risks and arguments. Through automated checking of formal arguments, they identify relevant risks and mitigations from publicly available security catalogs. 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 while conforming to specific information security standards. We further demonstrate

the effectiveness of our approach through empirical means by applying the approach to an industrial case study.

2.2.2 Security-Control Identification

Decision support for selecting security controls has been previously studied in the field of operations research. Almeida and Respício (2018) propose a framework that helps organizations optimize a set of relevant security controls to mitigate the security vulnerabilities identified. Their optimization aims to achieve the following two objectives: minimizing the cost of implementing security controls and ensuring that all the identified vulnerabilities are mitigated by at least one control. The framework casts this optimization problem into integer programming, focusing on security controls defined in the ISO 27002 standard.

Yevseyeva et al. (2015) develop an uncertainty-aware approach which selects an optimal subset of security controls using an integer programming. The main objective of their approach is diversifying security controls in order to mitigate a-priori unknown risks. More recently, Yevseyeva et al. (2016) cast the security-control selection problem as a two-stage decision-making process: defining a security budget, and then distributing the defined budget across security controls. The two-stage decision-making model is solved by using a quadratic programming technique while balancing risk and return when distributing the budget.

Kiesling et al. (2016) propose a model-driven security-control identification approach which combines models of security knowledge, IT applications and threats, a discrete-event simulation of attacks, and a multi-objective genetic algorithm that aims to identify equally viable subsets of security controls. This approach attempts to minimize six objectives: the implementation cost of security controls, the number of undetected attacks, the number of attacks achieving their goals, and the impact of attacks on confidentiality, integrity and availability.

In contrast to our work, none of the above work strands attempt to leverage historical data from past risk assessments. Further, we complement the existing literature by providing an explicit data model for the control-related concepts envisaged by the ISO 27001 and ISO 27002 standards.

2.2.3 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 (Rodeghero et al. 2017), identifying non-functional requirements (Casamayor et al. 2010), and requirements classification (Kurtanović and Maalej 2017). Rodeghero et al. (2017) present an approach for automatically detecting user-story information from recorded conversations between customers and developers. Their approach builds on an ML-based classifier for detecting user-story information. Casamayor et al. (2010) propose a semi-supervised learning technique for identifying non-functional requirements. The learning technique relies on categorized requirements and certain textual properties. In addition, their method exploits feedback from users to improve performance. Kurtanović and Maalej (2017) discuss how accurately they can automatically classify requirements using supervised

machine learning techniques with text, metadata, sentiment, and syntactic features. 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.

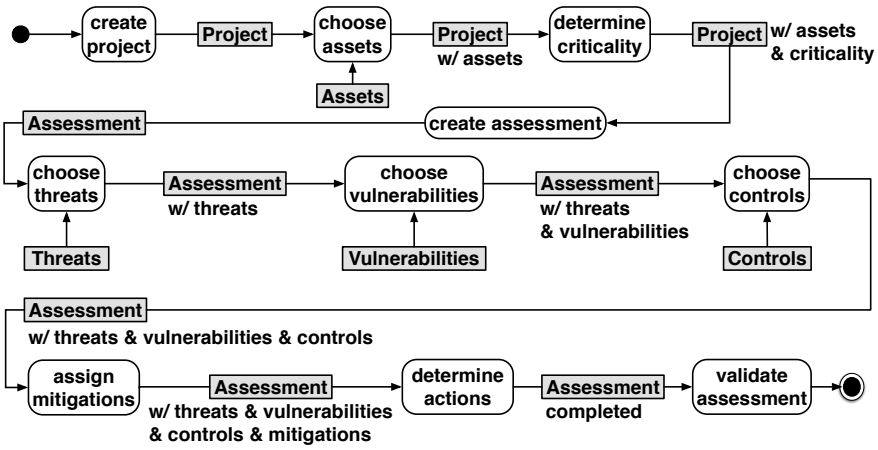
The security assessment process at our partner is a customized procedure shaped around the guidelines of the ISO 27000 standards (ISO and IEC 2018). A central goal of this process is to derive, for a given system, a set of ISO-specified controls that need to be elaborated further into security requirements.

In Figure 1(a), we show an overview of the security assessment process gleaned from our field study, and in Figure 1(b) – a (simplified) version of the underlying 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 Figure 1(b) is populated incrementally as the assessment workflow unfolds, with each step of the workflow adding new information.

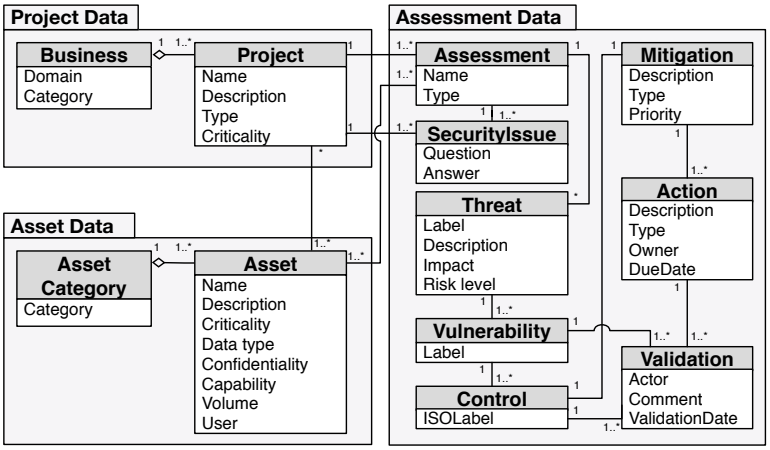
3.1 Security Assessment Process

As shown in Figure 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 (ISO and IEC 2005). 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



(a) Security assessment process



(b) Data model

Fig. 1 Main outcomes of our field study: (a) security assessment process and (b) data model.

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 article, 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 (ISO and IEC 2005). Threats are identified in the "choose threats" step of the process of Figure 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 (ISO and IEC 2005). An example vulnerability would be "oversight in defining access control rules to shared information". At our partner, vulnerabilities are identified using a pre-defined catalog with 154 entries. This catalog encompasses all the vulnerabilities known to the partner in its application domain.

