Spring 3-2019

Matching Possible Mitigations to Cyber Threats: A Document-Driven Decision Support Systems Approach

Martha Wagner McNeil
Dakota State University

MATCHING POSSIBLE MITIGATIONS TO CYBER THREATS: A DOCUMENT-DRIVEN DECISION SUPPORT SYSTEMS APPROACH

A dissertation submitted to Dakota State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Information Systems

Spring 2019

By

Martha Wagner McNeil

Dissertation Committee:

Dr. Cherie Noteboom, Chair

Dr. Omar El-Gayar

Dr. Jun Liu

DISSERTATION APPROVAL FORM

This dissertation is approved as a credible and independent investigation by a candidate for the Doctor of Philosophy in Information Systems degree and is acceptable for meeting the dissertation requirements for this degree. Acceptance of this dissertation does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department or university.

Student Name: Martha Wagner McNeil

Dissertation Title: Matching Possible Mitigations to Cyber Threats

A Document-Driven Decision Support Systems Approach

Dissertation Chair/Co-Chair: _____ Date: _____

Dissertation Chair/Co-Chair: _____ Date: _____

Committee member: _____ Date: _____

Committee member: _____ Date: _____

Committee member: _____ Date: _____

Committee member: _____ Date: _____

ACKNOWLEDGMENT

This dissertation is dedicated to all those people who helped and supported me on this journey beginning with my first DSU class in 2014. Wow! Where did all that time go? I have always tried to thank each of you personally along the way as you have given your time, energy, and talent to support me. If I've succeeded in that goal, then what follows is simply a summary of thanks already given.

First, I'd like to thank my committee: Dr. Cherie Noteboom for her thoughtful support, guidance, and leadership as my committee chair as well as Dr. Omar El-Gayar and Dr. Jun Liu. Your guidance, support, and recommendations have been invaluable. Dr. El-Gayar was instrumental in encouraging me to submit to AMCIS last year. Dr. Liu was the first person I had the pleasure to know at DSU. As my advisor, you made me feel welcome and helped me keep my plan of study in order throughout my studies. I'd also like to thank all my DSU professors for lessons learned, insights shared, and making the whole online student concept work. As I look back, I can see how much I have grown from this experience thanks to all of you.

I'd also like to thank my colleagues at the Johns Hopkins University Applied Physics Laboratory for your encouragement and support in many forms, in particular those who provided peer review and thoughtful feedback to my papers. I'd especially like to thank Dr. Thomas Llanso who, in addition to lending his cybersecurity expertise, was a trailblazer for me, completing his doctoral degree at DSU in 2018.

Finally, I'd like to thank my family and friends. To my parents, Lillian and Marshall Wagner, thanks for always emphasizing the value of education. To my husband, Ron, thanks for supporting me in my pursuit of this long-held goal and in life in general. To my children, Ron and Amy, both college students, thanks for tiptoeing around when "Mom's doing her homework" and being willing to run out for take-out when necessary. To my sidekicks, Jean and Diane, best friends since high school, thanks for cheering me on. Thanks to my late furry friend, Charlie Brown, for always being beside me on the couch or on the floor at my feet as I tapped away on my laptop sometimes late into the night. I miss you. Finally, thanks to my new furry friend, Buddy, who constantly reminds me of the joys of youthful optimism, boundless energy, hound-like determination, and taking time to play.

ABSTRACT

Cyber systems are ubiquitous in all aspects of society. At the same time, breaches to cyber systems continue to be front-page news (Calfas, 2018; Equifax, 2017) and, despite more than a decade of heightened focus on cybersecurity, the threat continues to evolve and grow, costing globally up to $575 billion annually (Center for Strategic and International Studies, 2014; Gosler & Von Thaer, 2013; Microsoft, 2016; Verizon, 2017). To address possible impacts due to cyber threats, information system (IS) stakeholders must assess the risks they face. Following a risk assessment, the next step is to determine mitigations to counter the threats that pose unacceptably high risks. The literature contains a robust collection of studies on optimizing mitigation selections, but they universally assume that the starting list of appropriate mitigations for specific threats exists from which to down-select. In current practice, producing this starting list is largely a manual process and it is challenging because it requires detailed cybersecurity knowledge from highly decentralized sources, is often deeply technical in nature, and is primarily described in textual form, leading to dependence on human experts to interpret the knowledge for each specific context. At the same time cybersecurity experts remain in short supply relative to the demand, while the delta between supply and demand continues to grow (Center for Cyber Safety and Education, 2017; Kauflin, 2017; Libicki, Senty, & Pollak, 2014). Thus, an approach is needed to help cybersecurity experts (CSE) cut through the volume of available mitigations to select those which are potentially viable to offset specific threats.

This dissertation explores the application of machine learning and text retrieval techniques to automate matching of relevant mitigations to cyber threats, where both are expressed as unstructured or semi-structured English language text. Using the Design Science Research Methodology (Hevner & March, 2004; Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007), we consider a number of possible designs for the matcher, ultimately selecting a supervised machine learning approach that combines two techniques: support vector machine classification and latent semantic analysis. The selected approach demonstrates high recall for mitigation documents in the relevant class, bolstering confidence that potentially viable mitigations will not be overlooked. It also has a strong ability to discern documents in the non-relevant class, allowing approximately 97% of non-relevant mitigations to be excluded automatically, greatly reducing the CSE's workload over purely manual matching. A false

positive rate of up to 3% prevents totally automated mitigation selection and requires the CSE to reject a few false positives.

This research contributes to theory a method for automatically mapping mitigations to threats when both are expressed as English language text documents. This artifact represents a novel machine learning approach to threat-mitigation mapping. The research also contributes an instantiation of the artifact for demonstration and evaluation. From a practical perspective the artifact benefits all threat-informed cyber risk assessment approaches, whether formal or ad hoc, by aiding decision-making for cybersecurity experts whose job it is to mitigate the identified cyber threats. In addition, an automated approach makes mitigation selection more repeatable, facilitates knowledge reuse, extends the reach of cybersecurity experts, and is extensible to accommodate the continued evolution of both cyber threats and mitigations. Moreover, the selection of mitigations applicable to each threat can serve as inputs into multifactor analyses of alternatives, both automated and manual, thereby bridging the gap between cyber risk assessment and final mitigation selection.

DECLARATION

I hereby certify that this dissertation constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the dissertation describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

_____

Martha Wagner McNeil

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Background

Cyber systems[1] are ubiquitous in all aspects of society. At the same time, breaches to cyber systems continue to be front-page news (Calfas, 2018; Equifax, 2017) and, despite more than a decade of heightened focus on cybersecurity, the threat continues to evolve and grow, costing globally up to $575 billion annually (Center for Strategic and International Studies, 2014; Gosler & Von Thaer, 2013; Microsoft, 2016; Verizon, 2017). Symantec reported that "Cyber attackers revealed new levels of ambition in 2016, a year marked by extraordinary attacks, including multi-million-dollar virtual bank heists, overt attempts to disrupt the US electoral process by state-sponsored groups, and some of the biggest distributed denial of service (DDoS) attacks on record powered by a botnet of Internet of Things (IoT) devices" (Chandrasekar et al., 2017).

Regrettably, subsequent years have not been less exciting on the cybersecurity front (Symantec, 2019; Verizon, 2017). The Cisco 2018 Annual Cybersecurity Report identifies a number of recent changes in the threat landscape which continue to impact growth of the mitigation landscape. For example, self-propagating malware has moved to the network where it can spread very rapidly. In addition, adversaries continue to improve their abilities to evade existing security measures. Also, supply chain threats are on the rise and mitigation strategies against them are immature. Moreover, the years 2017 and 2018 saw a dramatic rise in ransomware along with rapid adoption of cloud and Internet of Things technologies for which mitigation strategies remain in the early stages (Cisco Systems, 2018).

To address possible impacts due to cyber threats, information system (IS) stakeholders must assess the risks they face. To that end, there is an extensive body of research and practice in the cyber risk assessment discipline. Many mature organizations employ formal risk

[1] Definitions of cyber terms are provided in Appendix A. In this paper, we use the term "mitigation" synonymously with "countermeasure" and "security control" to mean a tool or technique that may counter a cyber threat.

assessment methodologies in an attempt to achieve rigor, although ad hoc approaches are also used. We briefly discuss a selection of risk assessment methods in the Literature Review section below. These methods help stakeholders identify and prioritize cyber risks. After completing the risk assessment, in whatever form, stakeholders may have a better understanding of threats to their mission-critical IS assets.

Following risk assessment, the next step is to determine mitigations to counter the threats that pose unacceptably high risk, but this is challenging for several reasons. First, cyber threats and the means to counter them continue to proliferate (Center for Strategic and International Studies, 2014; Gosler & Von Thaer, 2013; Microsoft, 2016; Verizon, 2017). Consequently, the universe of documents describing cyber threats and potential mitigations is quite large and continually growing but there is currently no comprehensive source of threat-mitigation mappings. For example, NIST 800-53 (National Institute of Standards and Technology, 2017) is a well-known catalog of security control documents often referenced during the mitigation stage of cyber risk assessment. While it contains valuable knowledge, NIST 800-53 does not relate mitigations to specific threats, and thus, does not deter application of mitigations that over- or under-address the actual threats. On the other hand, the National Intelligence Cyber Threat Framework is a comprehensive threat framework, but it does not currently offer mitigation mappings (National Security Agency, 2018). The Common Attack Pattern Enumeration and Classification (CAPEC) is another threat framework (MITRE, 2017a). While CAPEC does contain a few representative mappings of mitigations to threats, these have been manually generated by cybersecurity experts, they are not all-inclusive, and mitigation selection is not the primary intent of the CAPEC framework. Second, over-applying mitigations wastes resources while under-applying or incorrectly applying mitigations, leaves residual risk and can result in a false sense of security. Third, to propose sensible mitigations one must acquire detailed cybersecurity knowledge. In current practice, knowledge about mitigations and threats is primarily contained in documents. This knowledge resides in numerous, highly decentralized sources, which are often deeply technical in nature and are primarily described in textual form, resulting in dependence on human experts to interpret the knowledge for the specific context.

To date, manual selection by cyber security experts continues to be the de facto method for identifying mitigations to cyber threats. Several issues arise from reliance solely on manual

selection by experts for cybersecurity mitigation decisions. First, cybersecurity experts continue to be in short supply relative to the demand, while the delta between supply and demand continues to grow (Center for Cyber Safety and Education, 2017; Kauflin, 2017; Libicki et al., 2014). In addition, the time-consuming nature of manual matching necessarily limits the number of sources of possible mitigations that can be consulted during any cyber risk assessment. Moreover, human variation in expertise and in sources consulted can lead to uneven and non-repeatable application of the available knowledge (Bolger & Wright, 1994; Hallberg, Bengtsson, Hallberg, Karlzén, & Sommestad, 2017; Holm, Sommestad, Ekstedt, & Honeth, 2014).

Problem Statement and Research Gap

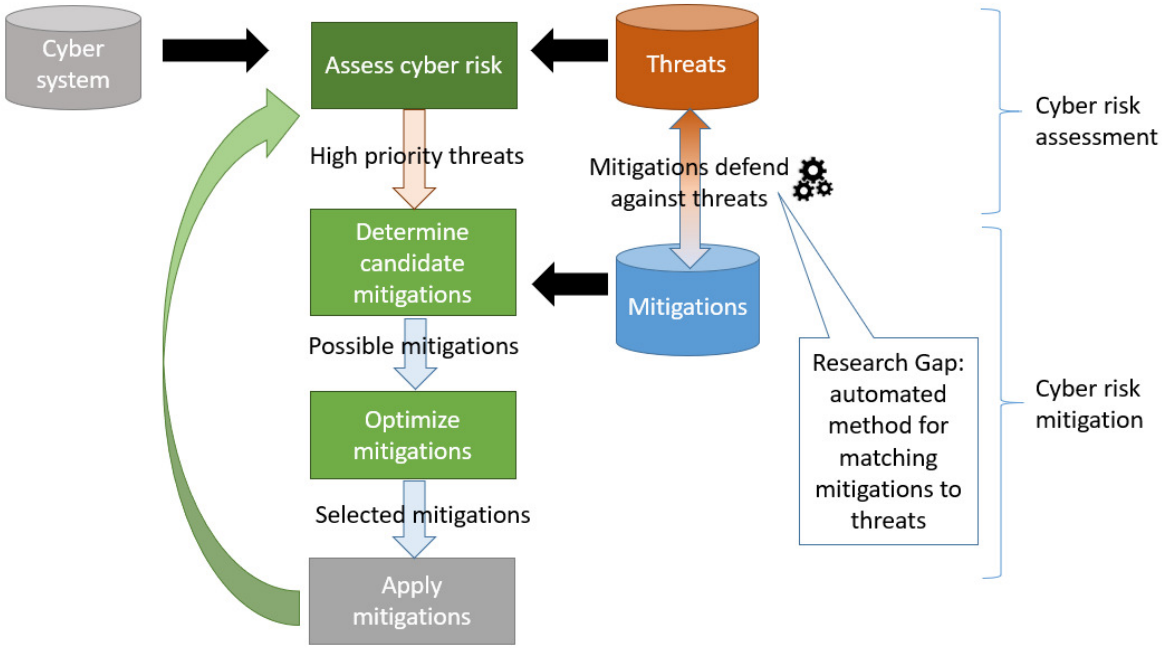

Figure 1. Research Gap

In this research, we set out to fill the research gap illustrated in Figure 1 by devising a method for matching mitigations to cyber threats expressed as English language text documents using machine learning and text retrieval techniques in support of cyber risk assessment. A fundamental goal of all cyber risk assessments, whether methodical or ad hoc, is to identify the threats faced in a particular environment with enough detail that specific, applicable mitigations can be determined, prioritized, and implemented. The first step in a cyber risk assessment is to

assess the cyber risk of the system by considering the threats against it. The output of this step is a list of **high priority threats** to be mitigated. The next step is to **determine candidate mitigations** to address the threats. As discussed above, this is difficult and has inherent issues of scalability, consistency, and repeatability because, absent automation to help match mitigations to threats, mitigation selection is primarily a manual process done by human experts using disparate textual sources. There are two dimensions to the mitigation selection problem. The first is a technical dimension, that is, for each threat, enumerating a set of **possible mitigations** that are capable of countering it. The second, **optimizing mitigations**, is an organizational dimension where budgetary and other organizational constraints necessitate winnowing the list of potentially applicable mitigations to those that are organizationally feasible. Our research focuses on the first dimension and is distinct from the second dimension.

The literature contains abundant research on the second dimension, herein referred to as mitigation optimization but also sometimes called trade space or analysis of alternatives. We briefly discuss a selection of mitigation optimization methods in the Literature Review section below. These approaches universally assume that the applicable set of potential mitigations for input into the mitigation optimization analysis has already been determined; however, **cyber risk assessment approaches stop short of providing this list of potential mitigations leaving a gap.** This dissertation addresses the gap by developing an automated method for matching mitigations to threats to obtain the initial set of potentially relevant mitigations. It is distinct from the mitigation optimization problem which commences after the initial list is made and forms the reservoir from which downstream risk-informed mitigation and mitigation optimization analyses can draw.

Objectives and Intended Contributions of the Project

The objective of this research project is to investigate the application of machine learning and text retrieval techniques for matching mitigations to cyber threats where both are expressed as unstructured or semi-structured English language text. We hypothesize that we can devise an automated or semi-automated method that has the potential to reduce workload for the CSE by recommending possible mitigations for a given threat when both are English language documents. We use Fedorowicz's definition of "document" as "a chunk of information, usually dealing with a relatively limited topic or subject area." (Fedorowicz, 1996)

Significant research exists both in threat-informed cyber risk assessment methodologies and mitigation optimization techniques. This research project addresses the gap between these two areas as described in the prior section. To that end, we investigate applicable text mining techniques from the machine learning and document-driven decision support systems (DSS) disciplines, assess to what degree these techniques apply to the domain of cyber threats and mitigations, look for domain-specific peculiarities, and recommend changes in how cybersecurity practitioners describe threats and mitigations to support the use of automated, document-matching schemes.

The primary contribution of this research to theory is the artifact, a novel machine learning method for matching mitigations documents to threats. We also provide instantiations of the method for demonstration. From a practical perspective, an automated approach to matching mitigations to threats benefits all threat-informed cyber risk assessment approaches by aiding decision-making and reducing workload for cybersecurity experts whose job it is to mitigate the identified cyber threats. Moreover, an automated approach can support development and maintenance of a knowledge base to make mitigation selection more repeatable, facilitate knowledge reuse, and extend the reach of cybersecurity experts. The approach will be extensible to accommodate the continued evolution of both cyber threats and mitigations. The selection of mitigations applicable to each of the threats can serve as inputs into mitigation optimization approaches thereby bridging the gap between cyber risk assessment and final mitigation selection.

The remainder of this paper is organized as follows. In Chapter 2, we discuss related literature in three domains: cyber risk assessment, mitigation optimization analysis, and document-driven decision supports systems. In Chapter 3, we discuss our research methodology, which is grounded in the principles of the Design Science Research Methodology (DSRM) (Hevner & March, 2004; Peffers et al., 2007), seeking tangible IS solutions to "wicked problems" (Vaishnavi & Kuechler, 2004). Per the DSRM, we identify objectives of a solution to our stated research problem, then we discuss the design and development of the solution artifact drawing from the knowledge base of applicable theory. We also discuss our approach to demonstrating the use of the artifact to solve a real-life problem and the evaluation criteria used to measure the success of the artifact. In Chapter 4, we discuss our results and assess the

validity of our research. Finally, in Chapter 5 we discuss conclusions and limitations of the present research and propose future work.

CHAPTER 2

LITERATURE REVIEW

A solution to the problem of automatically selecting mitigations pertinent to a given threat lies at the nexus of threat-informed cyber risk assessment methodologies and mitigation optimization analysis. We investigate literature in these two domains to ensure that our solution broadly supports existing methodologies. We also survey existing threat taxonomies and control catalogs to further delineate the gap. Despite an extensive search of the literature, we did not find any published research dealing specifically with automated matching of mitigations to cyber threats; hence, the DSS section of this literature review considers research that we consider analogous to our research problem.

Cyber Risk Assessment Methodologies

A number of threat-informed cyber risk assessment methodologies are described in the literature and in use today. They include AURUM (Fenz, Ekelhart, & Neubauer, 2011), BluGen (Llanso, McNeil, Pearson, & Moore, 2017), Crown Jewels Analysis and Threat Assessment and Remediation Analysis (CJA+TARA) (MITRE, 2015), Mission Information Risk Analysis (MIRA) (Llanso, Hamilton, & Silberglitt, 2012; Llanso, Tally, Silberglitt, & Anderson, 2013), NIST SP 800-30 (National Institute of Standards and Technology, 2012), Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) (Caralli, Stevens, Young, & Wilson, 2007), and Risk IT (ISACA, 2009; Schmittling, 2010). These methods are representative of approaches in use by organizations that employ structured threat-informed cyber risk assessment and their descriptions are available in open literature. This is not an exhaustive survey, and in particular does not include proprietary and other closed-source methodologies. The approaches mentioned here, which are described in more detail in Appendix E, have several themes in common, including an enumeration of the critical IT assets and data, consideration of threats (e.g. in terms of vulnerabilities, adverse events, or adversary capabilities), expert scoring (e.g. estimated likelihood of event occurrence, level of adversary effort to cause the effect, consequence/mission event impact), and methods which combine the scores in order to

identify high priority threats. For purposes of the research gap we seek to fill, the key take-away about threat-informed cyber risk assessment is this: most existing cyber risk assessment methods stop short of recommending mitigations.