After identifying the vulnerabilities for the project being assessed, in the "choose controls" step, the analysts select the appropriate security controls for the vulnerabilities. The source for the security controls at our partner is the ISO 27002 standard (ISO and IEC 2005). We thus refer to these controls as ISO controls. The catalog of ISO controls used by our partner is comprised of 134 entries.

Once the applicable ISO controls are identified, the analysts propose mitigations in the "assign mitigations" step. Essentially, a mitigation is a technical elaboration of an ISO control. For example, given an ISO control, say, "Protect data over an untrusted network", the corresponding mitigation could be "Use cryptographic mechanisms to protect data over an untrusted network". Next, in the "determine actions" step, actions can be specified for operationalizing the mitigations defined in the previous step. For instance, related to the above mitigation example, the analysts can suggest the following actions: (1) procure cryptography software, (2) provide training for cryptography software, and (3) raise staff awareness about the risks posed by untrusted networks. Finally, in the "validate assessment" step, the assessment is validated by both a chief information security officer and the involved business owner.

3.2 Data Model

Figure 1(b) shows a (simplified) version of the data model that underlies the process of Figure 1(a). Table 1 describes each of the data attributes in the data model. As seen from Figure 1(b), the data model is composed of three packages: "Project Data", "Asset Data", and "Assessment Data". The "Project Data" package contains the entities

Table 1 Glossary for the data model of Figure 1(b). Note that the "name" and "description" attributes of the data entities have been excluded from the glossary since the definitions were trivial.

Data entity	Attribute	Description
Project	Type	The type of a project (total of 3 types: usual business, large scale, and integration)
	Criticality	The criticality of a project (total of 3 criticality levels: very critical, critical, and non-critical)
Business	Domain	The area of business under which a project falls (total of 48 domains, e.g., web banking and wealth management)
	Category	The category for a group of business domains (total of 4 categories: legal/regulatory, information technology, management, and banking)
Assessment	Type	The type of an assessment (total of 5 types: full, light, consultancy, template, and derogation)
Security Issue	Question	A security question (total of 12 questions). An example question is: "What is the project's level of exposure to external attacks?"
	Answer	An answer provided by the analysts to the question (three-point scale: low, significant, and very high)
Asset	Criticality	The criticality of an asset (total of 2 criticality levels: critical and non-critical)
	Data type	The data type of an asset (total of 4 types: (T1) personal, identifiable, and secret; (T2) personal, not identifiable, and secret; (T3) personal, not identifiable, and not secret; and (T4) others)
	Confidentiality	The confidentiality level of an asset (total of 4 levels: public, restricted, confidential, and secret)
	Capability	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
	Volume	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
	User	The number of users who can access an asset
Asset Category	Category	The category for an asset (total of nine categories: application, OS, enterprise application suite (EAP), web application, mobile application, middleware, internal tool, database, and data)
Threat	Label	The label of a threat
	Impact	The impact types of a threat (total of 4 types of impact: confidentiality, integrity, availability, and traceability)
	Risk level	Estimated risk of each threat on a scale of 1-8 (negligible to extremely high)
Vulnerability	Label	The label of a vulnerability
Control	ISOLabel	An ISO control label listed in the ISO 27002 standard
Mitigation	Type	The platform type where the mitigation takes place, e.g., web server and DBMS
	Priority	The priority of a mitigation (total of 3 priority level: high, medium, and low)
Action	Type	The type of an action (total of 3 types: business, process, and infrastructure)
	Owner	The owner to whom an action was assigned
	DueDate	The due date of an action
Validation	Actor	The assessor who performs a validation
	Comment	The comment left by the assessor
	ValidationDate	The date of a validation

that define a project to be assessed alongside its business domain/category. The "Asset Data" package contains the "Asset" entity which captures the required assets for developing a project. The "Assessment Data" package is composed of the data entities, e.g., "Threat", "Vulnerability", and "Control", that are being updated through the risk assessment process of Figure 1(a).

The automation approach that we are going to describe in the next section focuses on ISO control identification. In particular, we assume that the steps prior to the "choose controls" step of the process of Figure 1(a) have been already performed for a project, and proceed to automatically recommend relevant ISO controls for the project in question. As such, we do not use data from the "assign mitigations", "determine actions" and "validate assessment" steps of the process. In the remainder of this article, we discard the final three steps of the process of Figure 1(a) as well as the data associated with these three steps (mitigations, actions and validations), since these have no impact on our approach.

4 ML-based Recommendation System for Security Controls

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 ten years. From this database, we extract various attributes. Our attributes, which are based on the data model of Figure 1(b), are discussed next.

4.2 Machine Learning Features

We engineered our features for learning through a joint endeavor with IT security specialists. Specifically, as shown in Figure 2, we defined the features for learning from historical data based on the data model of Figure 1(b). Table 2 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 2, 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.

Table 2 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., legal/regulatory, information technology, management, and banking). (I) The feature provides an extra layer of abstraction for distinguishing different business domains.
Security answer (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.
Threat (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.

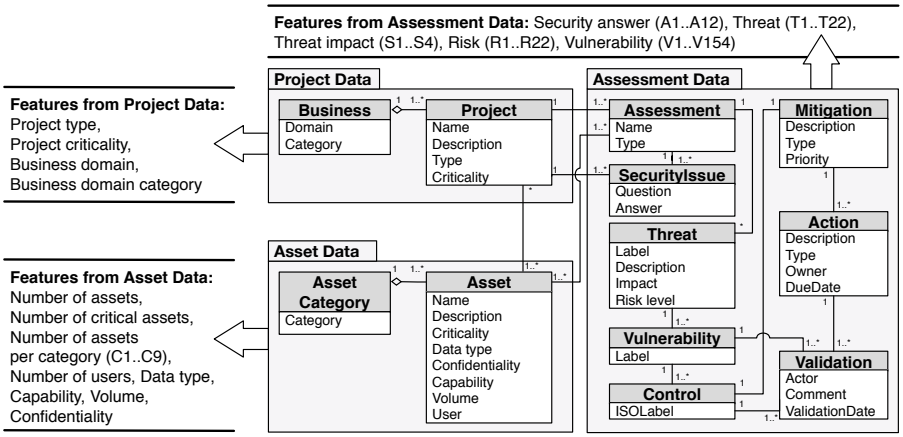


Fig. 2 Feature extraction from the data model in Figure 1(b).