Mitigation Optimization Analysis

A number of authors have tackled the problem of mitigation optimization analysis; that is, taking a longer list of possible mitigations and prioritizing or down-selecting to a shorter list based on a set of defined objectives. These methods are summarized below and described in more detail in Appendix F.

Multi-criteria decision-making (MCDM), also known as multiple-criteria decision analysis (MCDA), is widely applied to security portfolio[2] selection (Barnard & von Solms, 2000; Fenz et al., 2011; Patterson, Nutaro, Allgood, Kuruganti, & Fugate, 2013; Sawik, 2013; Schilling & Werners, 2016; Weishäupl, 2017; Yevseyeva, Basto-Fernandes, Emmerich, & Van Moorsel, 2015). MCDM is used to analyze problems where measures of costs and benefits exist and can be traded off to arrive at the best solution under the given constraints. Some MCDM techniques applied to mitigation optimization include or are based on fuzzy set theory (Otero, 2014), multi-attribute utility theory (i.e. value functions, knapsack strategy) (Fielder, Panaousis, Malacaria, Hankin, & Smeraldi, 2016; Panaousis, Fielder, Malacaria, Hankin, & Smeraldi, 2014; Shapasand, Shajari, Golpaygani, & Ghavamipoor, 2015; Smeraldi & Malacaria, 2014), evolutionary multi-objective optimization (EMO) also known as genetic algorithms (Gupta, Rees, Chaturvedi, & Chi, 2006; Kiesling, Ekelhart, Grill, Strauss, & Stummer, 2016; Kiesling, Strauß, & Stummer, 2012; Rees, Deane, Rakes, & Baker, 2011; Sarala, Zayaraz, & Vijayalakshmi, 2016; Viduto, Maple, Huang, & López-Peréz, 2012), analytic hierarchy process (AHP) (El-Gayar & Fritz, 2010), grey relational analysis (GRA) (Breier & Hudec, 2013), simple additive weighting (SAW) (Llanso, 2012; Llansó, McNeil, & Noteboom, 2019), the technique for order preference by similarity to ideal solution (TOPSIS) (Breier & Hudec, 2013), and preference ranking organization method for enrichment evaluation (PROMETHEE) (Lv, Zhou, & Wang, 2011). In addition, several authors combine game theory with MCDM techniques for security portfolio selection (Fielder et al., 2016; Panaousis et al., 2014; Wang &

[2] An organization's chosen list of mitigations is often referred to as a security portfolio.

Zhu, 2016). For purposes of the research gap we seek to fill the key take-away about mitigation optimization analyses is this: These approaches all assume that a starting set of possible mitigations exists on which to apply the prioritization/selection method; however, as we noted above, cyber risk assessment methods stop short of providing this data. A method to produce this initial mapping of potential mitigations to threats is the gap the current research seeks to fill.

Threat Taxonomies and Control Catalogs

A number of control catalogs exist in practice today, such as the Payment Card Industry Data Security Standards (PCI-DSS) (PCI Security Standards Council, 2015), HIPPA Security Standards (Centers for Medicare and Medicaid Services, 2007), and NIST Security and Privacy Controls for Federal Systems (National Institute of Standards and Technology, 2012). These catalogs are intended to prescribe controls for compliance with security mandates, however, they do not map the controls to the specific threats they counter. Likewise, a number of threat frameworks exist in practice, including the Common Attack Pattern Enumeration and Classification (MITRE, 2017a), Carnegie-Mellon taxonomy of operational cyber security risks (Cebula, Popeck, & Young, 2014), National Intelligence Cyber Threat Framework (National Security Agency, 2018), Open Threat Taxonomy (Enclave Security, 2015), and others (European Union Agency For Network And Information Security, 2016; Launius, 2018; Simmons, Shiva, Bedi, & Dasgupta, 2014). Of these, the CAPEC and Carnegie-Mellon frameworks contain representative mappings of threats to mitigations, but there is currently no published comprehensive source of threat-mitigation mappings.

Document-Driven Decision Support Systems

Casting our research problem as an information retrieval (IR) problem gives rise to three veins of DSS research for investigation: (1) using classification to judge whether each item in the mitigation corpus should be included in or excluded from a particular threat's mitigation set, (2) using a retrieval/ranking model such as commonly used in search engines to enumerate mitigations ranked according to their likelihood of relevance to the threat, and (3) some combination of the two. Lacking existing research dealing specifically with automated

matching of mitigations to cyber threats, our discussion here considers supportive analogous research.

Classification. Classification is a supervised machine learning technique in which a new item is assigned to its appropriate category by a classifier, an algorithm or model which has been trained to make such decisions after learning from training data consisting of items whose categories are already known. Classification-based document selection has been researched extensively in the context of the medical systematic reviews (SRs) underpinning evidence-based medicine. A number of studies have demonstrated the viability of using supervised machine learning classification to reduce manual workload in the abstract triage process for updating existing SRs (Aphinyanaphongs & Aliferis, 2003; Bañez et al., 2016; Bekhuis & Demner-Fushman, 2012; Bekhuis, Tseytlin, Mitchell, & Demner-Fushman, 2014; Cohen, Hersh, Peterson, & Yen, 2006; Frunza, Inkpen, & Matwin, 2010; García Adeva, Pikatza Atxa, Ubeda Carrillo, & Ansuategi Zengotitabengoa, 2014; Howard et al., 2016; Liu, Timsina, & El-Gayar, 2016; Matwin, Kouznetsov, Inkpen, Frunza, & O'Blenis, 2010; Mo, Kontonatsios, & Ananiadou, 2015; Shemilt et al., 2014; Timsina, Liu, & El-Gayar, 2016). Updating SRs has historically entailed a labor-intensive, time-consuming, multi-step process in which subject matter experts attempt to identify and down-select from the massive corpus of medical research all research pertinent to a particular medical question so that the research can be synthesized to answer the question. During the initial stage in the selection process, known as broad screening or abstract triage, human experts must review and make relevant/not-relevant judgments on many thousands of abstracts returned by an initial keyword search. The goal of the triage stage is to exclude those abstracts that are obviously irrelevant, but include the rest for further consideration in the second stage. The triage stage demands high recall[3] (>95% (Cohen et al., 2006)) to ensure all relevant research is considered, but is less stringent about precision, tolerating a few false positives. This reflects the customs of the problem domain: It is unacceptable to overlook research relevant to the problem for this could impact the overall quality of the SR. On the other hand, it is tolerable to include some potentially irrelevant documents because these will be screened out by human reviewers in the next stage of the

[3] Recall is the ratio of relevant records retrieved to the total relevant records in the corpus. Precision is the ratio of relevant records retrieved to total records retrieved (Singhal, 2001).

process (Matwin et al., 2010). Comprehensiveness and currency of SRs is confounded by the large, continually-evolving, and highly technical nature of medical literature. In addition, SRs typically operate on a large corpus of candidate studies where only a small percentage (e.g. <15%) will ultimately be true positives selected for inclusion in the synthesis (Kontonatsios et al., 2017; Shemilt et al., 2014), a condition known as imbalance.

The document selection process for SRs bears stark similarities to our research problem in which we have a large corpus of continually-evolving, highly technical cybersecurity literature and we want to present mitigation documents for a given threat while omitting those that are extraneous. Moreover, like SRs, threat-mitigation matching operates on an imbalanced corpus of candidate mitigations where only a small percentage are relevant to any particular threat. A key similarity between selecting literature for a SR and selecting mitigations for a threat may be the value judgment that high recall is more important than high precision. We elect to favor recall in the precision-recall tradeoff for the same reason this choice was made in the case of medical SRs and we assume that a few false positives can be manually screened out, if necessary.

Ranked Retrieval. Commonly used in search engines, ranked retrieval considers relevance between a query and a document, not as a binary concept, but as a matter of degree. A retrieval model assigns a relevance score to each query-document pair via a ranking function. When ordered in descending sequence by the relevance score, those documents at the top of the list are the documents deemed to be most relevant to the query. Unfortunately, for purposes of making binary relevant/non-relevant decisions using ranked results, one must determine a cut-off point in the ordered list. This is a challenging problem because, in general, the number of relevant results expected is not known a-priori (Manning, Raghavan, & Schutze, 2009).

Similarity-based text retrieval models judge the relevance of document to a query in a manner that does not require all the words in the query to be present in the document. The Vector Space Model is a well-known document representation scheme in text retrieval. Each document is represented as a vector of the document words or terms where each word has a weight indicating its overall importance in the document. Some common weighting schemes are binary (term presence or absence), term frequency (TF, the number of times the term appears in the document), and term frequency/inverse document frequency (TFIDF), a technique that counterbalances the term frequency with a factor accounting for the total number of documents

that contain the term. When aggregated, the document vectors form a term-document matrix that can be manipulated using matrix mathematics. The Vector Space Model represents the corpus of document vectors in a common vector space in which the similarity between two documents or between a document and a query can be calculated via a distance measure known as cosine similarity (Manning et al., 2009; Turtle & Croft, 1992). The result of testing the similarity of a query to a corpus of documents will be a ranked ordering of the documents from most to least similar based on the individual words in the documents.

Latent semantic analysis (LSA) (also called latent semantic indexing (LSI) in some contexts) is another similarity-based retrieval model. It a statistical technique that attempts to address language complexity, such as synonymy, by considering the term-document relationships as a statistical distribution representing an "underlying latent structure" of the document corpus (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990). In LSA, the term-document matrix is transformed via singular-value decomposition (SVD) resulting in a semantic space representing the "major associative patterns" (Deerwester et al., 1990) in the corpus. This semantic space contains the "best K orthogonal factors" (Foltz, 1990) which approximate the original document matrix and, importantly, the most closely associated terms and documents are clustered near one another such that terms that did not appear in a given document (e.g. synonyms) may still be located near the document due to overall word association patterns. Deerwester et al. observed substantial improvement for LSA-based text retrieval over keyword-based retrieval. They also noted as a practical matter that the transformed matrix is substantially smaller than the original term-document matrix, requiring only 50-150 factors compared with the hundreds or thousands of words typical of a large document corpus.

Several studies analogous to our present research utilized similarity-based ranked retrieval to perform technical document matching, two based on keywords and one based on LSA. Swanson et al. (Swanson & Smalheiser, 1997) developed an automated method based on keyword searching for linking complementary sets of articles in the MEDLINE database. In another study, Goldrich et al. (Goldrich et al., 2014) applied search engine technology, including Apache Lucene (Apache Foundation, 2018), keyword matching, key phrases, query expansion with synonyms, and the WordNet lexical database (Miller, 1995; Princeton University, 2017) to match cybersecurity requirements stated as text to descriptions of research

projects in order to point out research aligned with the requirements. Finally, Foltz (Foltz, 1990) applied LSA to find new relevant documents in a corpus based on an existing profile of documents that had been previously deemed relevant. Foltz first constructed a semantic space of articles a priori deemed relevant. To determine if a new document was relevant, it was first transformed to the semantic space of relevant articles. Then, if its nearest neighbor was another relevant document or if it was neighbors with more relevant articles than non-relevant articles it was relevant. Using the nearest neighbor approach and averaging the precision at 3 levels of recall (.25, .5, and .75), Foltz's LSA-based method demonstrated between 13% and 25% improvement in retrieval results on three data sets over keyword matching based on 190-240 dimensions.

Hybrid Approaches. A few authors have explored the combined use of classification and ranked retrieval techniques in text mining. For example, Manning et al. (Manning et al., 2009) discussed an approach for machine-learned relevance scoring where each training data instance consists of query terms (q), a document (d) reference, a binary judgment of the relevance of d to q, the cosine similarity (s) of d and q and the query term proximity between d and q. Nakamoto (Nakamoto, 2011) discussed a concept similar to Manning et al., except using Okapi BM25 (Robertson, Walker, Jones, Hancock-Beaulieu, & Gatford, 1994) and PageRank scores as features (instead of relevance and cosine similarity) and returning a relevance ranking instead of a binary decision. Wiener et al. (Wiener, Pedersen, & Weigend, 1995) utilized LSI for feature reduction instead of term selection (picking a representative subset of the original terms) to identify topics using a neural network classifier in a corpus consisting of more than 11,000 unique terms. Gee (Gee, 2003) described a method for classifying email as spam or not-spam using an "LSI-inspired" ensemble classifier implemented in three stages, where the stages are similar to Foltz (Foltz, 1990). Gee's method achieved very high (>0.98) precision and recall on both the spam and not-spam classes when tested using just the nearest neighbor classification strategy, just the majority strategy, and the two strategies in ensemble with the tie-breaker logic.

IR Evaluation. The ability to evaluate the effectiveness of a machine learning approach is crucial to ensuring that the results are useful and not just a manifestation of chance. As we have cast our research as an information retrieval problem, we now consider IR evaluation methods. The documents in the corpus will fall into one of four categories at the conclusion of a particular query: retrieved and relevant or true positive (TP), retrieved but not relevant or false

positive (FP), not retrieved but relevant or false negative (FN), and not retrieved/not relevant or true negative (TN). Accuracy, precision, and recall are the most common measures of effectiveness in IR. They are all proportions with values between 0 and 1 inclusive based on the above categorization of retrieval results (Manning et al., 2009). Powers points out a bias in that these common measures tend to understate a method's ability to correctly identify non-relevant instances (Powers, 2007). The ability to rule out non-relevant instances can be a useful measure of workload reduction.

Accuracy is the proportion of correctly classified items (TP + TN) to all items (TP + TN + FP + FN). It is generally a poor measure of IR effectiveness because it does not distinguish success between the relevant (R) and non-relevant (NR) document classes. In particular, accuracy is heavily swayed in cases where the data is imbalanced, which is almost always the situation in IR. For example, a method that arbitrarily classifies all documents as NR would appear highly accurate in a corpus with 90% NR documents (Manning et al., 2009) even though it would incorrectly classify all the R documents.

Precision is the proportion of retrieved and relevant items (TP) to all retrieved items (TP + FP) also called *confidence* in some fields. *Recall* is the proportion of retrieved and relevant items (TP) to all relevant items (TP + FN) also called *true positive rate* or *sensitivity*. There is an inverse relationship between precision and recall such that when one goes up the other goes down. The weighted harmonic mean of precision and recall (*F-measure*) is a measure used to trade-off precision and recall. The *balanced F measure* weights precision and recall equally but weights can be set to emphasize one over the other if desired. The *area under a precision-recall curve* (AUC) and the *balanced F-measure* are often used as measures of IR effectiveness when balanced performance is sought (Manning et al., 2009; Powers, 2007; Raghavan, Jung, & Bollman, 1989).

Because an IR query commonly results in a ranked list of retrieved results, the expected number of which is not known in advance, computation of a single overall precision and especially recall can be challenging. In IR precision/recall data points can instead be considered at each new relevant document in the ranked list. This gives rise to measures such as *R-precision* or precision at a selected recall value (P@R or P(R), e.g. P(R=0.9)), and *precision at rank* (P@K or P(K)), which is the precision calculated assuming a cut-off at a fixed location in the ranked list. In user-facing search applications, it is widely accepted that the user generally only looks

at the first page of search results; hence, P@K is often used to measure search effectiveness in user-facing search applications. Because of the arbitrary fixed cut-off, P@K does not take into account the variability in number of relevant results and thus it can be skewed when the actual number of relevant entries is much greater or less than the fixed cut-off. *R-precision* compensates for this weakness of P@K, essentially by computing P@K where K is the number of relevant entries that must be returned to achieve the desired recall (Manning et al., 2009; Raghavan et al., 1989).

Sensitivity and specificity are measures used in fields such medicine and behavioral science to judge the effectiveness of diagnostic tests. *Sensitivity* (or equivalently, true positive rate, recall, probability of detection) is the proportion of true positives to all positive instances or the extent to which actual positive instances are not ignored. In contexts where the objective is to correctly identify all positives, such as medicine, recall is a primary evaluation metric (Powers, 2007). *Specificity* (true negative rate) is the proportion of true negatives to all negative instances or the extent to which actual negative instances are classified as such (Altman & Bland, 1994). In contexts where the objective is to rule out large swaths of negative instances, such as SRs, specificity can be an effective evaluation measure. The *fallout* (or *false positive rate*) is the proportion of false positives to all negative instances, i.e. the probability that a non-relevant document will be retrieved.

The best evaluation measures can only be chosen by considering the requirements of the particular IR scenario. We discuss the evaluation methods we have chosen for our research in Chapter 3. In some applications, recall may be more important than precision (e.g. medical SRs and threat-mitigation matching) or vice versa. Recall should be emphasized when it is essential not to miss any relevant documents and some false positives can be tolerated. On the other hand, precision should be emphasized when a subset of documents is sufficient to answer the request (Manning et al., 2009). Finally, according to Raghavan et al. the "usefulness of a retrieval system is determined to a great extent by how closely it can characterize the dichotomy" of relevant vs non-relevant documents for its intended purpose (Raghavan et al., 1989).

Summary

In the Literature Review we discussed threat-informed cyber risk assessment and mitigation selection optimization approaches to delineate the boundaries of the gap that our research addresses. Casting the research problem as an information retrieval problem, we identified pertinent research upon which to build. This includes a robust body of work applying classification techniques to medical systematic reviews, a modest body of work applying similarity-based techniques to technical document matching, and examples of combining the two. Finally, we explored the literature supporting evaluation methods in DSS and IR. In subsequent chapters we will refer back to this existing theory as a basis for our research.

CHAPTER 3

RESEARCH METHODOLOGY

Design Science Research

We selected the Design Science Research Model (DSRM) (Hevner & March, 2004; Peffers et al., 2007) as the research framework within which to organize our research. DSRM attempts to solve so-called "wicked problems" through the development and evaluation of IT artifacts (Vaishnavi & Kuechler, 2004).

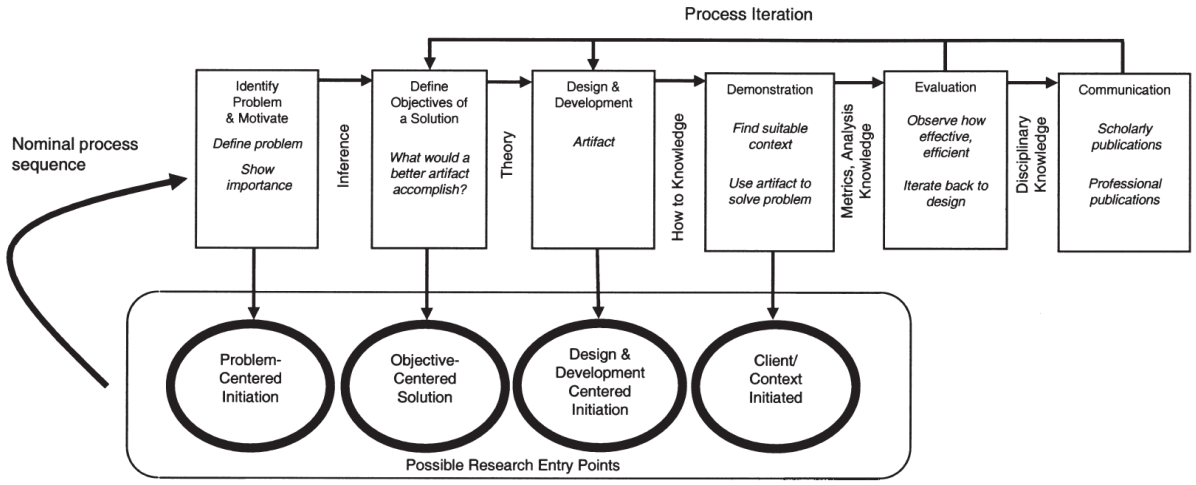

Figure 2. DSRM Model from (Peffers et al., 2007)

Peffers et al. describe an iterative process with six stages as illustrated in Figure 2. In Table 1, we enumerate the DSRM stages and demonstrate the alignment of our research to them. The DSRM is appropriate for this research because we want to create an IT artifact to solve a challenging problem for which a solution will contribute to theory and practice.

Table 1. DSRM Stages and Alignment

Stage		Alignment
1	Identify and motivate the problem	In Chapter 1 we identified and motivated the threat-mitigation matching problem and discussed our proposed contributions. These are crucial steps in the DSRM because

Stage		Alignment
		they establish problem relevance, where relevance is judged in the context of a "heretofore unsolved and important" problem for a "constituent community" of IS practitioners (Hevner & March, 2004).
2	Define objectives of a solution	In Chapter 3 (this chapter) we define the objectives of a solution to our research problem. The objectives provide a preview of the desired end state and set the stage for artifact evaluation.
3	Design and develop the artifact iteratively and based on existing theory	In Chapter 2, we discussed pertinent literature. The DSRM requires that we draw upon existing research as the basis for the artifact; the Literature Review paved the way for doing so. Later in Chapter 3 (this chapter) we discuss our iterative approach to design and development, synthesizing from the cyber and DSS domains.
4	Demonstrate the artifact by using it to solve an instance of the problem	In Chapter 4, we describe the artifact (method) and discuss the results of applying instantiations of the method to solve five test instances of the problem.
5	Evaluate how well the artifact solves the problem; iterate back to design	In Chapter 4, we evaluate the effectiveness of the artifact using evaluation measures drawn from the literature. In Chapter 5, we summarize our contributions and propose future work. These discussions set the stage for future iterations of design in the spirit of the DSRM.
6	Communicate results to scholarly and practitioner communities	This dissertation and the associated defense presentation satisfy the DSRM requirement for communication. We designed the artifact using rigorous, practitioner-accepted modeling techniques to facilitate communication to both scholarly and practitioner communities.

Objectives of a Solution

Defining the objectives of a solution to the research problem at hand is an important predecessor to artifact design because it previews the desired end state. Objectives also provide the foundation on which to build an evaluation strategy. A solution to our research problem described above will:

- Process existing English language text documents where each separately describes either a threat or a mitigation (e.g. threat models, practice manuals, control catalogs, vendor product white papers)
- Provide an automated method for recommending (matching) relevant mitigations when presented with a threat
- Match a high percentage of relevant mitigations for a given threat while avoiding selection of non-relevant mitigations
- Accommodate (be extensible to) new and evolving threats and mitigations,
- Provide utility to cybersecurity experts in mitigation selection, and
- Be able to be used in a system that allows for reuse of the artifact and the matches produced by the artifact.

Theoretical Background

The DSRM emphasizes design and evaluation rigor through building upon existing research from the literature. Because knowledge about threats and mitigations is largely expressed in unstructured or semi-structured text documents, our idea is to cast the threat-mitigation problem as an information retrieval problem, using the threat as a query and the mitigation documents as the corpus to be searched, and then build on applicable DSS research. Applying techniques described in the literature we considered artifact designs from three categories for the threat-mitigation matcher:

1. **Classification**. Drawing from the medical SRs research, approaches based on classifying mitigation documents as relevant or not relevant to a given threat
2. **Ranked Retrieval**. Drawing from (Swanson & Smalheiser, 1997), (Goldrich et al., 2014), and Foltz (Foltz, 1990), approaches based on ranked retrieval, and

3. **Hybrid**. Drawing from (Manning et al., 2009), (Nakamoto, 2011), and (Gee, 2003), hybrid approaches that combine techniques from ranked retrieval in conjunction with classification.

In Chapter 4, we describe an iterative process wherein we experiment with several artifact instantiations in each design category. We discuss the results of these trials, which instantiations we decided to advance, which we left behind, and why, with evaluation criteria drawn from the theoretical bases discussed in the Evaluation section of the Literature Review.

Design and Development of the Artifact

The nature of design is best described as a cycle consisting of brainstorming ideas and testing them against the solution objectives (Simon, 1997), continuously refining ideas until the desired end state is reached. Prototyping, solution validation, and feedback are emphasized, aligning with the iterative nature of Design Science Research and, importantly, helps distinguish Design Science Research from routine professional design (Hevner & Chatterjee, 2010). In Chapter 4, we discuss highlights of the iterative design process we followed during development of our artifact.

The DSRM requires that artifact be constructed and evaluated with rigor (Hevner & March, 2004). To promote design rigor, we have drawn from research and practice in the cyber risk assessment, mitigation optimization analysis, and DSS domains, as discussed in the preceding Literature Review. Moreover, we represented design using the unified modeling language (UML) (Booch, Rumbaugh, & Jacobson, 2000) and entity-relationship drawings (ERD) (Chen, 1976). These are rigorous methods for modeling software that are commonly accepted and understood in the IS practitioner community. In addition, we designed the artifact using object-oriented software practices intended to increase modularity, improve quality, and make designs and software more resilient to evolution. The artifact produced by this research is described in Chapter 4. An architecture for the practical use of the artifact is described in Appendix C. We touch on evaluation rigor briefly here and more fully in Chapter 4.

Demonstration and Evaluation Plan

In the DSRM, demonstration and evaluation work together to show that the artifact effectively solves the problem. Hevner and March state a number of evaluation methods that top the rigor threshold, classifying them into the following categories: observational (e.g. case or field study), analytical (e.g. quantitative comparisons, such as of time or cost), experiment or simulation, testing, and descriptive (e.g. argument or scenarios) (Hevner & March, 2004). In the present research, we demonstrate instantiations of the artifact by applying them to a corpus based on the Common Attack Pattern Enumeration and Classification (CAPEC) dataset version 2.11 (MITRE, 2017a). Table 2 summarizes our evaluation plan, including artifact evaluation criteria aligned with the solution objectives. To ensure evaluation rigor, the evaluation methods are drawn from among those given by Hevner and specific performance measures are drawn from the DSS and IR domains. Note that the most important objective is the third one as the others are only germane after the artifact achieves satisfactory matching performance. Utility is also important as we wish to solve a practical problem. The results of artifact evaluation are discussed in Chapter 4.

Table 2. Solution Objectives with Evaluation Methods and Criteria

Objective	Evaluation Criteria
Process existing English language text documents where each separately describes either a threat or a mitigation.	**Testing**: Demonstrate that the instantiated artifact accepts English language text documents about threats and mitigations.
Provide an automated method for recommending (matching) relevant mitigations when presented with a threat.	**Testing**: Demonstrate that the instantiated artifact will propose matching mitigations when a threat is given.
Match all or nearly all of the relevant mitigations for a given threat while avoiding selection of non-relevant mitigations.	**Analytical**: Achieve acceptable performance measures on test data. We emphasize high recall to retrieve nearly all relevant mitigations. We emphasize moderate to high precision and low false positives to avoid selecting non-relevant mitigations.

Objective	Evaluation Criteria
Accommodate (be extensible to) new and evolving threats and mitigations.	**Descriptive**: Describe how the artifact is extensible for future threats and mitigations **Analytical**: Achieve acceptable performance measures on test data.
Provide utility to cybersecurity experts in mitigation selection.	**Descriptive**: Integrate results of performance, extensibility, and reuse evaluations to make a logical argument about utility. We emphasize high sensitivity to rule out most non-relevant mitigations leading to reduced workload for the CSE.
Be able to be used in a system that allows for reuse of the artifact and the matches produced by the artifact.	**Descriptive**: Describe how the artifact is reusable and how the knowledge produced by the artifact is reusable.

CHAPTER 4

RESULTS AND DISCUSSION

Data Source Description

We used version 2.11 of the Common Attack Pattern Enumeration and Classification (CAPEC) dataset (MITRE, 2017a) as the data source for this research. The CAPEC dataset is available for download and can also be browsed online (MITRE, 2017b). CAPEC is an existing corpus of attach patterns (i.e. threats) expressed in English language documents packaged in an XML structure. Although mitigation mapping is not the focus of CAPEC, some attack patterns include illustrative mitigations. CAPEC contains a hierarchical representation of attack patterns, where the highest level consists of meta attack patterns. These are architecture/design-focused and not based on specific technologies or implementations. Each meta pattern decomposes into several standard attack patterns, which are more detailed and include information about the goal of and technique used in the attack. Each standard pattern decomposes into detailed patterns, which are the most granular. For our purpose, we focus on the standard patterns, which strike a good middle ground between the meta and detailed patterns and are most representative of the level of specificity for threats in the cyber risk assessment domain. There are 127 standard threats in CAPEC. There are approximately 600 mitigation texts in the corpus. The number of mitigations mapped to each standard threat varies from 0 to about 10. These mappings are intended to be representative and not comprehensive as threat-mitigation mapping is not the intent of CAPEC.

CAPEC has existed in the cybersecurity community since 2007. We consider the CAPEC threat-mitigation mappings to be ground truth and we recognize that the quality of the data is key to our results. While we do not have objective evidence of the quality of the CAPEC threats, mitigations, and mappings, we accept CAPEC's heritage as an indicator of sufficient quality for this proof of concept research. By personal inspection, we searched CAPEC for threats which had at least a paragraph of descriptive text and about 10 relevant mitigations for use as labeled data. We were able to find five threats and associated mitigations which are

suitable test cases for our purpose. We also found some weaknesses in this data source, which we discuss below.

We used XML parsing to decompose CAPEC into its component threat documents, mitigation documents, and mappings between the two. During parsing, we preserved selected information from the document structure (e.g. title, description, threat category) per related work (Cohen, 2008; Matwin et al., 2010; Mo et al., 2015; Small, Wallace, Brodley, & Trikalinos, 2011) which suggests that certain parts of the document may yield impactful features for classification. The following data was extracted from CAPEC for threat documents:

- ID #
- Title (free text)
- Description (free text)
- Abstraction level (meta, standard, detailed)
- Domain of attack
- Mechanism of attack
- Parent attack pattern
- Immediate children attack patterns

The following data was extracted from CAPEC for mitigation documents:

- ID #
- Title (constructed by taking first 75 characters of the description)
- Description (free text)

The following data was extracted from CAPEC for existing threat-mitigation mappings. A subset of these mappings was used as labeled data for training models and the rest was used for testing.

- Threat ID #
- Mitigation ID #
- Relevant/not relevant indicator

Strengths of the CAPEC data for our purpose include detailed threat descriptions, metadata including categorical and hierarchical relationships, open[4] accessibility, and available threat-mitigation mappings. The CAPEC data has several key weaknesses when considered for our application. We highlight those weaknesses and the work-arounds we implemented here. First, the threat documents are more robust in length and content than the mitigation documents. Since we want to treat the threat as a query, the opposite situation would have been better. Second, the data is imbalanced; that is, there are a relatively small number of relevant mitigation instances per threat compared to non-relevant instances. We lessened this weakness by drawing in some additional mitigations from other sources. Third, and perhaps most concerning, the quality and style of the prose within the threat and mitigation documents varies significantly from one document to the next. For document-driven DSS methods to produce good threat-mitigation matches, the threats and mitigations must both be well-described. We addressed this weakness by selecting a handful of the best quality threat documents from CAPEC to use as our demonstration cases. Fourth, we found a few situations where, due to human error, the mappings were erroneous. Since we rely on the mappings as ground truth, we corrected the errors manually. Finally, we had initially hoped to utilize the Domain of Attack and/or Mechanism of Attack metadata in CAPEC as features to support classification in a manner similar to the way the Medical Subject Headings (MeSH) (Lowe & Barnett, 1994) support classification for medical SRs (Timsina et al., 2016). Unfortunately, the existence of this metadata within the CAPEC proved to be insufficient for our purpose, so we had to abandon this idea.

Although the CAPEC weaknesses represent minor inconveniences, they do not invalidate our research because our research is not specifically about the CAPEC data; it is more generally about the concept of threat-mitigation document matching. CAPEC is simply a vehicle, a convenient source of labeled data (the only non-proprietary source we could find). Finally, we note that none of the above criticism is meant to detract from the value of the CAPEC data for its original intended purpose. We acknowledge their efforts to produce it and thank them for making their work openly available for use.

---

[4] "The MITRE Corporation (MITRE) hereby grants you a non-exclusive, royalty-free license to use Common Attack Pattern Enumeration and Classification (CAPEC™) for research, development, and commercial purposes. Any copy you make for such purposes is authorized provided that you reproduce MITRE's copyright designation and this license in any such copy." (MITRE, 2017a)

Iterative Design

In this section, we discuss highlights of the iterative design and experimentation that led to our artifact and instantiations. At the outset, we had three design concepts for the threat-mitigation matcher: classification, ranked retrieval, and a hybrid of the two. We explored a number of designs, including various classifiers, feature sets, and feature reduction techniques. Details of the design iterations are contained in Appendix D and summarized in the next few sections.

We used precision, recall, and the rate of false positives to judge the merits of each design. We chose these measures because they are among the ones most commonly used to compare text classifiers and retrieval models. In mitigation selection, omitting a relevant mitigation (recall error or false negative) means a useful mitigation could be overlooked. On the other hand, including a non-relevant mitigation (precision error or false positive) means the CSE may be presented with a mitigation that does not actually protect against the threat. While both are undesirable situations, we emphasize recall (i.e. to present all relevant mitigations) in our artifact with the assumption that a few false positives are tolerable and we can rely on the CSE to reject them during the screening phase (similar to the process for medical SRs).

Tool Choices

For some of the classification designs, we used the Waikato Environment for Knowledge Analysis (Weka) data mining toolkit presented by the University of Waikato (Kaluža, 2013). We selected this toolkit because it is well-known in data mining, remains under active development and use, has a robust user interface for experimentation, and also has a Java application programming interface (API) which we found attractive for practical purposes. In particular, we used the Weka SMO classifier, which implements the sequential minimal optimization algorithm for training a support vector classifier as described in (Platt, 1998). We also utilized scikit-learn, a Python machine learning environment (Pedregosa, Weiss, & Brucher, 2011) developed under the auspices of INRIA (“About us,” 2019) for some classification trials. We selected this toolkit because it is well-known in data mining, remains under active development and use, is well-documented, has a robust API, and supports some additional evaluation methods beyond what we could obtain from Weka. In scikit-learn we used

C-support vector classification (SVC). Both Weka SMO and scikit-learn SVC are based on LIBSVM (Chang & Lin, 2018), the most common SVM library.

For the keyword/phrase-based designs, we used the keyword/phrase extraction library implemented by Paco Nathan (Nathan, 2010). It is based on the TextRank algorithm described in (Mihalcea & Tarau, 2004).

For some of the ranked retrieval designs, we used the Apache Lucene (Apache Foundation, 2013) implementation of the Vector Space Model. We selected Apache Lucene because it is well-known, actively developed, well-documented, and has a robust Java API. For other ranked retrieval trials, we used the Gensim topic modeling toolkit presented by Radim Rehurek (Rehurek, 2018). We selected Gensim primarily for its LSA implementation. It is well-known, actively developed, and well-documented. It is implemented in Python and has a robust API that facilitates integration into an overall architecture.

"One for All" Designs

We initially wondered if there was a way to implement a "one for all" approach where a single matcher would determine relevant mitigations for any threat contained in the corpus. We explored this concept in two hybrid designs, a SVM classifier based on LSA features and a three-stage voting classifier also based on LSA (Gee, 2003). These are discussed in more detail in Appendix D. Neither of the "one for all" designs produced results better than random guessing. Intuition suggests that the relationship between one threat and its relevant mitigations may be different from the next threat/mitigations, such that combining many such relationships in a single semantic space may dilute the relationships. Thus, we abandoned the "one for all" avenue of investigation and proceeded on the "per threat" route.

"Per Threats" Designs

Following the medical SR literature discussed in the Literature Review, we started with a single threat and some labeled mitigation data that contained instances relevant and not relevant to the threat. We had an intuition that the best approach for one threat would also work for other threats. In order for the "per threat" approach to solve the problem at hand, we would have to eventually train a classifier for each existing threat and likewise for new threats that

come along; however, this does not seem like an unreasonable requirement. First, while new threats do come along, the set of known threats is relatively stable over time. In the ten months since we started this research, the CAPEC dataset has undergone two subsequent releases but only two new standard threats have been added to CAPEC in that time. Second, building the classifiers can eventually be automated using the API provided by the machine learning toolkits.

By browsing threats using the online version of the CAPEC dataset (MITRE, 2017b), we selected threat 49, password brute force guessing, as our first test case. We selected this threat because it had robust descriptive text and at least 10 relevant mitigations in the labeled data. Figure 3 shows a summary of the precision, recall, and false positive rates (cross-validation statistics) for several "per threat" designs. The bracketed [C], [TR], and [H] in the design names indicate the design concept: classification, text retrieval, or hybrid. For the classification and hybrid approaches, we show the cross-validation statistics for both the R and NR class. For the text retrieval designs, it is customary to evaluate based just on relevant results retrieved.

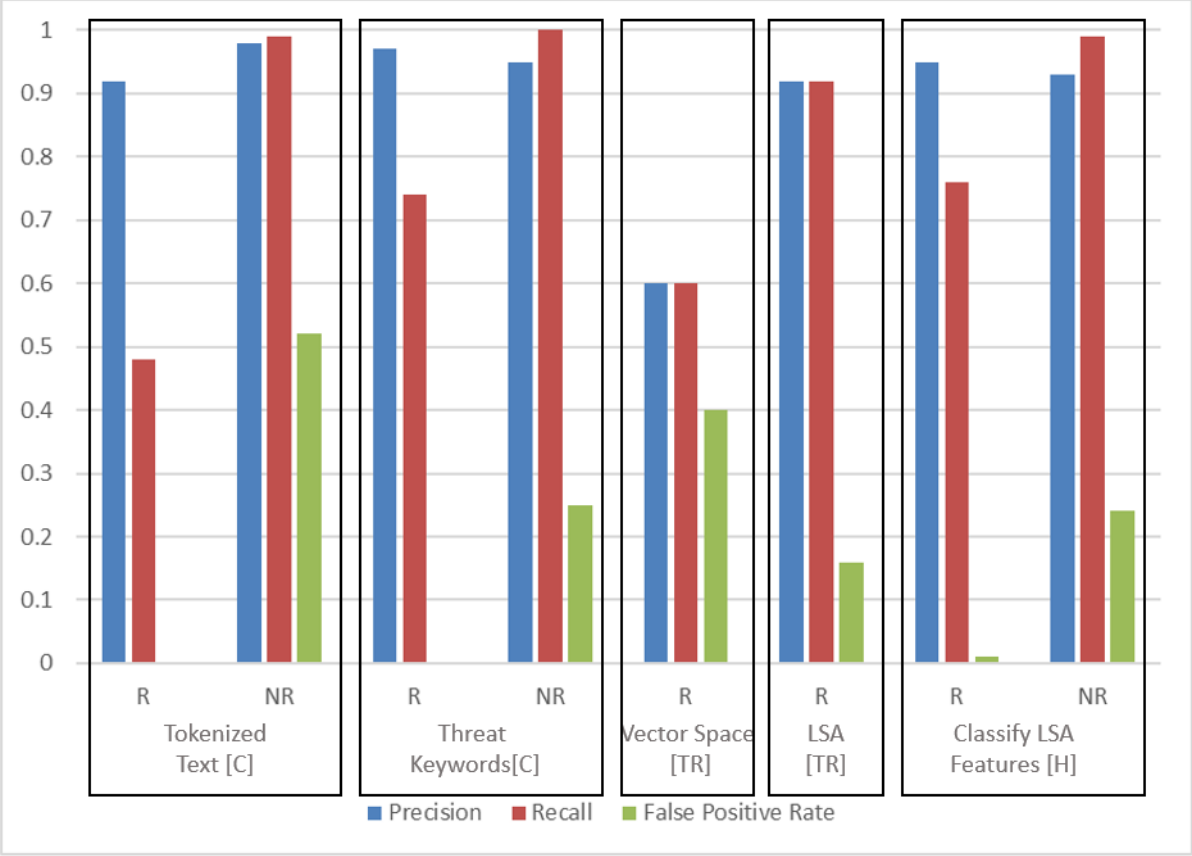

Figure 3. Summary of "Per Threat" Iterations

“Per Threat” Classification

We initially tested several classifiers before finally deciding to go forward with SVM. SVM has been shown to perform favorably for text classification, especially when the number of positive instances per category is small (Platt, 1998) and the feature set is large (Joachims, 1998). We discuss our selection of SVM further in the upcoming Artifact Design and Rationale sections. We experimented with two classification strategies for the “per threat” approach, one using the full text of the mitigations and the other using threat keywords/phrases.