4.3 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. Scoping our work to interpretable ML is important, because experts are unlikely to accept decisions for which they are not provided with 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 Section 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 (imbalance handling and classification): *What combination of imbalance handling technique and classification algorithm is the most accurate for recommending security controls?* In RQ1, we examine imbalance handling techniques and standard classification algorithms based on the existing best practices in the literature (Bishop 2007), 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 2.

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.
RQ4 (validation): *Do security analysts find the (interpretable) classification models generated by our approach useful?* Our approach is useful only if security analysts faced with real security risk assessment tasks can derive from the resulting (interpretable) classification model accurate justifications to support their decisions. RQ4 aims to assess the perceptions of security experts at our collaborating partner about the usefulness of the generated classification models as a decision aid.

5.2 Implementation

Our recommendation system is built using the Weka framework (Hall et al. 2009). Weka supports a broad spectrum of ML techniques. We ran our experiments on a computer equipped with an Intel i7 CPU with 16GB of memory.

5.3 Case Study Data

Our raw data is a database of 320 assessment projects conducted over a span of ten years, from 2009 until present. Of these assessment projects, we excluded 65 because they either were not carried through to completion, were built for testing and training purposes, or were using new experimental security control catalogs. This leaves us with 255 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 255 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 2 and 83 ISO controls across 255 assessment projects.

5.4 Experimental Setup

To answer the RQs in Section 5.1, we performed three experiments, EXPI, EXPII and EXPIII, and an expert interview survey, EIS, as described below.

EXPI. This experiment answers RQ1. We select the following interpretable ML algorithms as candidates for building our recommendation system: Naive Bayes (John and Langley 1995), Logistic Regression (le Cessie and van Houwelingen 1992), J48 (Quinlan 1993), CART (Breiman et al. 1984), JRip (Cohen 1995), and PART (Frank and Witten 1998). EXPI compares the accuracy of these six alternatives using the features of Table 2.

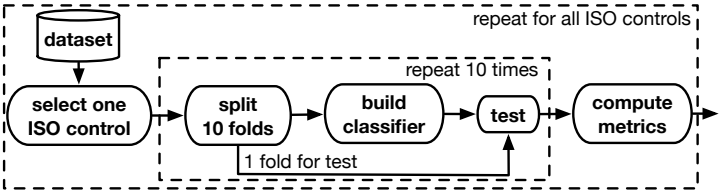


Fig. 3 10-fold validation for all ISO-control classifiers.

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 Section 2.1.2. As noted in this earlier section, we consider three techniques for handling imbalance: SMOTE, CSL, and a combination of SMOTE and ENN, denoted SMOTE+ENN. We recall from Section 2.1.2 that: 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; and SMOTE+ENN oversamples the dataset then discards the inconsistent samples. With regard to CSL, 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 14 to 1. We use this ratio for our experimentation with CSL, i.e., we set the cost ratio of false negatives versus false positives to 14 to 1.

For HPO, we use a step-wise grid search algorithm (Mitchell 1999) 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 Figure 3. The "repeat 10 times" block in the figure applies standard 10-fold cross validation (Bishop 2007) 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 Section 5.6.

EXPII. This experiment answers RQ2. We evaluate the importance of the features in Table 2 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

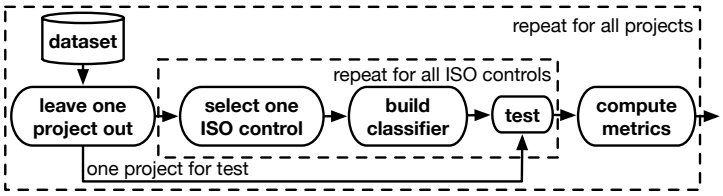


Fig. 4 Leave-one-out validation for all projects.

discuss in Section 5.6. 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 Figure 4. 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 the EXPIII accuracy metrics described in Section 5.6.

EIS. This expert interview survey addresses RQ4. We conducted an interview survey with risk assessors from the banking domain in order to assess the practitioners' perceptions about the usefulness of interpretable ML in the context of security-control identification. The detailed procedure for our survey is described in Section 5.5.

5.5 Expert Interview Survey

After developing our ML-based approach for security-control identification, we organized a series of interviews to collect feedback from the team of risk assessors at our collaborating partner. To this end, we held a separate interview with each of the six members of this team. They had at least three years of experience conducting security risk assessments, with more than 50 years of collective experience on the subject among the members. All interviewees possessed in-depth knowledge of the ISO 27001 and 27002 standards.

To conduct the interviews, we randomly selected seven ISO controls from the total of 83 covered in our ML experimentation. We then took the classification model generated for each of these seven controls by the best-performing ML configuration, i.e., J48 with CSL and the optimal hyper-parameter values as described in Section 6.1. The models yielded by J48 are decision trees. We created a visual representation of these trees, shown in Figure 5, and used the visual representation as the basis for the interviews.

Each decision tree in Figure 5 is targeted at a specific ISO control. For example, ISO 11.5.1 is a security control that concerns secure log-on procedures for systems and applications. The respective decision tree for identifying ISO 11.5.1 is shown at the