Full text strategy. We investigated SVM classification of the full mitigation text. We performed tokenization of the text, removed numbers and punctuation, converted the text to lower case, stemmed, removed stop words, and retained the most frequent 1,000 words as TFIDF features plus the R/NR label. The corpus consisted of about 600 mitigation instances, 9 of which were relevant. We evaluated models with and without an information gain filter for feature reduction. As shown in Figure 3, the best of full text models had high precision (0.92), no false positives, but unacceptably low recall (0.48) on the R class. On the NR class, precision and recall were very high (>0.99) but with a 50% false positive rate. The model was very good at correctly classifying non-relevant instances, partially due to the class imbalance in the data, but it was not good at correctly classifying relevant instances, likely for the same reason. It became apparent that it was necessary to do something about the class imbalance. In addition, note that this approach does not utilize any information from the threat; thus, such an approach may not generalize to other threats.

Keywords/phrases strategy. An inspection of the mitigation text for the relevant examples revealed that those which were correctly classified in the full-text strategy have in common some words from threat 49 suggesting keywords/phrases as a possible way to introduce information from the threat text into the approach, while also potentially improving the classification results. We used an implementation of TextRank (Nathan, 2010) to automatically extract keywords/phrases from the threat text. Some of these keywords/phrases were rather rough, so we decided to clean them up manually. Then we converted the keywords/phrases to lower case and removed stop words. We made an intentional decision not to stem the keywords/phrases, but in some cases we included important variances as keywords in their own right. Next, we investigated techniques to address the class imbalance in the data

(Cohen et al., 2006; Miwa, Thomas, O'Mara-Eves, & Ananiadou, 2014; Timsina et al., 2016). The most obvious solution was to supplement the relevant mitigations, so we extracted about a dozen additional documents relevant to threat 49 from the Internet and added them to the data. We also performed two-thirds random undersampling of the dominant (NR) class to reduce the NR instances and 100% Synthetic Minority Oversampling Technique (SMOTE) (He & Garcia, 2009; Liu et al., 2016) of the R class to increase the R instances. The SMOTE technique creates new instances of the minority class by drawing features from the K (e.g., 5) nearest minority neighbors based on Euclidean distance in the feature space. Undersampling can result in information loss while oversampling can lead to overfitting; however, due to the extreme imbalance, these were risks worth taking.

For this trial, the features consisted of threat 49 keyword/phrase counts. After balancing, the corpus consisted of about 220 mitigation instances, 20 of which were relevant. We used several different methods for determining the keyword/phrase counts, including a simple count of the times a keyword/phrase appeared in the mitigation document (TF), TFIDF, TF divided by the total number of words in the document, and 0 or 1 to indicate the keyword/phrase is present or absent in the document. Of these, the presence/absence approach yielded the best model. As shown in Figure 3, the best of the keyword/phrase models had high precision (0.97), no false positives, and improved recall (0.74) on the R class. On the NR class, precision and recall were very high (>0.99) but with a 24% false positive rate. Two important disadvantages of this design are as follows: manual intervention is required to extract the threat keywords/phrases and recall is still too low.

We were curious about the potential impact of additional under- and oversampling, so we experimented with 3/4 undersampling of the NR class, and 200% oversampling of the R class for threat 49. When comparing 3/4 undersampling versus 2/3 undersampling of the NR class for the same oversampling percentage (100%) of the R class, the precision, recall, and F-measure for 2/3 undersampling was better. When we increased oversampling of the R class to 200%, recall of the R class seemed to improve overall but with a small toll on precision. In the 200% oversampling case, the model failed to properly classify test samples. These results suggest that 3/4 NR undersampling was too much and, when combined with 200% R oversampling, the model was becoming overfit to the training data.

“Per Threat” Ranked Retrieval

As a possible alternative to classification, we investigated two ranked retrieval approaches to matching relevant mitigations for a given threat similar to (Foltz, 1990; Goldrich et al., 2014; Swanson & Smalheiser, 1997). First, we investigated ranking based on the Vector Space Model as implemented in Apache Lucene. We also investigated ranking based on Latent Semantic Analysis (Deerwester et al., 1990) as implemented in Gensim (Rehurek, 2018). The corpus consisted of about 600 mitigation instances, 25 of which were relevant. As expected per the Literature Review, LSA outperformed the Vector Space Model, retrieving 23 of 25 relevant items versus 15 of 25. To calculate precision and recall, we cut the ranked list at 25 and used the formulas discussed in the Literature Review. The main issue with this approach was lack of a general strategy for implementing the R vs NR cut-off point in the ranked list. While the number of R instances is known in the training data, it is unknown in the real world, making it challenging to choose a generalized cut-off point.

“Per Threat” Hybrid

Drawing from (Manning et al., 2009), (Nakamoto, 2011), and (Gee, 2003), we experimented with two hybrid approaches that combine ranked retrieval and classification. In one approach we used features from an LSA transform of the mitigation text plus the R/NR label in conjunction with the SVM classifier. This design was ultimately the one we selected for our artifact. We discuss it in greater detail in the upcoming Artifact Design section.

In the other hybrid approach, we developed a method inspired by Gee (Gee, 2003) and Foltz (Foltz, 1990) for classifying mitigations relevant/non-relevant to a given threat. First, we used LSA to create a semantic space from a training set of labeled mitigation documents and constructed an external index to maintain the known relevance status of each mitigation with regard to the threat. Each new mitigation document, M_n, was used as a query against the semantic space, returning a ranked list of other mitigation documents similar to M_n from most similar to least. The classifier used the ranked list to classify M_n in three stages. First, it was classified according to the class of its nearest neighbor in the space (i.e. the existing mitigation document whose similarity score is highest). Next, M_n was classified according to class of the majority of all results in the ranked list truncated at an arbitrary cut-off, C. Finally, if the

majority and nearest neighbor stages agreed, M_n was deemed to be of the nearest neighbor's class. If the majority and nearest neighbor stages did not agree, the dispute was settled by the third stage which attempts to detect the skew of M_n towards one class or the other. We implemented the first 2 stages using an arbitrary cut-off of top 5, but for the tie-breaker we took a simple default. This method yielded precision of 0.63 and recall of 0.83 with 3% false positives on the R class and 0.99/0.97/17% for the NR class. Two ties were encountered in the NR class indicating the need to consider better tie-breaker logic, but on further experimentation we did not observe viable tie-breaking logic so we removed the design from further consideration.

The best of the hybrid models was the design that combined SVM with LSA. This is the design on which we ultimately based our artifact. It is discussed in detail in the Artifact Design section. The corpus consisted of about 600 mitigation instances before balancing and 100 instances after balancing, 25 of which were relevant. For threat 49, the method yielded precision of 0.95 and recall of 0.76 with 1% false positives on the R class and 0.93/0.99/24% for the NR class. Recall was still too low, so we looked to the text to determine options for improvement.

Analysis of Text

Success in classifying textual data is heavily influenced by the characteristics of the text itself. Having experimented with a few variations, it made sense to pause and look closely at the text of threat 49 to gain insights on the matching successes and failures. In the training corpus, there were 25 known relevant mitigations for threat 49. Using diagnostic tools, we identified the mitigations commonly misclassified in the trials. We investigated these false positives (FP) and false negatives (FN) to better understand how they differed from the correctly classified instances. One thing the correctly classified instances had in common was that they contained text explaining how the mitigation addresses the threat. The false negatives lacked this explanatory text. The false positives fell into two categories: (a) some dealt with password vulnerabilities but not specifically password brute force guessing and (b) others dealt with brute force guessing but not of passwords. We hypothesized that improving the mitigation texts to include an explanation of how each one addresses the threat would improve the match results by reducing the FNs. Doing so also has practical benefits, allowing the CSE to better understand the reason a mitigation is relevant to the threat, to determine its applicability in context, and to

better convey the rationale to the decision-makers who fund mitigations. In some applications of text mining (e.g. ratings, surveys, news articles), the text "is what it is" and we have to use what we find. For threat-mitigation matching, it may be possible to influence the problem space; thus, we do have the luxury of recommending improvements to the threat and mitigation documents to better support automated matching in the future. With that in mind, we augmented the text of the FNs from other sources and then reran selected trials. A comparison of the cross-validation statistics for models trained on the unimproved and improved text is shown in Figure 5 and discussed below. In general, models trained with the improved text demonstrated better precision and recall in cross-validation statistics than models trained on the unimproved text. For threat 49 on the improved text, the method yielded precision of 0.96 and recall of 0.92 with 1% false positives on the R class and 0.97/0.99/8% for the NR class, leading us to select this design as the selected approach for our artifact.

Artifact Design

Our artifact is designed to leverage SVM classification and LSA ranked retrieval. The selected approach uses as features the R/NR label plus 200 features derived from an LSA transform of the mitigation text. Using LSA affords a feature reduction from 1,500 unique words in the plain text to 200 LSA topics. Model building is a three-step process, indexing, balancing, and training, as illustrated in Figure 4. Note that a model is built for each threat; thus, the mitigation documents input into the indexing stage are labeled as R/NR to the specific threat. The corpus consisted of about 600 mitigation instances before balancing and 100 instances after balancing, 25 of which were relevant.

In the **indexing** stage, for each mitigation text, stop words are removed, then the text is tokenized, lower-cased, and stemmed. A TFIDF representation of the corpus is computed then transformed using Gensim to an LSA semantic space or Latent Semantic Index retaining 200 topics. This is slightly higher than the number of standard threats in CAPEC and fits with optimal LSI dimensionality findings in (Bradford, 2008). Bradford observed favorable results when the number of topics was between 200 and 500 for a corpus with millions of documents. We selected the low end of Bradford's range because our corpus is much smaller. The LSA semantic space and an index containing the labels are saved for use in similarity queries.

During iterative design, we observed that the corpus was highly imbalanced in favor of NR instances. In the **balancing** stage, we utilize LSA similarity scores as a means to balance the training data. We query the mitigation LSA space using the full text of the threat document (tokenized, stemmed, lower-cased, and transformed to the semantic space) as a query. Then, we truncate the training data after the 100th ranked result, retaining the top 100 mitigation entries based on similarity to the threat text. This balances the data that will be input into training by reducing the number of NR instances. We intuit that this approach is better than simply undersampling at random and over-sampling with SMOTE for the following reasons. Undersampling at random could drop relevant entries of which we already have too few. Oversampling with SMOTE adds new instances to the corpus, but no new knowledge. Because the similarity score imparts some knowledge about the semantics of the entries, ingesting the most similar entries during training will keep most of the relevant entries and in addition the non-relevant entries that are most difficult to discriminate.

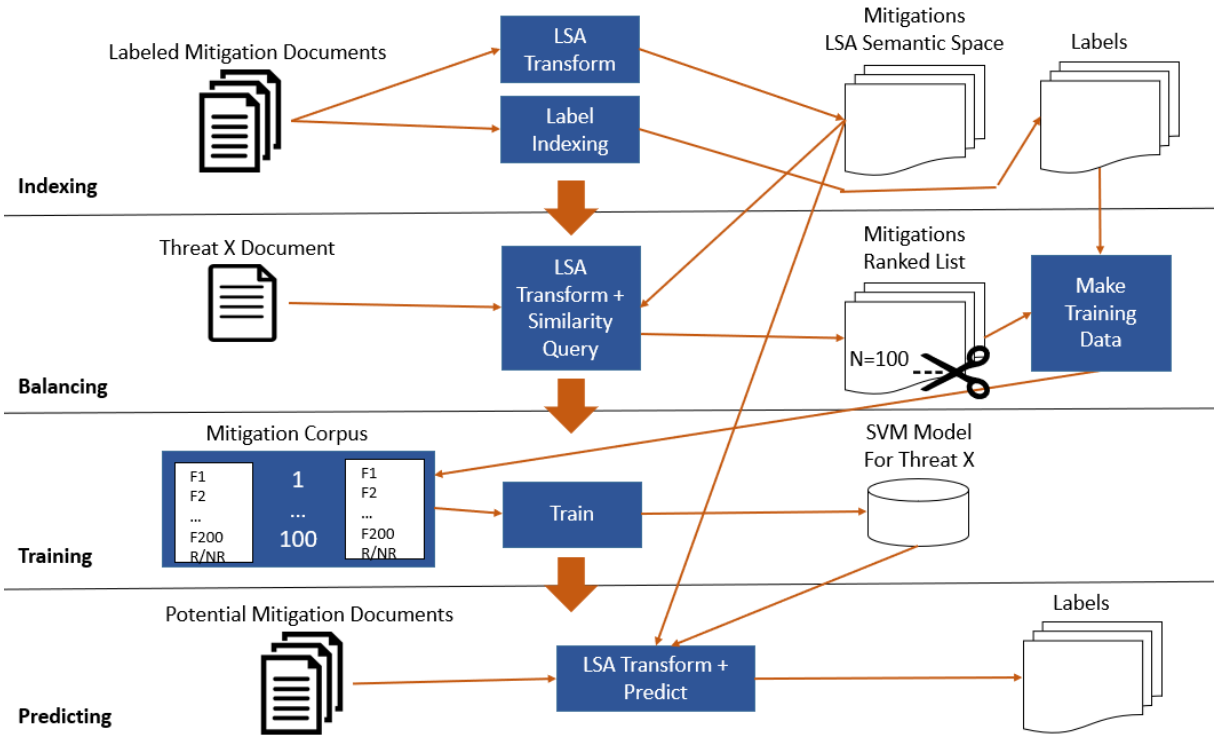

Figure 4. Artifact Design and Flow

In the **training** stage, we build an SVM classifier using scikit-learn for threat 49 (and later for other threats), inputting the top 100 most similar mitigations from the balancing stage for threat 49 and their labels into the learning process. The features consist of the LSA

representation of each mitigation (200 features) plus the R/NR indicator. We save the models for later use to predict the classes of new unlabeled mitigations.

We utilize the saved model in the **predicting** stage to classify new potential mitigations as relevant or not relevant to the threat associated with the model. First, the text is transformed to LSA features relative to the saved LSA space. Then the saved threat-specific classifier is applied to label the LSA-transformed mitigations. Evaluation of the model based on new test data is discussed in the upcoming Demonstration and Evaluation section.

Rationale for Selected Approach

In this section we discuss our rationale for the design of the selected method. We explain why we explored LSA, classification, and a combination of the two, why we selected SVM, and why we selected this approach as our method.

Why LSA? As we saw in the Literature Review section, Latent Semantic Analysis has been shown to improve retrieval of relevant documents from a corpus when compared to keyword search because LSA accounts for inherent complexities of natural language, including synonymy, by evaluating the entire corpus for recurring word patterns. These word patterns are used to construct a semantic space (a set of LSA topics) representing the corpus. Each document in the corpus is represented according to its degree of similarity with the topics of the space. In the literature, LSA is regarded as superior to keyword-based matching. In our experiments, we observed that LSA improved the matching of mitigations to threats over keyword-based matching, likely due to the cyber documents' complex word patterns.

Why Classification? Supervised machine learning classification, is a statistical approach for predicting the label or class of a new instance based on a model trained using existing instances whose classes are known. The instances are represented by features (independent variables) which are used to predict the label (dependent variable). The training process analyzes the features and associated labels and detects relationships that allow the class to be predicted for new instances represented according to the same features. Two-class classification of text documents has been successfully demonstrated in the literature for updating medical SRs as well as in our experiments for threat-mitigation matching. Moreover, classification does not suffer from the ambiguous cut-off problem encountered in matching by text retrieval.

Why SVM? SVM has been shown to perform favorably for text classification, especially when the number of positive instances per category is small (Platt, 1998). According to Joachims (Joachims, 1998), SVM is well-suited to text classification because many topics are linearly separable, the typical corpus has high dimensionality but few irrelevant features, and each document vector is sparse. Joachims provided experimental evidence that SVM "consistently achieved good performance on text classification," tolerated large feature sets without a need for reduction techniques, and did not require parameter tuning. None of our early experiments with SVM and other classifiers gave us reason to go against Platt's and Joachim's findings.

Why combine LSA and SVM? We used LSA in combination with SVM in our artifact for three reasons: (1) to reduce the tendency of the NR class to dominate the model by balancing the training data (from >99.99% NR before balancing to about 75% NR after), (2) as a feature reduction technique (from >1500 features before the LSA transform to 200 features after), and (3) because the LSA features are semantically richer, accounting for synonymy.

We crafted this design for the above reasons and selected it because of its high precision and recall and low false positive rate based on cross-validation statistics, along with the ability to fully automate construction of the "per threat" classifiers. The latter is a practical consideration; since we will have to build a large number of classifiers for a "per threat" design and may want to periodically rebuild the classifiers as new data is labeled, we prefer not to do it manually. The next best design was the threat keyword design, but it required manual intervention for every threat to extract the keywords.

Extensibility to Other Threats

Having seen promising results from the selected design, we wanted to know if these results would extend to other CAPEC standard threats. We chose threats 66 (SQL injection), 134 (email injection), 268 (audit log manipulation), and 593 (session hijacking) according to the same criteria we used to select threat 49. Then we compared cross-validation statistics for models trained for these five threats before and after text improvement. The left-most five sets of bars in Figure 5 show the precision, recall, and false positive rates for models trained for the 5 test threats before and after the text improvement. In the figure, "U" and "I" stand for

unimproved text and improved text, respectively. The rightmost set of bars shows the mean precision, recall, and false positive rate averaged across the 5 test threats. At a glance, this figure shows that the cross-validation measures are better after the text improvement, except for threat 268. Because threat 268 had 1.0 precision before the text improvement, precision declined slightly as expected when recall went up after the text improvement. As illustrated in Figure 5, precision is between 0.86 and 1.0 and recall is between 0.86 and 0.95 for all 5 test threats for improved text with false positive rate of 4% or less. Overall, although not a guarantee of generality, these classifier cross-validation statistics are favorable.