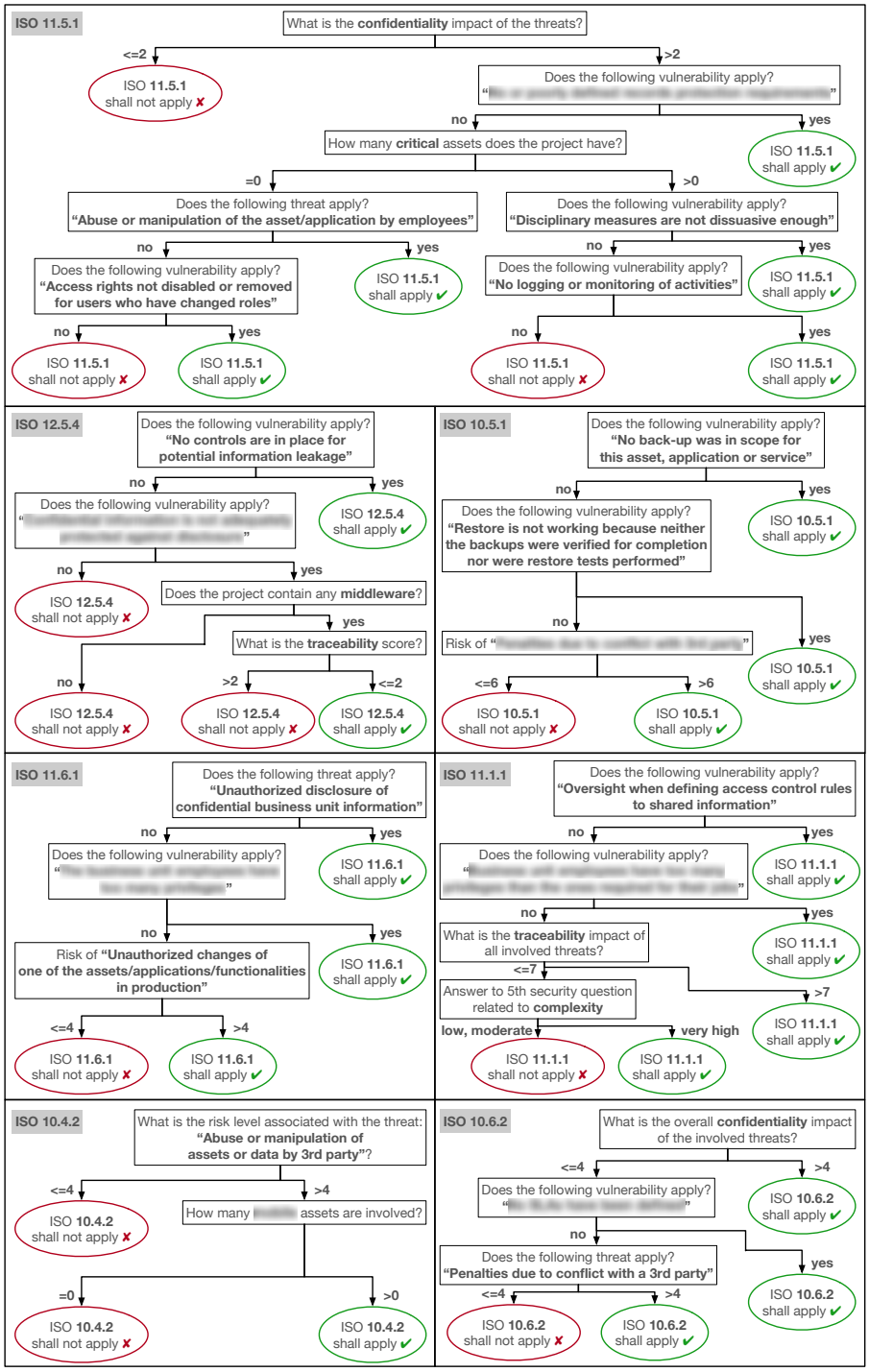


Fig. 5 The decision trees (produced by J48) used for answering RQ4. One node is blurred out of each tree for confidentiality.

Statement 1. The presented decision tree is easy to understand.
□ Strongly Agree □ Agree □ Neutral □ Disagree □ Strongly Disagree

Statement 2. The decision tree helps me identify the main factors in deciding whether ISO X should apply.
□ Strongly Agree □ Agree □ Neutral □ Disagree □ Strongly Disagree □ I Don't Know

Statement 3. The presented decision tree is consistent with how I decide about the applicability of ISO X.
□ Strongly Agree □ Agree □ Neutral □ Disagree □ Strongly Disagree □ I Don't Know

Overall Statement. Decision trees help provide a documented method for choosing the applicable security controls.
□ Strongly Agree □ Agree □ Neutral □ Disagree □ Strongly Disagree □ I Don't Know

Fig. 6 Statements for assessing the usefulness of interpretable ML classification models for security-control identification. Statements 1, 2, 3 are asked per classification model per expert; the Overall Statement is asked only once from each expert at the end of the interview survey.

top of the figure. The decision tree contains seven decision factors regarding threats, vulnerabilities, and assets captured in the nodes of the tree, and decision conditions captured as edge predicates. The interpretation of a decision tree starts from the root decision factor, and then descends (deterministically) to the appropriate child node based on the evaluation of the associated edge predicates. Each leaf node represents a yes/no answer as to whether a specific ISO control should apply.

In our interview survey, each participant was asked to evaluate, one by one, the trees in Figure 5 and then rate, for each tree, Statements 1, 2, and 3 shown in Figure 6. Once a participant had evaluated all the seven trees, (s)he was asked to rate the Overall Statement in Figure 6. To mitigate confounding effects such as fatigue, the order in which to examine the trees was chosen randomly for each participant.

The statements in Figure 6 were designed by taking inspiration from Rogers' theory of innovation diffusion (Rogers 2003). This theory introduces five characteristics that influence the adoption of innovative solutions. These characteristics, all of which are based on practitioners' perceptions, are:

- *Complexity*, referring to the degree to which an innovation is perceived to be difficult to understand or use.
- *Compatibility*, referring to the degree to which an innovation is perceived as consistent with existing values, experiences, and needs of practitioners.
- *Trialability*, referring to the degree to which an innovation can be tried on a limited basis or adopted in increments.
- *Observability*, referring to the degree to which the results of an innovation are visible to others.
- *Relative advantage*, referring to the degree to which an innovation is perceived to be better than what is currently used.

Among these five characteristics, we focus on the first three, namely, *complexity*, *compatibility*, and *trialability*; we cannot tackle the last two characteristics yet since our approach has not yet been deployed in practice. Statement 1 in Figure 6 concerns *complexity*. In particular, this statement assesses the degree to which a generated classification model is understandable. Statements 2 and 3 concern *compatibility*. In particular, they assess the degree to which a generated classification model is

aligned with the cognitive processes that risk assessors follow for security-control identification. Finally, the Overall Statement concerns *trialability*. In particular, the statement examines how helpful the generated models are as an instrument for codifying reusable knowledge about security controls.

The risk assessors participating in our interview survey used a five-point Likert scale (Likert 1932) for expressing their opinions about the statements in Figure 6. The possible options were: "Strongly Agree", "Agree", "Neutral", "Disagree", and "Strongly Disagree". Besides, the risk assessors were allowed to choose "I Don't Know" for all but Statement 1, as shown in Figure 6. Providing an option to opt out of rating a statement was meant at ensuring that the participants would choose the most accurate option rather than being forced to choose an option even when they could not form an opinion (Kitchenham and Pfleeger 2002).

All six interviews were conducted by the first author. To avoid fatigue, the maximum duration of an interview was limited to two hours. When an interview could not be finished within a two-hour slot, it was stopped and continued the next day the interviewee was available. In each interview, we first introduced the statements in Figure 6. To ensure that the risk assessors had understood the statements correctly and to properly collect the rationale behind their answers, we asked the assessors to verbalize their reasoning. We were expressly forbidden from (voice) recording the interviews due to concerns about security and confidentiality, but we were allowed to take notes.

5.6 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 (Quinlan 1986). This metric, which is commonly used for ranking ML features, is a modification of the information gain metric aiming to reduce 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 Section 5.1 based on the results of our case study.

6.1 RQ1

Table 3 shows the results of EXPI, described in Section 5.4. Specifically, the table reports the average precision and recall – average P^c and R^c, defined in Section 5.6, 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 3 thus clearly suggest that J48, which yields an average recall of 94.51% and average precision of 60.42%, is the best choice among the ML classification algorithm considered. When J48 is applied alongside CSL with a cost ratio of 14 to 1 for false negatives versus false positives (see Section 5.4), the optimal hyper-parameters are as follows: *pruning confidence=0.02* and *minimal number of instances per leaf=9*.

Table 3 Comparison of the average precision and recall of different ML classification algorithms with optimized hyper-parameters.

Algorithm	CSL		SMOTE		SMOTE+ENN	
	P^c(avg.)	R^c(avg.)	P^c(avg.)	R^c(avg.)	P^c(avg.)	R^c(avg.)
J48	60.42	**94.51**	81.86	76.67	77.22	68.29
CART	46.72	89.13	80.56	61.35	79.78	61.01
JRip	64.33	89.93	73.22	83.17	72.49	80.03
PART	66.79	91.31	77.31	74.69	68.63	73.14
Logistic regression	61.62	43.28	62.15	63.57	69.69	57.72
Naive Bayes	34.89	56.91	18.62	61.93	16.93	66.07

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 94.51% and average precision of 60.42% 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 Section 5.4, we use gain ratio for estimating the importance of our features (Table 2). 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 2) as the percentage of the number of classifiers having some feature of F in their top five most important features. Table 4 shows the results. For example, most ($\approx 99\%$) 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 4 were consistent with their intuition about the most important factors in determining the relevance of ISO controls.

Table 4 Most important features for ISO-control classification.

Vulnerability	Risk	Threat	Threat impact	# assets per category	Security answer	# assets
98.79%	57.83%	40.97%	13.49%	12.05%	2.41%	1.19%

The answer to **RQ2** *is that* overall and in descending order of magnitude, vulnerabilities, risks, threats, threat impact, the number of assets per category, security answers, and the number of 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 7 summarizes through a boxplot the results of EXPIII, described in Section 5.4. Specifically, the boxplot shows the distributions of precision, P^p, and recall, R^p, as defined in Section 5.6. On average, our approach has a recall of 93.52% and precision of 63.12% when tasked with identifying the ISO controls relevant to a given

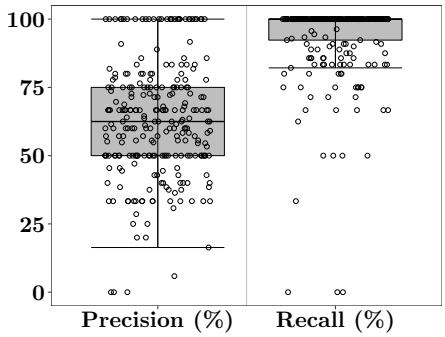


Fig. 7 Precision and recall distributions resulting from leave-one-out validation.

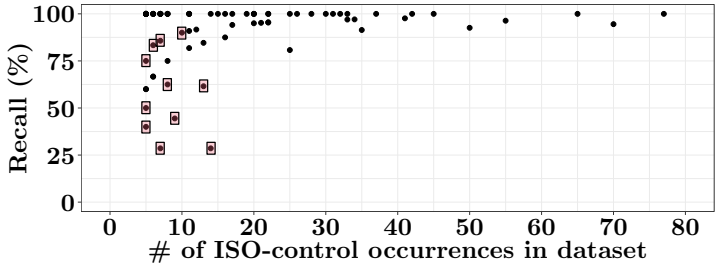


Fig. 8 Recall values for the ISO control classifiers; datapoints highlighted with □ are the culprits behind the low-recall projects (where $R^p < 75\%$).

project. The high recall suggests that the analysts can focus most of their attention on the recommended ISO controls, since the recommendations most likely contain all the relevant controls. The precision is reasonable too: On average, our approach recommends 8.9 ISO controls – both true and false positives – for a project. Of these, one can expect an average of 5.6 recommendations to be correct and 3.3 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.

From Figure 7, we further observe that the recall (R^p) for 17 out of the total of 255 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 of the ISO controls relevant to these projects have low prevalence in the dataset. In Figure 8, we plot the recall of each ISO-control classifier (R^c) against the prevalence of the respective ISO control in the dataset. The 11 datapoints encircled by □ represent the ISO controls that bring about low recall in the above-mentioned 17 projects. A complementary insight from Figure 8 is that recall is highly stable for those ISO controls that occur at least 15 times 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 242 seconds in total; this gives an average training time of 2.92 seconds per ISO control.