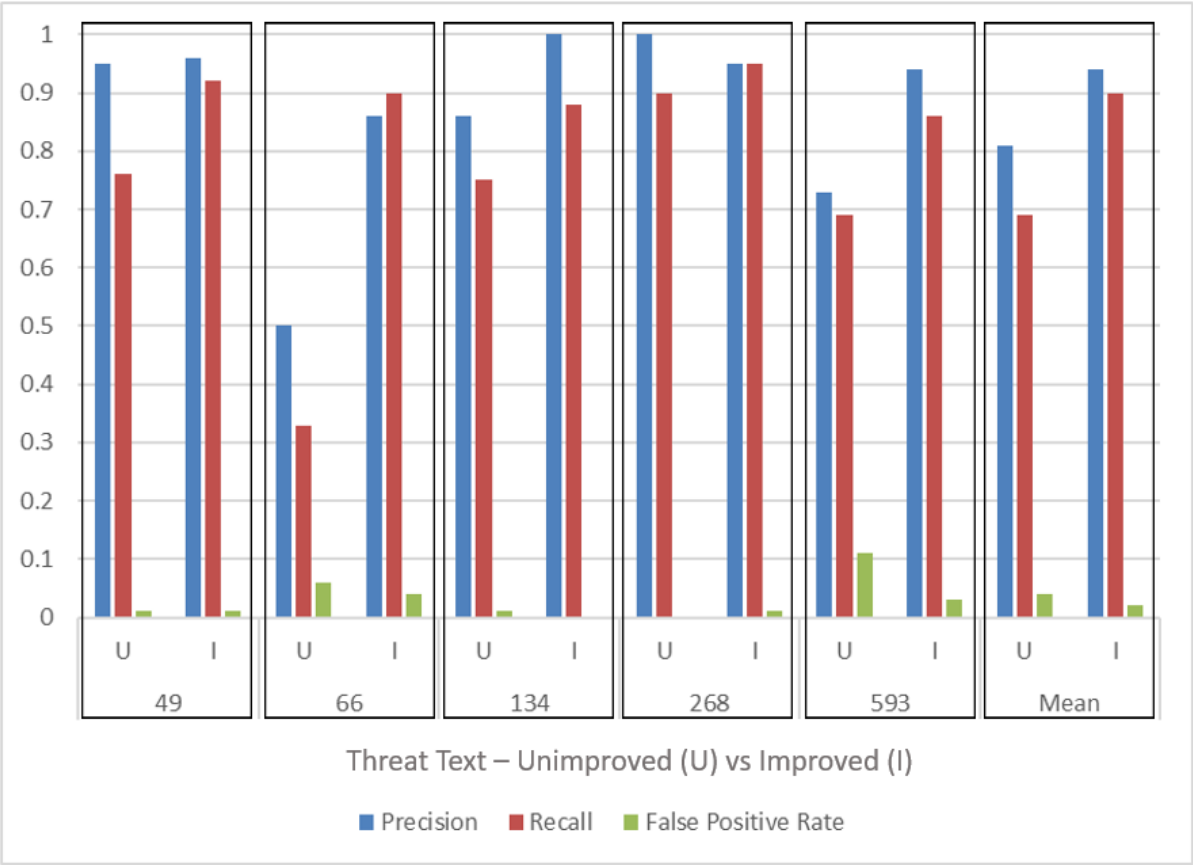

Figure 5. Unimproved vs Improved Text Comparison for 5 Threats

Solution Architecture and Use Cases

For the artifact described above and evaluated below to be truly useful to the CSE in the context of mitigation selection for cyber risk assessment, it must be incorporated into a system

with which the CSE can interact. Appendix B describes the common use cases for the CSE's usage of such a system and Appendix C describes the data model and a high-level architecture of such a system as illustrated in Figure 6. The architecture has been designed modularly and using object-oriented principles so that any of the threat-mitigation matching techniques investigated in this chapter could be incorporated. Some key characteristics of the system include the following: (a) provides for models to be saved and reused to label additional mitigations (b) persists the threats, mitigations, and known matches in a data store for reuse, (c) is extensible to additional threats, and (d) provides a means for the CSE to view, augment, and utilize the data.

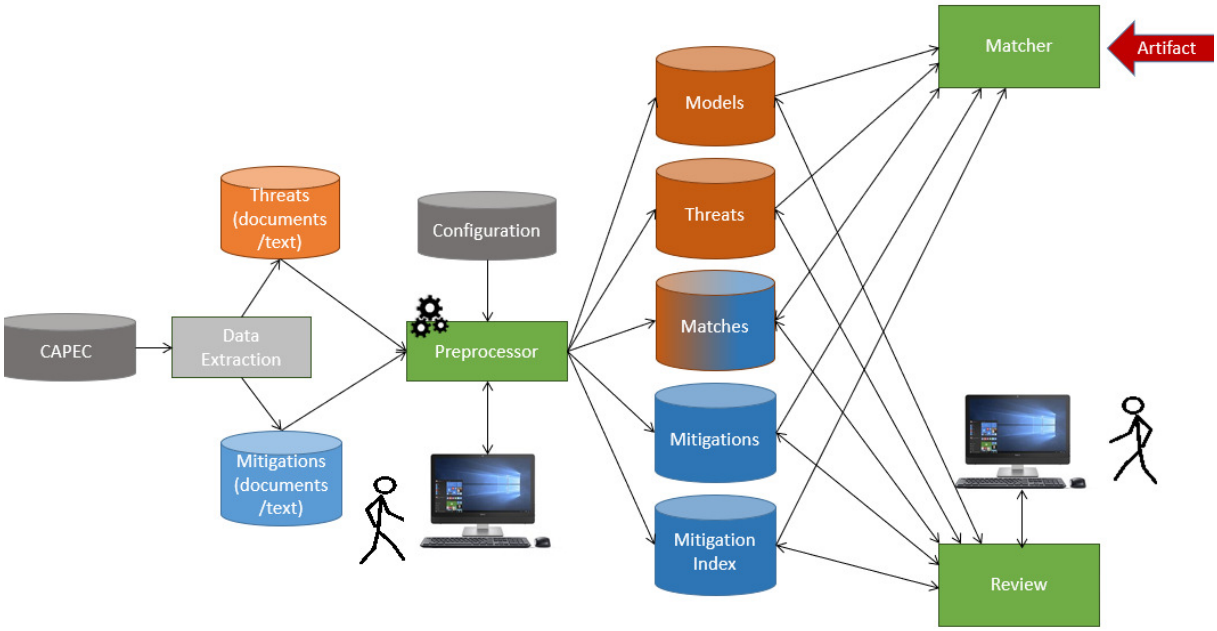

Figure 6. Solution Architecture

Demonstration and Evaluation

Demonstration and evaluation work together to show that the artifact effectively solves the problem. Hevner and March (Hevner & March, 2004) state a number of rigorous evaluation methods, classifying them into the following categories: observational (e.g. case or field study), analytical (e.g. quantitative comparisons, such as of time or cost), experiment or simulation, testing, and descriptive (e.g. argument or scenarios). In the present research, we demonstrate and evaluate the artifact by applying it to predict the labels for new mitigation documents that were held aside and not used for training. The test data set consists of 276 documents, 261 of

which were extracted from the CAPEC mitigations for threats other than 49, 66, 134, 266, and 593, and 15 of which were drawn from the Internet, 3 new relevant mitigations for each of the 5 test threats. We discuss the evaluation of the artifact in the next few paragraphs by revisiting each solution objective stated in the Research Methodology section. Quantitative machine learning and IR performance metrics are shown in Figure 7 and Table 3. The evaluation conclusions are summarized in Table 4.

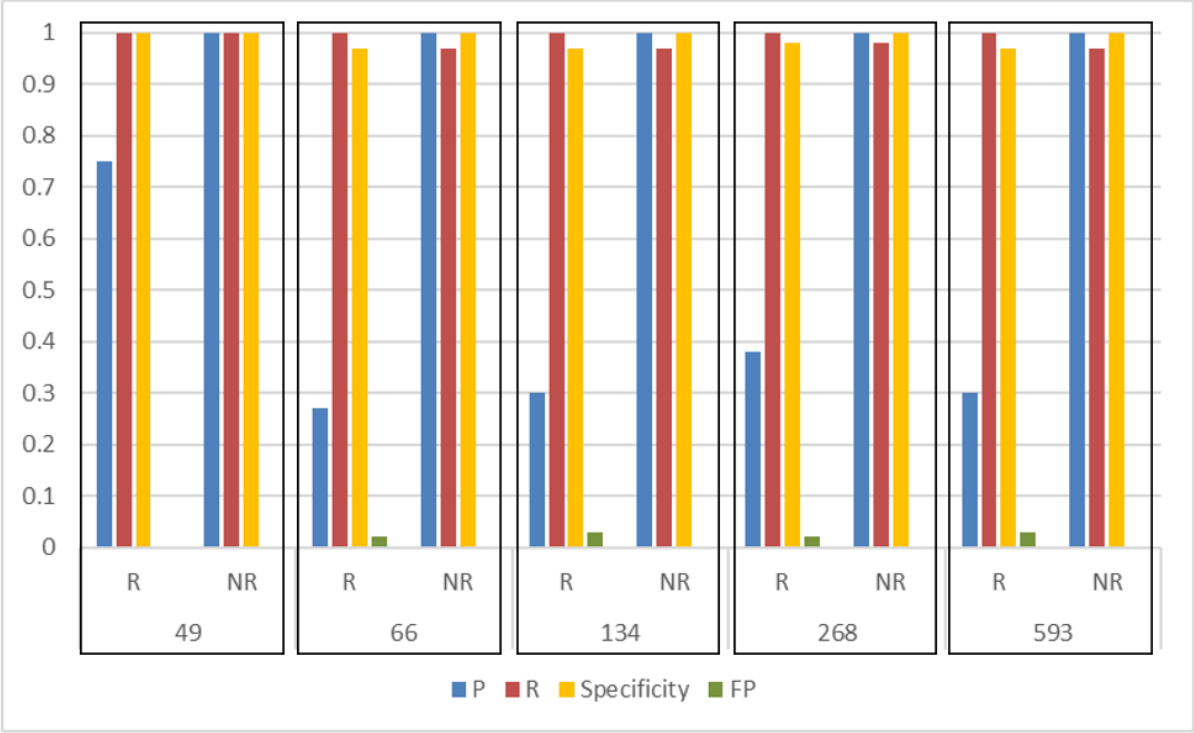

Figure 7. Test Results – Improved Text

Objective: Match most of the relevant mitigations for a given threat while avoiding selection of non-relevant mitigations. This is one of the most important objectives as the others are only germane after the artifact achieves satisfactory matching performance. As mentioned in the design section, we experimented with several artifact designs to see which obtained the best performance. Thus, we needed some objective measures for comparison. Following medical SRs research, we measured recall, precision, false positive rate, specificity, and the number of instances correct and incorrectly labeled to evaluate performance of the artifact. We applied cross-validation, using the 10-fold approach to obtain these measures during the training stage (Altman & Bland, 1994; Bekhuis & Demner-Fushman, 2012; Bekhuis et al., 2014; Cohen, 2008; Cohen et al., 2006; García, Mollineda, & Sánchez, 2014;

Jonnalagadda & Petitti, 2014; Liu et al., 2016; Matwin et al., 2010; Mo et al., 2015; Su, 1992; Timsina et al., 2016).

During design, we used the cross-validation statistics output during training to compare the model designs, deciding which to advance or leave behind. Although suitable for comparing models, these measures are not definitive for new document instances. During the evaluation stage, we re-evaluated the classifiers on test data held aside and not used during training as is customary in machine learning evaluation. We computed the recall, precision, false positive rate, and specificity by comparing the predicted and actual labels for the test instances. Figure 7 and Table 3 show the test results on the improved text for five threats. These results are discussed in more detail in the next few paragraphs.

Recall that training measures yielded precision and recall > 0.93 for the NR class. This foreshadowed excellent discernment of the NR class. Although we are most interested in the R class, the model's ability to discriminate NR instances is also a benefit. Test results for precision and recall on the NR class lived up to the promises made by the training statistics. In addition, all five models had high specificity (97-100%) meaning at least 97% labor savings for the CSE when compared to totally manually matching efforts because the models are very good at accurately discarding non-relevant documents.

For the R class, training measures yielded precision between 0.86 and 1.00 (mean 0.94), recall between 0.86 and 0.95 (mean 0.90), and FP rate between 0 and 4% (mean 2%). During testing, the models all yielded 1.00 recall on the R class, performing better than anticipated based on cross-validation statistics. This means each of the models excels at recognizing relevant mitigations for its designated threat and thus we are not likely to ignore relevant mitigations. Unfortunately, precision during testing was lower than anticipated (between 0.27 and 0.75, mean 0.40). For the 266 test instances, there were between 0 and 8 false positives (0-3%) per threat. In practical usage, we can tolerate a few false positives in our approach but, similar to the process of medical SRs, we would have to have CSE review of the mitigations labeled as relevant as illustrated in the architecture in Figure 3 and Appendix C before recording them in a knowledge base as reusable recommendations.

Table 3. "Per Threat" Test Summary - Improved Text

Threat	Class	Precision	P@3	Recall	FP Rate	Specificity	# Correct	# Incorrect
49	R	0.75	1.00	1.00	0.00	1.00	3	0
	NR	1.00		1.00	0.00	1.00	272	1
66	R	0.27	1.00 (*)	1.00	0.02	0.97	3	0
	NR	1.00		0.97	0.00	1.00	265	8
134	R	0.30	1.00 (*)	1.00	0.03	0.97	3	0
	NR	1.00		0.97	0.00	1.00	266	7
268	R	0.38	1.00 (*)	1.00	0.02	0.98	3	0
	NR	1.00		0.98	0.00	1.00	268	5
593	R	0.30	1.00 (*)	1.00	0.03	0.97	3	0
	NR	1.00		0.97	0.00	1.00	266	7

Precision @ K, discussed in the Literature Review, is a measure of precision commonly applied in text retrieval applications. We considered P@K for all 5 models; we used K=3 because we knew in advance that our test data set contained exactly 3 relevant mitigations per threat. For threat 49, there were 4 positive predictions, 3 correct and 1 false positive. The correct predictions were ranked in the top 3, each with 1.0 probability and the false positive was ranked fourth at 0.51 probability. Thus, P@3 for threat 49 is 1.0. For threat 66, there were 11 positive predictions, 3 correct and 8 FPs. All 3 of the TPs were ranked at 1.0, but 5 FPs were also ranked at 1.0. This complicated the P@K calculation because any of the 8 items ranked 1.0 could be in the top 3. We found sparse treatment of tie-breaking for P@K in the literature. A simplistic but commonly accepted approach for dealing with ties from TREC[5] (National Institute of Standards and Technology, 2005) is to choose one of the possible orderings and evaluate P@K for it. One such ordering is for all the positive instances to be in the top 3 and, thus, P@3 would be 1.0. However, this is admittedly very optimistic (indicated with * in Table 3) as other arbitrary orderings of the results could yield appreciably different results for P@3, including 0.0, 0.33, and 0.67. McSherry and Najork proposed an alternative method for computing P@K which accounts for ties by averaging P@K over all the possible orderings (McSherry & Najork, 2008). There are 40,320 possible orderings for the 8 samples labeled positive for threat 66 and over half of them would contain 3 NR entries (P@3=0.0). These would drive the average down

[5] For more than 25 years, Text Retrieval Conference (TREC) has been a pre-eminent information retrieval conference supporting text retrieval research with large test corpi and uniform scoring procedures to facilitate comparison of results. It is sponsored by NIST. https://trec.nist.gov/

dramatically; thus, without implementing McSherry's measure, we estimated that it would be not be better than the value in the precision column. Similarly, for threat 134, there were 10 positive predictions, 9 of which were ranked at 1.0 including the 3 known positives; for threat 268, there were 6 positive predictions, 6 of which were ranked at 1.0 including the 3 known positives; and for threat 593, there were 10 positive predictions, 9 of which were ranked at 1.0 including the 3 known positives. The bottom line is P@K did not help with evaluation as much as we originally thought it would due to the ties.

We ultimately based evaluation of our artifact on recall, specificity, and false positive rate as shown in Figure 7 and Table 3. These measures are defined in Equations 1, 2, and 3. Recall is the probability that all relevant documents will be retrieved. Specificity is the probability that all non-relevant documents will be ruled out. False positive rate is the probability that a non-relevant document will be retrieved. As Powers points out, taken alone, precision and recall tend to understate a method's ability to correctly identify non-relevant instances (Powers, 2007). This ability is measured using specificity, and we think it is important for threat-mitigation mapping because ruling out true negatives can lead to substantial workload reduction for the CSE.

$$Recall = \frac{TP}{TP + FN} \quad \text{(Equation 1)}$$

$$Specificity = \frac{TN}{TN + FP} \quad \text{(Equation 2)}$$

$$FP\ Rate = \frac{FP}{FP + TN} \quad \text{(Equation 3)}$$

In summary, with recall of the R class registering 1.00 on test data for all 5 models, we can be confident that the model will not overlook relevant mitigations. This is desirable because we do not want to obscure any relevant mitigations from the CSE's view. With a false positive rate between 0 and 3% and specificity between 0.97 and 1.00, we are encouraged that the model will reliably eliminate instances that are not in the R class. Precision is lower than we desired and with this comes a few false positives. This shortfall can be mitigated in practice by providing for CSE screening of the recommended matches before they are committed to the knowledge base for reuse. The high precision (1.00), recall (>0.97), and specificity (1.00) of the NR class means the models will accurately eliminate most (>97%) of the NR instances

without any manual intervention, greatly reducing the CSE workload when compared to purely manual matching and leaving just a few false positives for the CSE to remediate. In a practical setting where the objective would be to build a reusable knowledge base of threat-mitigation mappings, this remediation activity would only have to be done for new matches.

Objective: Process existing English language text documents where each separately describes either a threat or a mitigation. The CAPEC dataset and the additional example mitigations are English language documents. During the training and testing of each trial design, we demonstrated that the artifact accepts these English language text documents about threats and mitigations.

Objective: Provide an automated method for recommending (matching) relevant mitigations when presented with a threat. During training and testing, we demonstrated that the artifact will label mitigations as relevant or not relevant to a given threat.

Objective: Accommodate (be extensible to) new and evolving threats and mitigations. During testing we demonstrated that the artifact can accept new mitigations which it will label as relevant or not relevant on a "per threat" basis using a stored model trained from labeled data. The method can also accept new threats with the caveat that labeled data consisting of known relevant mitigations for the threat will have to be created so that a model can be trained.

Objective: Provide utility to cybersecurity experts in mitigation selection. Merriam-Webster equates utility with usefulness and "practical worth or applicability" ("Usefulness," 2019). Hevner et al. emphasize that "the artifact works and does what it is meant to do…achieving its goals." (Gregor & Hevner, 2013) Finally, according to Raghavan et al. the "usefulness of a retrieval system is determined to a great extent by how closely it can characterize the dichotomy" of relevant vs non-relevant documents for its intended purpose (Raghavan et al., 1989). We use these definitions to assert a reasoned argument for utility of the artifact. We have shown that the artifact meets the objectives we set forth at the beginning of the research in 5 test cases, and especially that it matches most of the relevant mitigations for a given threat while ruling out at least 97% of the non-relevant mitigations. These results are favorable for utility, but we leave formal utility assessment to future work after the artifact has been operationalized into a system such as the one in Figure 6.