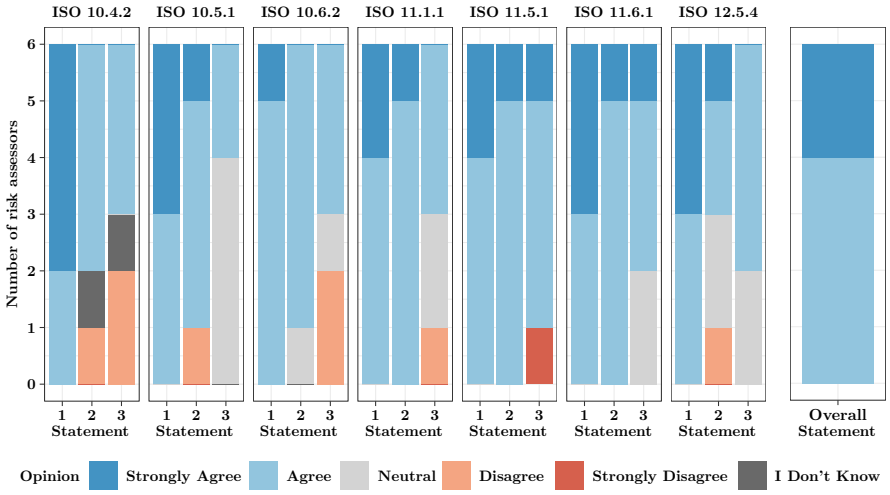


Fig. 9 Barchart representation of the interview survey results for the decision trees of Figure 5; the rightmost barchart aggregates the ratings for the Overall Statement.

With the classifiers built, issuing recommendations for a given project takes an average of 1.95 seconds. 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 (93.52%) and acceptable precision (63.12%) 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.

6.4 RQ4

Figure 9 shows the results for the individual decision trees considered in our interview survey as well as the overall feedback. These results were obtained by following the procedure described in Section 5.5. Every set of three adjacent bars represents the ratings collected for Statements 1, 2, and 3 (Figure 6) over one tree. The rightmost barchart in Figure 9 reports the results for the Overall Statement (Figure 6).

To facilitate our discussion of the results, we combine the participants' responses across the seven decision trees for each statement of Figure 6. The combined results – captured as a heatmap (Grinstein et al. 2001) – are depicted in Figure 10. Each value on this heatmap corresponds to the frequency of a certain rating for a given statement. We recall that, in total, we had six participants and seven decision trees. Hence, the heatmap has 42 datapoints (Likert-scale ratings) for each of the Statements 1, 2, and 3, and six datapoints for the Overall Statement. We further note that the "I Don't Know" response does not apply to Statement 1 as marked on the heatmap.

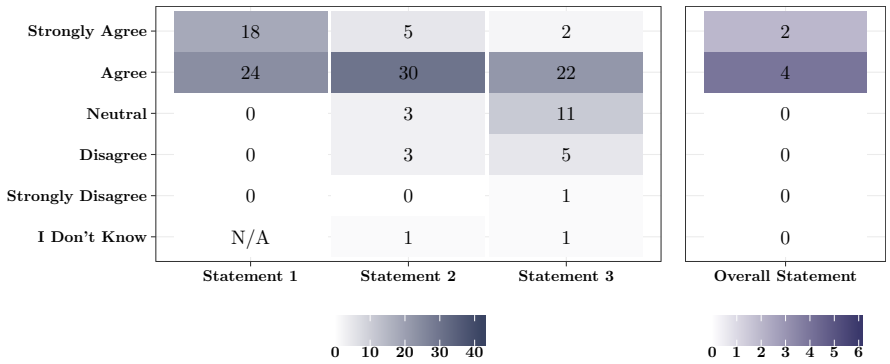


Fig. 10 Heatmap representation of the interview survey results.

We observe from the results of Statement 1 that all participants felt they were able to understand the presented decision trees and expressed positive feedback, i.e., "Strongly Agree" or "Agree".

With regard to Statement 2, in 83.33% (35/42) of the cases, the participants strongly agreed or agreed that the trees were helpful for identifying the main factors for deciding whether or not a security control should apply. In 7.14% (3/42) of the cases, the participants were neutral; and in the remaining 7.14% (3/42) of the cases, they disagreed. With regard to Statement 3, the participants strongly agreed or agreed in 57.14% (24/42) of the cases that the decision trees were aligned with how they decided about the applicability of the security control in question. In 26.19% (11/42), the participants were neutral; and in the remaining 14.28% (6/42) of the cases, the participants disagreed or strongly disagreed.

An unsurprising but important conclusion from our results for Statements 2 and 3 – considering that both statements target the technical dimension of decision making about security controls – is that, like in most decision-making tasks, the experts exercise a certain degree of subjectivity. This is evidenced by the mix of positive ("Strongly Agree" and "Agree") and non-positive ("Neutral", "Disagree", or "Strongly Disagree") ratings for these two statements, as shown in Figure 9. Overall, the results for Statements 2 and 3 are tilted to the positive side. Particularly, from Figure 10, we observe that out of the total of 84 ratings for Statements 2 and 3, only 23 (27.38%) are non-positive. We compute the average of the participants' responses by quantifying the agreement scale from 0 for "Strongly Disagree" to 4 for "Strongly Agree" and excluding the "I Don't Know" responses. This gives us, on average, 2.90 (between "Neutral" and "Agree") for Statement 2 and 2.46 (between "Neutral" and "Agree") for Statement 3.

Using the rationale and qualitative feedback that the participants provided throughout the interviews, we examined all the 23 non-positive ratings for Statements 2 and 3 in order to identify the issue(s) that prompted these ratings. We distinguish five reasons for the non-positive ratings. These reasons are shown in Table 5 alongside the number of ratings (out of 23) falling under each.

Table 5 Reasons why the participants answered with non-positive ratings ("Neutral", "Disagree", or "Strongly Disagree") alongside the frequency of these ratings.