Objective: Be able to be used in a system that allows for reuse of the artifact and the matches produced by the artifact. The solution produces models that can be saved and reused. As described in the architecture section, if the artifact were to be operationalized in an architecture such as the one in Figure 6, the system could provide for models to be saved and reused to label new mitigations as they are encountered. The threat documents, mitigation documents, and labeled matches between the two could be persisted in a data store so that they can be reused to satisfy threat queries by the CSE. A user interface could allow the documents and matches to be viewed and utilized.

Evaluation Summary. In Chapter 1 we motivated the problem of matching mitigations to cyber threats and in Chapter 3 we set forth objectives for a solution to that hard problem. We evaluated the artifact against those objectives and showed that it achieves its goals. Table 4 summarizes the artifact evaluation based on the solution objectives stated in the Research Methodology section above. To show practical worth and applicability, we provided use cases and an architecture into which the artifact can be integrated for practical use by cybersecurity professionals engaged in cyber risk assessment. In particular, we produced a method for automatically matching mitigations to threats that is both extensible and reusable and that will match most of the relevant mitigations for a given threat while avoiding selection of non-relevant mitigations. Moreover, five instantiations of the method accurately eliminated most (>97%) of the non-relevant mitigations without any manual intervention, leaving just a few false positives for the CSE to remediate manually. This robust discrimination of the R and NR classes aligns with Raghavan's definition of usefulness for retrieval systems (Raghavan et al., 1989).

Table 4. Evaluation Results Based on Solution Objectives

Objective	Evaluation
Process existing English language text documents where each separately describes either a threat or a mitigation	**Pass.** By testing, we demonstrated that the artifact accepts English language text documents about threats and mitigations.
Provide an automated method for recommending (matching) relevant mitigations when presented with a threat	**Pass.** By testing, we demonstrated that the artifact proposes matching mitigations when a threat is given.

Objective	Evaluation
Match most of the relevant mitigations for a given threat while avoiding selection of non-relevant mitigations	**Pass.** By evaluation of the models on test data, we demonstrated that the models can eliminate about 97% of non-relevant mitigations. Moreover, with recall at 1.00, it will not overlook relevant mitigations.
Accommodate (be extensible to) new and evolving threats and mitigations	**Pass.** The artifact can accept new mitigations which it will match to existing threats using a stored model trained from labeled data. The method can also accept new threats with the caveat that labeled data consisting of known relevant mitigations for the threat would have to be created so that a model can be trained.
Provide utility to cybersecurity experts in mitigation selection	**Pass.** By satisfying the preceding objectives, the artifact as instantiated provides practical value and to the CSE engaged in cyber risk assessment and meets the utility criteria for retrieval systems established by (Raghavan et al., 1989). It has potential to reduce CSE workload by about 97% over purely manual matching.
Be able to be used in a system that allows for reuse of the artifact and the matches produced by the artifact	**Pass.** The artifact provides models that can be saved and reused to label additional mitigations at a later time. The artifact could be used in a system such as the one shown in Figure 6 where the threats, mitigations, and matches could be persisted in a data store and a user interface could be provided to allow this data to be viewed and reused.

Validity

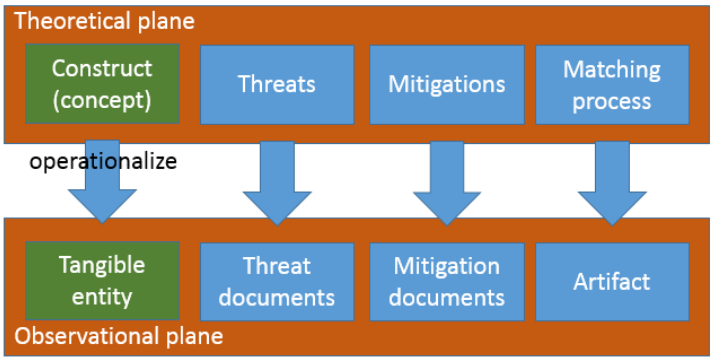

Figure 8. Validity

Validity centers on interactions between the theoretical and observational research planes (Trochim & Donnelly, 2006) as illustrated in Figure 8. Constructs in the theoretical plane are intangible. In our research, the major constructs are cyber threats, mitigations, and a cognitive process that matches appropriate mitigations to threats. We operationalized these intangible ideas in the observational plane order to conduct the research. In our case, as shown in Figure 8, we operationalized threats and mitigations as textual documents describing instances of each of the corresponding constructs and we operationalized the cognitive matching process in our artifact. In the context of DSRM, Hevner et al. mention validity in the context of artifact evaluation stating that "validity means that the artifact works and does what it is meant to do; that it is dependable in operational terms in achieving its goals." (Gregor & Hevner, 2013) The types of validity commonly discussed in scholarly research include, face validity, construct validity, internal validity, and external validity. In addition, Lukyanenko et al. recently put forth the concept of instantiation validity specifically for Design Science (Lukyanenko & Parsons, 2014).

Face validity is a subjective assessment about whether the operationalization of the research constructs make sense when taken at face value. We argue for face validity of our operationalizations of the threat and mitigation constructs on the basis that the textual documents represent the traditional method by which such knowledge is codified. Likewise, the artifact parallels the cognitive matching process that the experts perform in their brains.

Construct validity is "the degree to which inferences can legitimately be made from the operationalizations in a study to the theoretical constructs on which those

operationalizations were based" - in other words that the operationalizations are reasonable indicators of the underlying latent concepts (Trochim & Donnelly, 2006). While there are tests for construct validity in quantitative research (e.g., convergent and discriminant validity), the picture is less clear for Design Science. One major threat to construct validity is failure to properly understand and explain the constructs before operationalizing them. We have addressed this threat via our Literature Review of the pertinent content domains.

Trochim defines **internal validity** as "the approximate truth about inferences regarding cause-effect or causal relationships" and furthermore asserts that "internal validity is only relevant in studies that try to establish a causal relationship." (Trochim & Donnelly, 2006) Our research does not seek to establish a causal relationship; therefore, internal validity is mentioned here for completeness but is not pertinent to our research since we are not trying to establish causality.

External validity "is the degree to which the conclusions in the study would hold for other persons in other places and at other times" also referred to as generalizability (Trochim & Donnelly, 2006). The generalizability we seek is that our method applies to more threats than the 5 we tested here. We see initial indications of generality from similarities in the 5 test cases; however, the method must be applied to additional and more diverse sources of threats and mitigations before we can be sure.

Instantiation validity is an assessment of how well an artifact created via Design Science Research instantiates constructs of the theory on which the artifact is based (Lukyanenko & Parsons, 2014). We have addressed and promoted instantiation validity in our research by appropriate alignment of the artifact with literature in the problem domains per the Literature Review and by developing the artifact using a rigorous approach as described in the Design and Development of the Artifact section.

Communication

Communication of research results to both practitioner and scholarly audiences is a key tenet of Design Science Research. Via a combination of UML drawings and prose, sufficiently detailed design documentation has been created to convey the construction details of the artifact. Per Hevner, this enables "practitioners to take advantage of the benefits" (Hevner &

March, 2004) while also promoting critical feedback and opportunities for extension by the research community. This dissertation satisfies the communication requirement of the DSRM.

CHAPTER 5

CONCLUSIONS

In this research, we set out to devise a method for matching mitigations to cyber threats expressed as English language text documents using machine learning and text retrieval techniques in support of cyber risk assessment. In the preceding chapters, we have discussed an iterative process framed within the Design Science Research Method where we evaluated and down-selected designs by comparing their respective measures of performance. We ultimately arrived at a matching method that achieves the stated objectives and we instantiated 5 examples as SVM "per threat" classifiers based on LSA features. We rigorously evaluated the instantiations in 5 test cases and were encouraged by the results. We illustrated the utility of the method by describing an architecture into which it can be integrated for practical use. Overall, we are encouraged by the results achieved thus far.

Contributions

Mitigation selection to remediate cyber threats has heretofore been primarily a manual process done by human experts using textual sources which are extensive and disparate. Reliance solely on human experts brings issues of scalability, consistency, and repeatability. The ongoing shortage of cybersecurity experts combined with a burgeoning cyber threat landscape compelled us to look for a way to improve this situation.

This research contributes to theory by taking steps towards a novel machine learning method for automatically mapping mitigations to threats, both expressed as English language text, and demonstrating instantiations of the method. Moreover, the research fills a research gap in the cyber risk assessment literature by providing a semi-automated method to produce a starting list of possible mitigations for threats identified during risk assessment providing the data needed to flow into mitigation optimization techniques. The method is extensible to accommodate the continued evolution of both cyber threats and mitigations, an important consideration in light of the dynamic cyber landscape. We have also demonstrated one way to

improve the textual descriptions of threats and mitigations to better support automated matching.

From a practical perspective, our method for matching mitigations to threats benefits all threat-informed cyber risk assessment approaches by providing a means to recommend relevant mitigations to remediate specific threats thereby aiding decision-making for IS stakeholders and cybersecurity experts. This is important because under-mitigating the actual threats provides a false sense of security while over-mitigating is costly and wasteful. When operationalized into a knowledge base, such as the one shown in Figure 6, where models and matches can be saved for reuse, the method may make mitigation selection more repeatable, facilitate knowledge reuse, save CSE time and labor, and extend the reach of cybersecurity experts who are currently in short supply. The list of mitigations applicable to each threat can serve as input into analyses of alternatives, enabling practitioners to leverage a large body of mitigation optimization research. Finally, the method can respond to the evolutionary nature of cyber threats and mitigations. Thus, it may improve overall security of cyber systems when used as part of a risk assessment and mitigation cycle such as the one shown in Figure 1 by making more frequent reassessments of cyber systems feasible.

Lessons Learned from the Text

In Chapter 4 Analysis of the Text, we identified that improving the mitigation texts to include an explanation of how each one addresses the threat would improve the match results by reducing the FNs. During the research, we noted domain-specific peculiarities in the documents. A number of issues are known to affect text-based processing in general, including synonymy, polysemy, misspellings, colloquialisms, and the use of acronyms and jargon. The cyber threat-mitigation matching problem suffered from all of these issues and, in addition, varying styles (e.g. prose versus bullets), varying degrees of brevity and verbosity, extraneous information (e.g. “this may be prohibitively expensive”), and expressions in the negative (i.e. what not to do). References to product names and technical standards sometimes served as short-hand, obscuring complex concepts. We also encountered considerable sameness in the language used to express different threats (e.g., SQL injection, email injection, script injection as well as some mitigations which apply to multiple threats (e.g., multifactor authentication, encryption, training). These conditions worked against discernment of relevance. Data

imbalance favoring the non-relevant class, limited matching mitigations per threat, and erroneous mappings presented tactical issues for classifier training.

Future Work

For this initial proof of concept research, we bounded the scope, providing ample opportunities for **incremental improvements**. The method we developed was instantiated and tested with English language documents. It would be interesting to extend it to other languages. Likewise, our instantiations were based on a narrow slice of cybersecurity documents. The method could be improved by exposure to more threat and mitigation sources. We made no effort to address redundant threats and mitigations in our corpus. In order to ingest documents from additional sources, the method should be preceded by an automated approach for dealing with duplication. In addition, analyses of the structure and semantics of threat and mitigation documents from various sources could lead to discovery of additional ways to improve the document content and by extension the matching method.

We used supervised machine learning which required some pre-existing matches. This work could be extended by investigating semi-supervised learning classification techniques to build classifiers for new threats where labeled data does not yet exist. Moreover, it is possible that semi-supervised learning could also be used to improve the classifiers initially trained for existing threats by taking into account new matches that come about as new mitigation documents are added.

We focused our research on defensive cybersecurity, identifying threats and seeking to determine relevant mitigations. It is possible that our method may be applicable or extensible to "white hat" offensive cybersecurity, such as to better understand attacker behavior or residual exposure. This perspective is characterized by identifying the mitigations present in a system and seeking to determine threats to counter them. Moreover, while we established a degree of utility for our method by demonstrating that the artifact solves the problem for 5 examples, survey research to investigate the perceived utility by actual CSEs would be beneficial.

Finally, we identify several **lofty goals** for future extensions of this research. Improving the ways that threat and mitigation text is written, such as by addressing the limitations described in the Lessons Learned section, could improve the method. Furthermore, devising a robust ontology to capture the intricacy of threat/mitigation relationships would offer great

potential to improve the matches, helping to tease out complexities such as overlapping threats and one to many mitigation-threat mappings. This structure could be used as metadata to improve the matching models. In the long term, we envision the matcher as a component of an overarching architecture with a reusable, continually evolving, peer-reviewed knowledge base of threat-mitigation mappings with contributions coming from many sources, including threat frameworks, mitigation catalogs and vendor literature.

REFERENCES

Altman, D. G., & Bland, J. M. (1994). Diagnostic tests 1: sensitivity and specificity. *BMJ*, *308*(June), 1552.

Apache Foundation. (2013). Apache Lucene Scoring. Retrieved January 15, 2019, from https://lucene.apache.org/core/3_5_0/scoring.html

Apache Foundation. (2018). Apache Lucene. Retrieved March 24, 2018, from https://lucene.apache.org/

Aphinyanaphongs, Y., & Aliferis, C. F. (2003). Text categorization models for retrieval of high quality articles in internal medicine. *AMIA ... Annual Symposium Proceedings / AMIA Symposium. AMIA Symposium*, 31–35.

Bañez, L. L. E., Berliner, E., Erinoff, L. I. S. J., Lege-Matsuura, L. I. S. S., Potter, M. L. I. S. S., & Uhl, S. (2016). EPC Methods: An Exploration of the Use of Text- Mining Software in Systematic Reviews.

Barnard, L., & von Solms, R. (2000). A Formalized Approach to the Effective Selection and Evaluation of Information Security Controls. *Computers & Security*, *19*(2), 185–194.

Bekhuis, T., & Demner-Fushman, D. (2012). Screening nonrandomized studies for medical systematic reviews: A comparative study of classifiers. *Artificial Intelligence in Medicine*, *55*(3), 197–207.

Bekhuis, T., Tseytlin, E., Mitchell, K. J., & Demner-Fushman, D. (2014). Feature engineering and a proposed decision-support system for systematic reviewers of medical evidence. *PLoS ONE*, *9*(1), 1–10.

Bolger, F., & Wright, G. (1994). Assessing the Quality of Expert Judgment - Issues and Analysis. *Decision Support Systems*, *11*(1), 1. Retrieved from isi:A1994MP91400001

Booch, G., Rumbaugh, J., & Jacobson, I. (2000). *The Unified Modeling Language User Guide* (6th ed.). Addison-Wesley.

Bradford, R. B. (2008). An empirical study of required dimensionality for large-scale latent semantic indexing applications. *Proceeding of the 17th ACM Conference on Information and Knowledge Mining - CIKM '08*, 153. Retrieved from http://portal.acm.org/citation.cfm?doid=1458082.1458105

Breier, J., & Hudec, L. (2013). On selecting critical security controls. *Proceedings - 2013*

International Conference on Availability, Reliability and Security, ARES 2013, *7799*, 582–588.

Calfas, J. (2018). T-Mobile Says a Data Breach Is Affecting Millions of Its Customers. Here's What You Need to Know. Retrieved August 31, 2018, from http://time.com/money/5377773/tmobile-data-breach-august-2018/

Caralli, R. A., Stevens, J. F., Young, L. R., & Wilson, W. R. (2007). *Introducing OCTAVE Allegro : Improving the Information Security Risk Assessment Process*. Retrieved from https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=8419

Cebula, J. J., Popeck, M. E., & Young, L. R. (2014). A Taxonomy of Operational Cyber Security Risks Version 2. *Carnegie-Mellon Univ Software Engineering Inst*, (May), 1–47. Retrieved from http://www.sei.cmu.edu

Center for Cyber Safety and Education. (2017). The 2017 Global Information Security Workforce Study: Benchmarking Workforce Capacity and Response to Cyber Risk.

Center for Strategic and International Studies. (2014). *Net Losses : Estimating the Global Cost of Cybercrime*. Retrieved from http://www.mcafee.com/us/resources/reports/rp-economic-impact-cybercrime2.pdf

Centers for Medicare and Medicaid Services. (2007). *HIPPA Security Standards: Technical Safeguards*. *HIPAA Security Series* (Vol. 2). Retrieved from http://www.hhs.gov/ocr/privacy/hipaa/administrative/securityrule/techsafeguards.pdf

Chandrasekar, K., Cleary, G., Cox, O., Lau, H., Nahorney, B., Gorman, B. O., … Wueest, C. (2017). *Symantec Internet Security Threat Report*. Retrieved from https://digitalhubshare.symantec.com/content/dam/Atlantis/campaigns-and-launches/FY17/Threat Protection/ISTR22_Main-FINAL-JUN8.pdf?aid=elq_

Chang, C.-C., & Lin, C.-J. (2018). LIBSVM -- A Library for Support Vector Machines. Retrieved February 22, 2019, from https://www.csie.ntu.edu.tw/~cjlin/libsvm/

Chen, P. (1976). The entity-relationship model---toward a unified view of data. *ACM Transactions on Database Systems*, *1*(1), 9–36.

Cisco Systems. (2018). Cisco 2018 Annual Cybersecurity Report. *Cisco [Online]*, 675-. Retrieved from https://www.cisco.com/c/dam/m/hu_hu/campaigns/security-hub/pdf/acr-2018.pdf

Cohen, A. M. (2008). Optimizing feature representation for automated systematic review

work prioritization. *AMIA ... Annual Symposium Proceedings / AMIA Symposium. AMIA Symposium*, 121–5. Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2656096&tool=pmcentrez&rendertype=abstract

Cohen, A. M., Hersh, W. R., Peterson, K., & Yen, P. Y. (2006). Reducing workload in systematic review preparation using automated citation classification. *Journal of the American Medical Informatics Association*, *13*(2), 206–219.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407. Retrieved from http://www.psychology.uwo.ca/faculty/harshman/latentsa.pdf

El-Gayar, O. F., & Fritz, B. D. (2010). A web-based multi-perspective decision support system for information security planning. *Decision Support Systems*, *50*(1), 43–54.

Enclave Security. (2015). Open Threat Taxonomy. Retrieved from https://www.auditscripts.com/free-resources/open-threat-taxonomy/

Equifax. (2017). Equifax Releases Details on Cybersecurity Incident, Announces Personnel Changes. Retrieved October 13, 2017, from https://www.equifaxsecurity2017.com/2017/09/15/equifax-releases-details-cybersecurity-incident-announces-personnel-changes/

European Union Agency For Network And Information Security. (2016). *Threat taxonomy: A tool for structuring threat information. Initial report.* Retrieved from https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/enisa-threat-landscape/etl2015/enisa-threat-taxonomy-a-tool-for-structuring-threat-information

Fedorowicz, J. (1996). *Document-based Decision Support*. (R. Sprague Jr. & H. J. Watson, Eds.), *Decision Support for Management*. Upper Saddle River, N.J.: Prentice-Hall.

Fenz, S., Ekelhart, A., & Neubauer, T. (2011). Information security risk management: In which security solutions is it worth investing? *Communications of the Association for Information Systems*, *28*(1), 329–356.

Fielder, A., Panaousis, E., Malacaria, P., Hankin, C., & Smeraldi, F. (2016). Decision support approaches for cyber security investment. *Decision Support Systems*, *86*, 13–23.

Foltz, P. W. (1990). Using latent semantic indexing for information filtering. *Proceedings of*

the ACM SIGOIS and IEEE CS TC-OA Conference on Office Information Systems (COCS '90), *11*(2–3), 40–47. Retrieved from http://portal.acm.org/citation.cfm?id=91486

Frunza, O., Inkpen, D., & Matwin, S. (2010). Building systematic reviews using automatic text classification techniques. *COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, (August), 303–311. Retrieved from http://dl.acm.org/citation.cfm?id=1944601

García Adeva, J. J., Pikatza Atxa, J. M., Ubeda Carrillo, M., & Ansuategi Zengotitabengoa, E. (2014). Automatic text classification to support systematic reviews in medicine. *Expert Systems with Applications*, *41*(4 PART 1), 1498–1508.

García, V., Mollineda, R. A., & Sánchez, J. S. (2014). A bias correction function for classification performance assessment in two-class imbalanced problems. *Knowledge-Based Systems*, *59*, 66–74.

Gee, K. R. (2003). Using latent semantic indexing to filter spam. *Proceedings of the 2003 ACM Symposium on Applied Computing - SAC '03*, 460. Retrieved from http://portal.acm.org/citation.cfm?doid=952532.952623

Goldrich, L., Hamer, S., McNeil, M., Longstaff, T., Gatlin, R., & Bello-Ogunu, E. (2014). REQcollect: Requirements collection, project matching and technology transition. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 4887–4894.

Gosler, J., & Von Thaer, L. (2013). *Resilient Military Systems and the Advanced Cyber Threat*. Retrieved from http://www.acq.osd.mil/dsb/reports/ResilientMilitarySystems.CyberThreat.pdf

Gregor, S., & Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly*, *37*(2), 337–355.

Gupta, M., Rees, J., Chaturvedi, A., & Chi, J. (2006). Matching information security vulnerabilities to organizational security profiles: A genetic algorithm approach. *Decision Support Systems*, *41*(3), 592–603.

Hallberg, J., Bengtsson, J., Hallberg, N., Karlzén, H., & Sommestad, T. (2017). The Significance of Information Security Risk Assessments Exploring the Consensus of Raters' Perceptions of Probability and Severity. *International Conference on Security*

and Management, 131–137.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

Hevner, A. R., & Chatterjee, S. (2010). *Design Research in Information Systems Theory and Practice*. (R. Sharda, Ed.), *Integrated Series in Information Systems Volume 22*. New York, New York, USA: Springer. https://doi.org/10.1007/978-1-4419-5653-8

Hevner, A. R., & March, S. T. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–105.

Holm, H., Sommestad, T., Ekstedt, M., & Honeth, N. (2014). Indicators of expert judgement and their significance: An empirical investigation in the area of cyber security. *Expert Systems*, *31*(4), 299–318.

Howard, B. E., Phillips, J., Miller, K., Tandon, A., Mav, D., Shah, M. R., … Thayer, K. (2016). SWIFT-Review: A text-mining workbench for systematic review. *Systematic Reviews*, *5*(1), 1–16. https://doi.org/10.1186/s13643-016-0263-z

ISACA. (2009). *The Risk IT Framework*. Retrieved from www.isaca.org

Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proceeding ECML'98 Proceedings of the 10th European Conference on Machine Learning* (pp. 137–142).

Jonnalagadda, S. R., & Petitti, D. (2014). A new iterative method to reduce workload in the systematic review process. *Int J Comput Biol Drug Des*, *6*(0), 5–17.

Kaplan, S., & Garrick, B. J. (1981). On the quantitative definition of risk. *Risk Analysis*, *1*(1), 11–27.

Kauflin, J. (2017). The Fast-Growing Job With A Huge Skills Gap: Cyber Security. *Forbes*. Retrieved from https://www.forbes.com/sites/jeffkauflin/2017/03/16/the-fast-growing-job-with-a-huge-skills-gap-cyber-security/#7c14f80f5163

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*(1–2), 85–117.

Kiesling, E., Strauß, C., & Stummer, C. (2012). A multi-objective decision support framework for simulation-based security control selection. *Proceedings - 2012 7th*

International Conference on Availability, Reliability and Security, ARES 2012, 454–462.

Kontonatsios, G., Brockmeier, A. J., Przybyła, P., McNaught, J., Mu, T., Goulermas, J. Y., & Ananiadou, S. (2017). A semi-supervised approach using label propagation to support citation screening. *Journal of Biomedical Informatics*, *72*, 67–76.

Launius, S. (2018). Evaluation of Comprehensive Taxonomies for Information Technology Threats. *SANS Information Security Reading Room*. SANS. Retrieved from https://www.sans.org/reading-room/whitepapers/threatintelligence/evaluation-comprehensive-taxonomies-information-technology-threats-38360

Libicki, M., Senty, D., & Pollak, J. (2014). The Economics of the Cybersecurity Labor Market. In *Hackers Wanted: An Examination of the Cybersecurity Labor Market*. RAND Corporation.

Liu, J., Timsina, P., & El-Gayar, O. (2016). A comparative analysis of semi-supervised learning: The case of article selection for medical systematic reviews. *Information Systems Frontiers*, 1–13. Retrieved from http://dx.doi.org/10.1007/s10796-016-9724-0

Llanso, T. (2012). CIAM: A Data-driven Approach for Selecting and Prioritizing Security Controls. *IEEE International Systems Conference SysCon 2012*, *March*, 1–8.

Llanso, T., Hamilton, P. A., & Silberglitt, M. (2012). MAAP : Mission Assurance Analytics Platform. In *IEEE Conference on Technologies for Homeland Security (HST)*.

Llanso, T., McNeil, M., Pearson, D., & Moore, G. (2017). An Analytic Framework for Mission-Cyber Risk Assessment and Mitigation Recommendation. *Hawaii International Conference on System Sciences*, 10.

Llansó, T., McNeil, M. W., & Noteboom, C. (2019). Multi-Criteria Selection of Capability-Based Cybersecurity Solutions. *Hawaii International Conference on System Sciences*.

Llanso, T., Tally, G., Silberglitt, M., & Anderson, T. (2013). Mission-Based Analysis For Assessing Cyber Risk In Critical Infrastructure Systems. In J. Butts & S. Shenoi (Eds.), *International Federation for Information Processing (IFIP) - Critical Infrastructure Protection VII* (Vol. VII, pp. 135–148). Springer.

Lowe, H., & Barnett, O. (1994). Understanding and using the MeSH to perform literature searches. *JAMA*, *271*(14), 1103–1108.

Lukyanenko, R., & Parsons, J. (2014). Instantiation Validity in IS Design Research. In *DESRIST* (pp. 241–256). Retrieved from

http://www.scopus.com/inward/record.url?eid=2-s2.0-84901338298&partnerID=tZOtx3y1

Lv, J.-J., Zhou, Y.-S., & Wang, Y.-Z. (2011). A Multi-criteria Evaluation Method of Information Security Controls. *2011 Fourth International Joint Conference on Computational Sciences and Optimization*, 190–194.

Manning, C. D., Raghavan, P., & Schutze, H. (2009). *An Introduction to Information Retrieval* (Online edi). Cambridge University Press. Retrieved from http://dspace.cusat.ac.in/dspace/handle/123456789/2538

Matwin, S., Kouznetsov, A., Inkpen, D., Frunza, O., & O'Blenis, P. (2010). A new algorithm for reducing the workload of experts in performing systematic reviews. *Journal of the American Medical Informatics Association*, *17*(4), 446–453.

McSherry, F., & Najork, M. (2008). Computing Information Retrieval Performance Measures Efficiently in the Presence of Tied Scores. *ECIR 2008*, 414–421.

Microsoft. (2016). *Microsoft Security Intelligence Report* (Vol. 21). Retrieved from https://www.microsoft.com/security/sir/story/default.aspx#!10year_timeline

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39–41. Retrieved from http://portal.acm.org/citation.cfm?doid=219717.219748

MITRE. (2015). *An Overview of MITRE Cyber Situational Awareness Solutions*.

MITRE. (2017a). Common Attack Pattern Enumeration and Classification. Retrieved February 4, 2018, from https://capec.mitre.org/index.html

MITRE. (2017b). *Common Attack Pattern Enumeration and Classification - About CAPEC*. Retrieved from https://capec.mitre.org/about/

MITRE. (2017c). Cyber Risk Remediation Analysis. Retrieved October 8, 2017, from https://www.mitre.org/publications/systems-engineering-guide/enterprise-engineering/systems-engineering-for-mission-assurance/cyber-risk-remediation-analysis

Miwa, M., Thomas, J., O'Mara-Eves, A., & Ananiadou, S. (2014). Reducing systematic review workload through certainty-based screening. *Journal of Biomedical Informatics*, *51*, 242–253. Retrieved from http://dx.doi.org/10.1016/j.jbi.2014.06.005

Mo, Y., Kontonatsios, G., & Ananiadou, S. (2015). Supporting systematic reviews using LDA-based document representations. *Systematic Reviews*, *4*(1).

Nakamoto, Y. (2011). A Short Introduction to Learning to Rank. *IEICE Transactions on*

Information and Systems, *E94–D*(1), 1–2. Retrieved from http://joi.jlc.jst.go.jp/JST.JSTAGE/transinf/E94.D.1?from=CrossRef

National Institute of Standards and Technology. (2005). TREC 2005 Robust Track Guidelines. Retrieved February 24, 2019, from https://trec.nist.gov/data/robust/05/05.guidelines.html

National Institute of Standards and Technology. (2012). *National Institute of Standards and Technology Special Publication 800-30 R1: Guide for Conducting Risk Assessments*. Retrieved from http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf%5Cnhttp://csrc.nist.gov/publications/PubsSPs.html%5Cnhttp://dx.doi.org/10.6028/NIST.SP.800-30r1

National Institute of Standards and Technology. (2017). *NIST Special Publication 800-53 R5: Security and privacy controls for federal information systems and organizations*. https://doi.org/10.6028/NIST.SP.800-53r4

National Security Agency. (2018). NSA/CSS Technical Cyber Threat Framework v2. Retrieved from https://www.nsa.gov/Portals/70/documents/what-we-do/cybersecurity/professional-resources/ctr-nsa-css-technical-cyber-threat-framework.pdf

Otero, A. R. (2014). An Information Security Control Assessment Methodology for Organizations. *Nova Southeastern University. Retrieved from NSUWorks*, (266).

Panaousis, E., Fielder, A., Malacaria, P., Hankin, C., & Smeraldi, F. (2014). Cybersecurity Games and Investments: A Decision Support Approach. *Lecture Notes in Computer Science*, 266–286.

Patterson, I., Nutaro, J. J., Allgood, G., Kuruganti, P. T., & Fugate, D. (2013). Optimizing investments in cyber-security for critical infrastructure. In *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop* (p. 20).

PCI Security Standards Council. (2015). *PCI DSS Quick Reference Guide*. *PCI Security Standard*. Retrieved from https://www.pcisecuritystandards.org/security_standards/documents.php

Pedregosa, F., Weiss, R., & Brucher, M. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peffers, K., Tuunanen, T., Rothenberger, M. a., & Chatterjee, S. (2007). A Design Science

Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77.

Platt, J. (1998). Fast Training of Support Vector Machines using Sequential Minimal Optimization. In B. Schoelkopf, C. Burges, & A. Smola (Eds.), *Advances in Kernel Methods - Support Vector Learning*.

Powers, D. M. W. (2007). Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. School of Informatics and Engineering - Flinders University. Retrieved from http://www.flinders.edu.au/science_engineering/fms/School-CSEM/publications/tech_reps-research_artfcts/TRRA_2007.pdf

Princeton University. (2017). WordNet: A Lexical Database for English. Retrieved October 14, 2017, from https://wordnet.princeton.edu/

Raghavan, V. V, Jung, G. S., & Bollman, P. (1989). A Critical Investigation of Recall and Precision as Measures of Retrieval System Performance. *ACM Transactions on Information Systems*, *7*(3), 205–229.

Random House Inc. (n.d.). cyberattack. Retrieved January 6, 2018, from http://www.dictionary.com/browse/cyberattack

Rees, L. P., Deane, J. K., Rakes, T. R., & Baker, W. H. (2011). Decision support for Cybersecurity risk planning. *Decision Support Systems*, *51*(3), 493–505.

Rehurek, R. (2018). gensim Topic Modeling for Humans. Retrieved January 18, 2019, from https://radimrehurek.com/gensim/

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. (1994). Okapi at TREC-3. *Proceedings of 3rd Text REtrieval Conference*, 109–126. Retrieved from http://pdf.aminer.org/000/630/294/okapi_at_trec.pdf

Sarala, R., Zayaraz, G., & Vijayalakshmi, V. (2016). Optimal Selection of Security Countermeasures for Effective Information Security. *Proceedings of the International Conference on Soft Computing Systems*, *398*.

Sawik, T. (2013). Selection of optimal countermeasure portfolio in IT security planning. *Decision Support Systems*, *55*(1), 156–164.

Schilling, A., & Werners, B. (2016). Optimal selection of IT security safeguards from an existing knowledge base. *European Journal of Operational Research*, *248*(1), 318–327.

Schmittling, R. A. M. (2010). Performing a Security Risk Assessment. *ISACA Journal*, *1*. Retrieved from http://www.isaca.org/Journal/Past-Issues/2010/Volume-1/Pages/Performing-a-Security-Risk-Assessment1.aspx

Shapasand, M., Shajari, M., Golpaygani, S. A. H., & Ghavamipoor, H. (2015). A comprehensive security control selection model for inter-dependent organizational assets structure. *Information & Computer Security*, *23*(3), 302–316.

Shemilt, I., Simon, A., Hollands, G. J., Marteau, T. M., Ogilvie, D., O'Mara-Eves, A., … Thomas, J. (2014). Pinpointing needles in giant haystacks: Use of text mining to reduce impractical screening workload in extremely large scoping reviews. *Research Synthesis Methods*, *5*(1), 31–49.

Simmons, C. B., Shiva, S. G., Bedi, H., & Dasgupta, D. (2014). AVOIDIT: A Cyber Attack Taxonomy. In *9th Annual Symposium on Information Assurance (ASIA '14).*

Simon, H. (1997). *The sciences of the artificial, (third edition). Computers & Mathematics with Applications* (3rd ed., Vol. 33). Massachusetts Institute of Technology. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S0898122197829410

Singhal, A. (2001). Modern Information Retrieval: A Brief Overview. *Bulletin of the Ieee Computer Society Technical Committee on Data Engineering*, *24*(4), 1–9. Retrieved from http://160592857366.free.fr/joe/ebooks/ShareData/Modern Information Retrieval - A Brief Overview.pdf

Small, K., Wallace, B. C., Brodley, C. E., & Trikalinos, T. (2011). The Constrained Weight Space SVM: Learning with Ranked Features. *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 865–872.

Smeraldi, F., & Malacaria, P. (2014). How to spend it : optimal investment for cyber security Position paper. *Proceedings of the 1st International Workshop on Agents and CyberSecurity*, 1–4.

Su, L. T. (1992). Evaluation Measures for Interactive Information Retrieval. *Information Processing and ManagementEvaluation*, *28*(4), 503–516.

Swanson, D. R., & Smalheiser, N. R. (1997). An interactive system for finding complementary literatures: a stimulus to scientific discovery. *Artificial Intelligence*, *91*(2), 183–203.

Symantec. (2019). *Internet Security Threat Report* (Vol. 24). Retrieved from

https://www.symantec.com/security-center/threat-report

Timsina, P., Liu, J., & El-Gayar, O. (2016). Advanced analytics for the automation of medical systematic reviews. *Information Systems Frontiers*, *18*(2), 237–252.

Trochim, W., & Donnelly, J. (2006). *The Research Methods Knowledge Base* (3rd ed.). Atomic Dog Publishing Inc.

Turtle, H. R., & Croft, W. B. (1992). A comparison of text retrieval models. *Computer Journal*, *35*(3), 279–290.

Usefulness. (2019). Retrieved February 25, 2019, from https://www.merriam-webster.com/dictionary/usefulness

Vaishnavi, V., & Kuechler, W. (2004). Design Science Research in Information Systems Overview of Design Science Research. Retrieved September 4, 2015, from http://www.desrist.org/design-research-in-information-systems/

Verizon. (2017). 2017 Data Breach Investigations Report. *Verizon Business Journal*, (1), 1–48. Retrieved from http://www.verizonenterprise.com/verizon-insights-lab/dbir/2017/

Viduto, V., Maple, C., Huang, W., & López-Peréz, D. (2012). A novel Risk Assessment and Optimisation Model for a multi-objective network security countermeasure selection problem. *Decision Support Systems*, *53*(3), 599–610.

Wang, Q., & Zhu, J. (2016). Optimal information security investment analyses with the consideration of the benefits of investment and using evolutionary game theory. *Proceedings of 2016 International Conference on Information Management, ICIM 2016*, 105–109.

Weishäupl, E. (2017). Towards a Multi-objective Optimization Model to Support Information Security Investment Decision-making. *Proceedings of the 4th Workshop on Security in Highly Connected IT Systems - SHCIS '17*, 37–42.

Wiener, E., Pedersen, J. O., & Weigend, A. (1995). A neural network approach to topic spotting. *Proceedings of SDAIR95 4th Annual Symposium on Document Analysis and Information Retrieval*, *332*(Las Vegas, NV), 317–332. Retrieved from http://www.stern.nyu.edu/~aweigend/Research/Papers/TextCategorization/Wiener.Pedersen.Weigend_SDAIR95.ps

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., Fernandes, V. B., Van Moorsel, A., Janicke, H., & Emmerich, M. (2016). Two-stage Security Controls Selection. *Procedia Computer Science*, *100*, 971–978.

APPENDICES

APPENDIX A: DEFINITIONS OF CYBER TERMS

In this appendix we define a few important cyber terms that recur in our dissertation.

- We use the word **cyber** to denote associations with the information technology (IT) and information systems (IS) domains, including computers, computer networks, hardware, and software.
- A **cyber system** is a system composed of IT/IS components, though it may also encompass non-cyber entities. Smart phones, automated teller machines, home automation systems, digital cameras, e-commerce platforms, and even the Internet are all examples of cyber systems of various sizes.
- A **cyber vulnerability** is a known or unknown weakness in a cyber system. When we hear vulnerability, we most often think of software flaws, but cyber systems are also vulnerable to a number of other conditions, such as natural disasters and human error.
- A **cyber threat** is any adverse event, regardless of intent, that disrupts a cyber system by activating a vulnerability. Common threats include errors, routine failures, natural disasters, and cyberattacks.
- A **cyberattack** is a purposeful "attempt to damage, disrupt, or gain access to" a cyber system (Random House Inc., n.d.). Cyberattacks are often undertaken for nefarious purposes, though sometimes they may be pranks.
- We use the term **cyber effect** to refer to the outcome after a cyber threat has been realized. Cyber effects are most commonly categorized in terms of loss of confidentiality, integrity, or availability of the cyber system or one of its parts.
- **Risk** is a condition faced by an organization or entity. It encompasses the likelihood that a threat or adverse event will occur and the degree of damage or injury (also known in organizational contexts as **mission impact**) if the threat is realized.