Reason	Description	Frequency
Missing factor	An important factor does not appear in the decision tree.	5
Orthogonal factor	Some factor that appears in the decision tree is not used by the expert in his/her decision making process.	8
Numeric threshold mismatch	A numeric threshold in an edge condition, e.g., 2 in "<= 2", is different from what the expert expected.	5
Conceptual mismatch	The expert was unable to reconcile the decision tree with his/her way of reasoning.	3
Vague control	The security control in question is vague to the expert, as a result of which the expert is unable to commit to a clear decision process.	2

The *Missing factor* category explains five of the non-positive ratings; in two cases the participants indicated that the missing factor is tacit, i.e., not captured explicitly and known only from (verbal) interaction with clients. In the three remaining cases, the participants believed that the missing factor would be extractable from the unstructured text, e.g., project and asset descriptions, that is stored in the database.

The *Orthogonal factor* category explains eight non-positive ratings. In four cases, the participants felt that the non-relevant factor should be replaced with a factor that is present in the assessment database; in one case, the replacement was deemed absent from the assessment database but retrievable from a different source (database). As for the remaining three cases, the participants only noted the non-relevance of a factor but did not suggest a clear resolution (e.g., discarding the factor altogether or replacing it).

The *Numeric threshold mismatch* category explains five non-positive ratings. In all cases, the participants preferred a different value for some edge predicate in the decision tree. The *Conceptual mismatch* category explains three non-positive ratings; in all cases the participants felt that the decision process they had internalized could not be expressed with a decision tree. The *Vague control* category explains two ratings; in both cases the participants felt the security control in question had not been defined clearly enough. Since the definition of the security controls is outside our control, it is sensible to treat the two ratings under *Vague control* as "I Don't Know" answers. Nevertheless, no suggestion to this effect was made to the participants in order to avoid interference with the ratings.

In terms of opportunities for improvement, the main finding from our analysis of non-positive ratings is that the unstructured textual data in the assessment database may hold cues for building better classification models. Similarly, the situations where decision trees turned out not to match the experts' way of thinking deserve additional investigation. Pursuing these directions is left for future work. Also, given the subjectivity seen in the expert responses, it would be worthwhile to develop a structured negotiation process for resolving disagreements about the pertinence of security controls. Having such a process should help reduce inconsistencies and get better, consistent decision procedures. Decision trees can be a useful vehicle for

this purpose by making inconsistencies explicit and helping establish a common understanding and practice.

Finally, as seen from Figures 9 and 10, the responses to the Overall Statement (Figure 6) were unanimously positive, indicating that the participants saw benefit in decision trees as an instrument for documenting the decision-making process for security controls. Two interesting remarks made to this end by the participants are as follows: *"The [decision tree] models could help standardize the selection of security controls"* and *"Such trees could provide a sneak peek to business owners to explain the logic behind why a control is being applied."* We observe that the unanimously positive results for the Overall Statement are in spite of Statements 2 and 3 having received some non-positive responses. We discussed this discrepancy with the participants; they confirmed their ratings of the Overall Statement, arguing that, despite the occasional inaccuracies they saw in the trees, for a majority of the controls, they found the trees to be useful and consistent with their expertise.

The answer to **RQ4** *is that* the security risk assessors participating in our interview survey had a generally positive perception of decision trees as a tool for making explicit the logic behind security-control identification.

7 Threats to Validity

The validity considerations most relevant to our work are construct, conclusion, 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 article – are required for dealing with the security controls to which ML cannot be meaningfully applied.

Conclusion validity: Our quantitative evaluation is based on standard classification accuracy metrics, precision and recall. These metrics are only indicative of in-vivo usefulness. To mitigate this threat to conclusion validity and better examine whether our approach has the potential of being useful in practice, we performed a complementary qualitative study in the form of an interview survey. This survey directly assessed the perceptions of practicing engineers about our approach.

External validity: Generalizability is an important concern for any single case study, including the one in this article. 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. The same argument holds for the interview survey we conducted to examine the benefits of our approach. The current survey reports on the

reflections of a group of experts working at the same institution. Surveys with a broader scope and pool of respondents remain necessary for obtaining a more conclusive picture of the practical usefulness of our approach.

8 Conclusion

In this article, 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 $\approx 94\%$ and average precision of $\approx 63\%$. We further examined through a survey the opinions of six domain experts about the usefulness of our approach in practice. The survey results suggest that the classification models derived from historical data are largely in line with how experts reason about the applicability 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 perform new case studies to investigate the usefulness of our approach in domains other than banking.

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

References

Almeida L, Respício A (2018) Decision support for selecting information security controls. Journal of Decision Systems 27(sup1):173–180

Batista GEAPA, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter 6:20–29

Bettaieb S, Shin SY, Sabetzadeh M, Briand LC, Nou G, Garceau M (2019) Decision support for security-control identification using machine learning. In: Proceedings of the 25th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'19), pp 3–20

Bishop CM (2007) Pattern Recognition and Machine Learning. Information Science and Statistics, Springer

Boutell MR, Luo J, Shen X, Brown CM (2004) Learning multi-label scene classification. Pattern Recognition 37:1757–1771

Breiman L, Friedman J, Stone CJ, Olshen R (1984) Classification and Regression Trees. Wadsworth International Group

Caralli RA, Stevens JF, Young LR, Wilson WR (2007) Introducing OCTAVE Allegro: Improving the information security risk assessment process. Tech. Rep. CMU/SEI-2007-TR-012, SEI, Carnegie Mellon University

Casamayor A, Godoy D, Campo MR (2010) Identification of non-functional requirements in textual specifications: A semi-supervised learning approach. Information & Software Technology (IST'10) 52(4):436–445

CASES (2018) Method for an optimised analysis of risks by @CASES-LU. `https://www.monarc.lu`, accessed September 2018

Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research (JAIR'02) 16:321–357

CLUSIF (2018) Method for harmonized analysis of risk. `https://clusif.fr/mehari`, accessed September 2018

Cohen WW (1995) Fast effective rule induction. In: Proceedings of the 12th International Conference on Machine Learning (ICML'95), pp 115–123

Cyber Threat Institute (2019) Vector matrix - risk assessment methodology, security, impact. `http://www.riskvector.com`, accessed June 2019

Dalpiaz F, Paja E, Giorgini P (2016) Security Requirements Engineering: Designing Secure Socio-Technical Systems. MIT Press

Dowd M, McDonald J, Schuh J (2006) The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities. Pearson Education

Elkan C (2001) The foundations of cost-sensitive learning. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI'01), pp 973–978

Frank E, Witten IH (1998) Generating accurate rule sets without global optimization. In: Proceedings of the 15th International Conference on Machine Learning (ICML'98), pp 144–151

Furnell S (2008) End-user security culture: A lesson that will never be learnt? Computer Fraud & Security 2008:6–9

Grinstein G, Trutschl M, Cvek U (2001) High-dimensional visualizations. In: Proceedings of the Visual Data Mining Workshop (KDD'01), pp 120–134

Haley CB, Laney RC, Moffett JD, Nuseibeh B (2008) Security requirements engineering: A framework for representation and analysis. IEEE Transactions on Software Engineering (TSE'08) 34(1):133–153

Hall MA, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: An update. ACM SIGKDD Explorations Newsletter 11:10–18

Ionita D, Wieringa RJ (2016) Web-based collaborative security requirements elicitation. In: Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium, Research Method Track, and Poster Track co-located with the 22nd International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ Workshops'16), pp 3–6

ISACA (2018) Framework for it governance and control. http://www.isaca.org/Knowledge-Center/cobit/Pages/Overview.aspx, accessed June 2018
ISO (2018) ISO 31000 - Risk Management. ISO Standard
ISO and IEC (2005) ISO/IEC 27002:2005 Code of Practice for Information Security Controls. ISO Standard
ISO and IEC (2018) ISO/IEC 27000:2018 Information Security Management Systems. ISO Standard
John GH, Langley P (1995) Estimating continuous distributions in bayesian classifiers. In: Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence (UAI'95), pp 338–345
Jufri MT, Hendayun M, Suharto T (2017) Risk-assessment based academic information system security policy using OCTAVE Allegro and ISO 27002. In: Proceedings of the 2nd International Conference on Informatics and Computing (ICIC'17), pp 1–6
Kiesling E, Ekelhart A, Grill B, Strauss C, Stummer C (2016) Selecting security control portfolios: A multi-objective simulation-optimization approach. EURO Journal on Decision Processes 4:85–117
Kitchenham BA, Pfleeger SL (2002) Principles of survey research: Part 3: Constructing a survey instrument. ACM SIGSOFT Software Engineering Notes 27(2):20–24
Kurtanović Z, Maalej W (2017) Mining user rationale from software reviews. In: Proceedings of the 25th IEEE International Conference on Requirements Engineering (RE'17), pp 61–70
le Cessie S, van Houwelingen JC (1992) Ridge estimators in logistic regression. Applied Statistics 41(1):191–201
Li T (2017) Identifying security requirements based on linguistic analysis and machine learning. In: Proceedings of the 24th Asia-Pacific Software Engineering Conference (APSEC'17), pp 388–397
Likert R (1932) A technique for the measurement of attitudes. Archives of psychology 22(140):5–55
Meier J, Mackman A, Vasireddy S, Dunner M, Escamilla R, Murukan A (2003) Improving web application security: Threats and countermeasures. Tech. rep., Microsoft
Mitchell TM (1999) Machine learning and data mining. Communications of the ACM 42(11):30–36
Myagmar S, Lee AJ, Yurcik W (2005) Threat modeling as a basis for security requirements. In: Proceedings of the IEEE Symposium on Requirements Engineering for Information Security (SREIS'05), pp 1–8
NIST (2012) NIST Special Publication 800-30: Guide for Conducting Risk Assessments. NIST Standard
OSA (2018) Open security architecture. http://www.opensecurityarchitecture.org, accessed September 2018
Park S, Fürnkranz J (2007) Efficient pairwise classification. In: Proceedings of the 18th European Conference on Machine Learning (ECML'07), pp 658–665
Quinlan JR (1986) Induction of decision trees. Machine Learning 1(1):81–106
Quinlan JR (1993) C4.5: Programs for Machine Learning. Morgan Kaufmann

Read J, Pfahringer B, Holmes G, Frank E (2009) Classifier chains for multi-label classification. In: Proceedings of the 2009 Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD'09), pp 254–269

Rodeghero P, Jiang S, Armaly A, McMillan C (2017) Detecting user story information in developer-client conversations to generate extractive summaries. In: Proceedings of the 39th International Conference on Software Engineering (ICSE'17), pp 49–59

Rogers EM (2003) Diffusion of Innovations, 5th edn. Free Press

Schmitt C, Liggesmeyer P (2015) A model for structuring and reusing security requirements sources and security requirements. In: Joint Proceedings of REFSQ-2015 Workshops, Doctoral Symposium, Research Method Track, and Poster Track co-located with the 21st International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ Workshops'15), pp 34–43

Sihwi SW, Andriyanto F, Anggrainingsih R (2016) An expert system for risk assessment of information system security based on ISO 27002. In: Proceedings of the 2016 IEEE International Conference on Knowledge Engineering and Applications (ICKEA'16), pp 56–61

Sindre G, Opdahl AL (2005) Eliciting security requirements with misuse cases. Requirements Engineering 10:34–44

Tsoumakas G, Vlahavas IP (2007) Random *k*-labelsets: An ensemble method for multilabel classification. In: Proceedings of the 18th European Conference on Machine Learning (ECML'07), pp 406–417

Türpe S (2017) The trouble with security requirements. In: Proceedings of the 25th IEEE International Conference on Requirements Engineering (RE'17), pp 122–133

Wilson DL (1972) Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics 2(3):408–421

Yevseyeva I, Basto-Fernandes V, Emmerich M, van Moorsel A (2015) Selecting optimal subset of security controls. Procedia Computer Science 64:1035–1042

Yevseyeva I, Basto-Fernandes V, van Moorsel A, Janicke H, Emmerich M (2016) Two-stage security controls selection. Procedia Computer Science 100:971–978

Yu Y, Franqueira VN, Tun TT, Wieringa RJ, Nuseibeh B (2015) Automated analysis of security requirements through risk-based argumentation. Journal of Systems and Software (JSS'15) 106:102–116

Zhang M, Zhou Z (2014) A review on multi-label learning algorithms. IEEE Transactions on Knowledge and Data Engineering (TKDE'14) 26(8):1819–1837