- **Cyber risk** is the risk that an organization or entity faces due to its association with or reliance on cyber systems.
- **Mitigations** represent tools or techniques that may counter or reduce the impact of cyber threats. In this paper, we consider the terms **security controls** and **countermeasures** to be synonymous with mitigations.
- **Risk assessment**, according to Kaplan and Garrick, is an attempt "to envision how the future will turn out if we undertake a certain course of action (or inaction)." (Kaplan & Garrick, 1981). In the case of **cyber risk assessment**, the objectives are to understand and prioritize identified cyber risks in order to understand the status quo and determine mitigating courses of action for high priority threats.

APPENDIX B. USE CASES

The following use cases describe the main uses of a system such as the one illustrated in the drawing in Figure 6 and described in detail in Appendix C. Instantiations of the artifact of this research can be a key part of such a system that would allow a CSE to leverage the practical utility of the artifact. We refer to these instantiations as the Matcher. Each Matcher is a classifier for a particular threat that; it labels new potential mitigation document instances as relevant to the given threat. Each instantiation of the Matcher comes to exist by virtue of a model building process shown in Figure 4. In the notional architecture in Figure 6, we have allocated the process of creating new Matcher instantiations to the Preprocessor. Use cases 1 and 2 relate to the Matcher (artifact). Use cases 3 through 7 relate to a system such as the one depicted in Figure 6 which would encompass the artifact and support practical usage of it.

Use Case 1	Label potential mitigations relevant or not relevant to a specified threat
Preconditions	Unlabeled potential mitigation documents exist to be labeled. A model (classifier) and semantic space exist that can be used to determine the relevance of new mitigation documents for the specified threat, *T*.
Success End Condition	Unlabeled mitigations have been labeled relevant or not relevant to *T*.
Actors	Matcher
Description	1. The Matcher pertinent to *T* ingests unlabeled potential mitigations. 2. The Matcher transforms each mitigation, *M*, to the features of the semantic space. 3. The Matcher applies the classifier to each transformed *M*. 4. The Matcher outputs a relevant or non-relevant label and a confidence value for each *M* relative to *T*.

Use Case 2	Create a model for a new threat
Preconditions	A new threat, *T*, exists. At least *n* labeled mitigations relevant to *T* and at least *m* non-relevant instances exist. (We arbitrarily used 20 for *n* and 200 for *m*.)
Success End Condition	A model (classifier) and semantic space exist that can be used to determine the relevance of new mitigation documents for the specified threat, *T*.
Actors	Preprocessor (Model Builder)
Description	1. The Model Builder applies LSA to create a threat-specific semantic space from the provided labeled mitigations. 2. The Model Builder saves the semantic space and the labels. 3. The Model Builder uses *T* as a query against the semantic space returning mitigations in order from most to least relevant *T*. 4. The Model Builder makes training data from the top 100 mitigations and trains a classifier for *T*.
Variations	

Use Case 3	Get a list of relevant mitigations for a given threat
Preconditions	Threat documents, mitigation documents, and mappings exist. A model exists that can determine the relevance of new mitigation documents for the given threat.
Success End Condition	A list of relevant mitigation documents for the given threat has been produced.
Actors	CSE, System
Description	1. The CSE specifies an existing threat *T* and requests a list of relevant mitigations. 2. If there are any unmapped mitigations in the system, the system first performs use case 4 to map them.

Use Case 3	Get a list of relevant mitigations for a given threat
	3. The system selects all mitigations labeled as R for threat *T* and returns the mitigation id, text, relevance indicator, relevance score, relevance source, verified indicator, and verified source.

Use Case 4	Classify unmapped mitigations relative to a specified threat
Preconditions	Threat documents, mitigation documents, and mappings exist. Some new mitigations exist that are not yet mapped to any threat. A model exists that can predict the relevance of new mitigation documents for the given threat.
Success End Condition	New mitigations have been labeled with their relevance to the specified threat and marked as unverified.
Actors	System
Description	1. The system loads the appropriate model to classify unlabeled mitigations for the specified threat, *T*. 2. The system applies the threat-specific model to the unlabeled mitigations. 3. The model predicts and outputs a label and a confidence value for each unlabeled mitigation to indicate its relevance or non-relevance to *T* as described in use case 1. 4. The system saves the threat-specific label determinations and relevance scores for each previously unlabeled mitigation, and marks the mapping as not verified.
Variations	Future: The system automatically marks new mappings verified when the confidence exceeds an established value *C*.

Use Case 5	Add a new mitigation
Preconditions	The CSE has a new mitigation to add. The CSE has verified that the mitigation to be added is not already in the data store.

Use Case 5	Add a new mitigation
Success End Condition	The new mitigation has been added to the system and is ready to be labeled upon request.
Actors	CSE, System
Description	1. The CSE requests to add the new mitigation to the data store. 2. The system accepts and saves the new mitigation.
Variations	Future: The system automatically detects and prevents addition of duplicate mitigations.

Use Case 6	Add a new threat
Preconditions	The CSE has a new threat to add. The CSE has at least *n* labeled mitigations relevant to the threat. The CSE has verified that the threat to be added is not already in the data store.
Success End Condition	The new threat, associated relevant mitigations, and verified mappings have been added to the system and a model has been created to handle the new threat.
Actors	CSE, System
Description	1. The CSE requests to add the new threat, *T*, and associated mitigations to the system. 2. The system accepts and saves the new threat, mitigations, and mappings for the relevant mitigations provided. The mappings are marked as verified. 3. The system trains a new model for *T* per use case 2 using the provided labeled data and *m* negative instances drawn at random from the mappings already in the system. 4. The system saves the model for future use.
Variations	Future: The system automatically detects and prevents addition of duplicate threats.

Use Case 7	Review/adjudicate matches
Preconditions	Mappings exists in the data store.
Success End Condition	The status has been changed for requested mappings.
Actors	CSE, System
Description	1. The CSE requests to review unverified mappings, potentially specifying a confidence threshold. 2. The system presents the new mappings to the CSE. 3. For each mapping, a. The CSE approves, rejects, or skips. b. For approved or rejected mappings, the system saves the action.
Variations	Future: The system also allows the CSE to review existing mappings by specifying selection criteria. This could be used to correct errors that made it past the review process.

APPENDIX C. SOLUTION ARCHITECTURE

Design and Architecture

In order for the approach described in Chapter 4 to be useful to the cybersecurity expert in the context of cyber risk assessment, it must exist within a system with which the CSE can interact. This appendix describes the data model and an overall architecture for such a system. It has been designed modularly and using object-oriented principles so that any of the threat-mitigation matching techniques investigated in this research could be incorporated as the Matcher.

Data Model

In this section, we present a logical view and description of the data types and relationships inherent in the artifact (Figure C.1). Note that, although this data model is based on the CAPEC data, it is not limited to CAPEC and is intended to be extensible to threat and mitigation documents from other sources.

Catalog is the main object. It is a container for all the threats, mitigations, and associated mappings. Each **Threat** has a unique identifier (ID), a short title, and a description which can be verbose. A threat may have one of three levels of Abstraction (meta, standard, or detailed). We are focusing on CAPEC threats at the *standard* level of abstraction, because they have the best balance of specificity versus generality for our purposes. Meta threats represent groupings of similar threats, accessed via the ParentThreat property of a standard threat. Mitigations at the meta level are associated to the standard threats that are children of the meta threat. The DomainOfAttack (e.g. hardware, software, communications) and MechanismOfAttack (e.g. subvert access control) properties are also used to group related threats. Detailed threats are further refinements of standard threats, accessible via the ImmediateChildren property. KeyPhrases are significant words or phrases extracted from the threat title and description which succinctly represent the meaning of the threat.

Each **Mitigation** has a unique identifier (ID), a short title, and a description, which can be verbose. The DomainOfAttack (e.g. hardware, software, communications) and MechanismOfAttack (e.g. subvert access control) properties are also used to group mitigations

that counter certain related categories of threats. KeyPhrases are significant words or phrases extracted from the mitigation title and description, which succinctly represent the meaning of the mitigation.

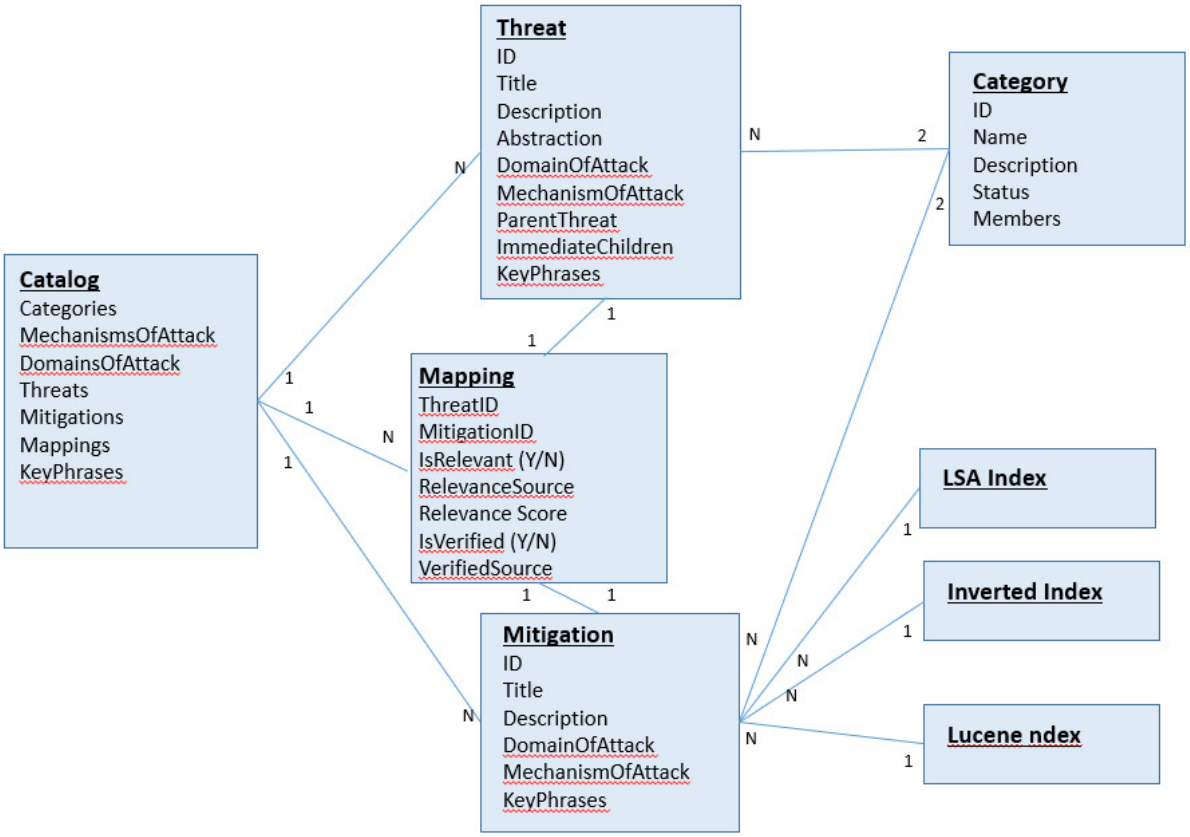

Figure C.1. Overall Data Model

A **Mapping** object represents a threat and mitigation pair, represented by a ThreatID and MitigationID, respectively. The IsRelevant and IsVerified properties are used to indicate the strength of the match. When IsRelevant is true, this means that the mitigation is a countermeasure for the threat, either because it was extracted based on a CAPEC threat-mitigation mapping, or, if a new mitigation, as a result of a decision by the Matcher. When IsVerified is true, this means that the match has been independently verified. IsVerified and IsRelevant will always be true for matches extracted from CAPEC. For decisions made by the Matcher, IsRelevant will be true but IsVerified will initially be false until a SME concurs with the match. Mappings where IsRelevant and IsVerified are both true can be used as training data. Mappings where IsRelevant is false are not usually stored, except for diagnostic purposes.

Architecture Overview

Figures C.2 and C.3 illustrate the architecture into which instantiations of the threat-mitigation matcher can be inserted. Figure C.2 illustrates the preprocessor architecture. During **Data Extraction**, the CAPEC XML structure described in the Data Source and One-time Data Preparation section above is unpacked and transformed into the structure shown in Figure C.1 and described above.

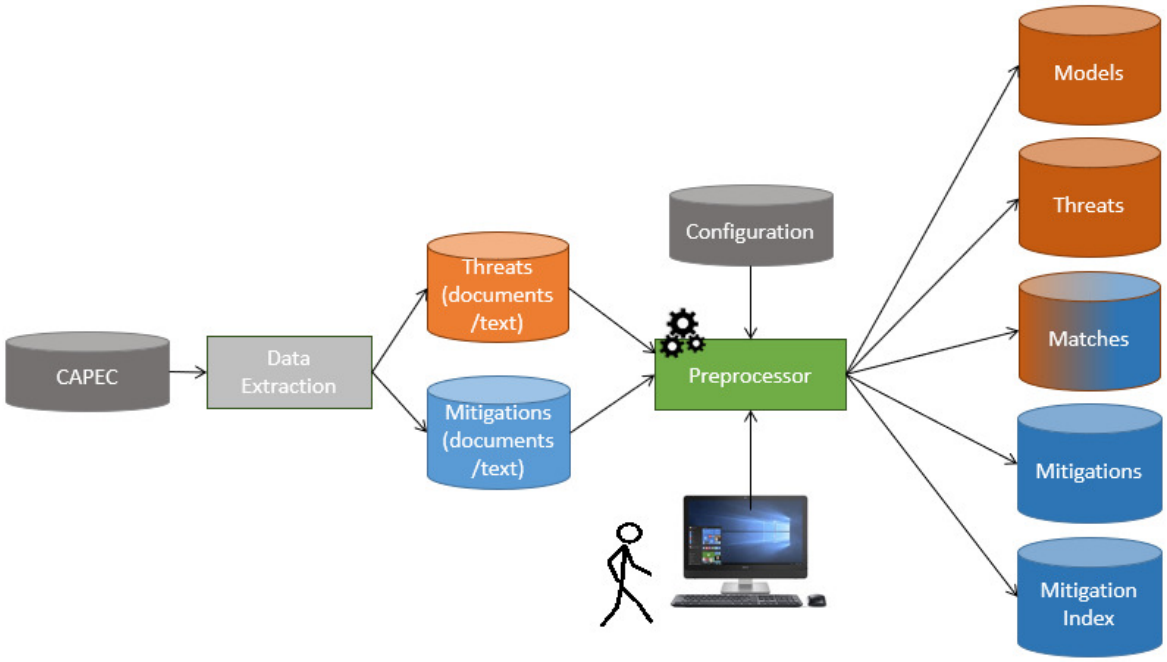

Figure C.2. Preprocessor Architecture

Preprocessor. The Preprocessor includes these functions: (a) convert the threat and mitigation text into threat documents, mitigation documents, and matches, (b) create indices to support the LSA representations of the documents, and (c) train model(s) as needed for the matcher. In (a) the threat and mitigation texts extracted from CAPEC are lower-cased, tokenized, and stemmed. In this architecture, a model will be trained for each threat then saved for reuse when matching is necessary. Over time, after substantial additional labeled data has been accumulated through the use of the system, it may make sense to train new models to take advantage of the new semantic knowledge provided.

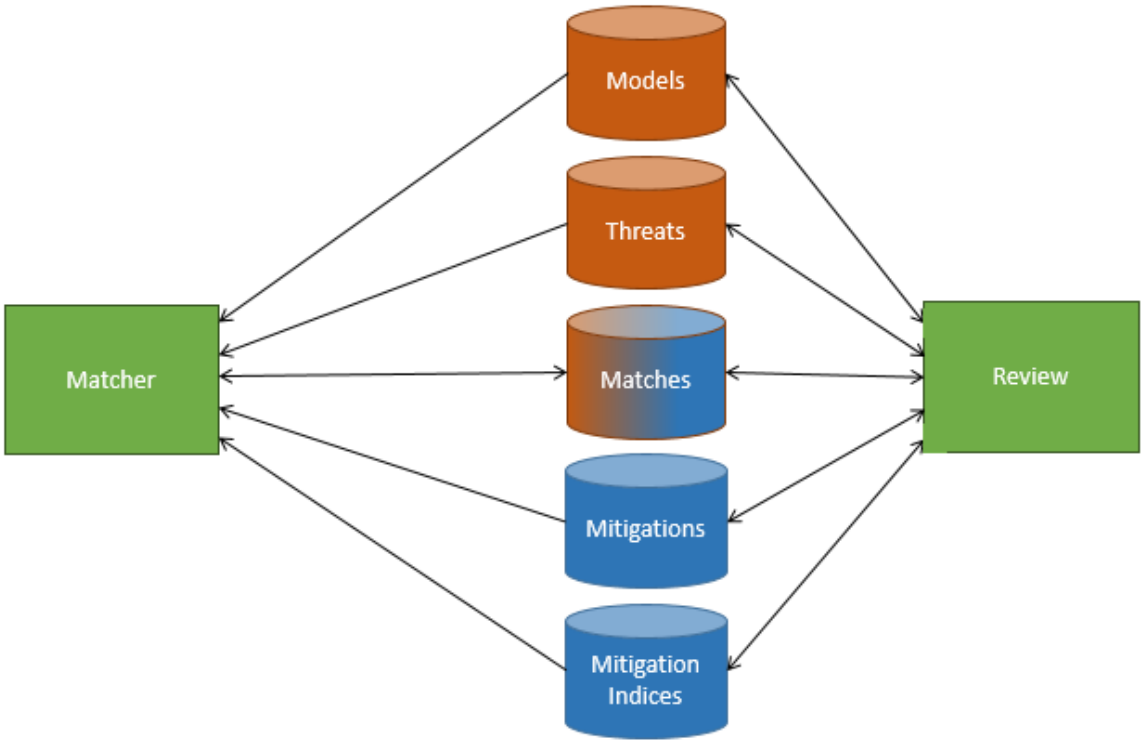

Figure C.3. Matcher Architecture

Matcher. The Matcher, shown in Figure C.3, is the main component of this research. It is executed on demand. To control the scope of the research, we assumed a fixed set of threat documents and a clear delineation between threats and mitigations. We decided to fix the pool of threats because our approach relies on the pre-existence of labeled data consisting of mitigations known to be relevant to the threat. We assumed that a document consists of either a threat or a mitigation but not both so that we did not have to invent a way to separate composite documents into the requisite parts. Our approach can accept new mitigations which it will match to existing threats. It can also accept new threats with the caveat that labeled data consisting of known relevant mitigations would have to be created so that a classifier can be trained.

The Matcher uses the data output from the Preprocessor. It is implemented as described in Chapter 4 to select relevant mitigations for a given threat. Existing threats, mitigations, and matches extracted from CAPEC reside in their respective data stores as a result of data extraction and preprocessing. New mitigations are classified as relevant or not relevant to a selected threat by applying the models previously trained and stored. Each match is written to

the Matches data store as a row that references a threat, a matching mitigation, the rank of the match (if applicable), and a flag to designate matches considered to be ground truth, such as from training data or SME confirmation. Matches recorded from the labeled data will be flagged as verified. Matches generated by the Matcher will initially be flagged as unverified.

Review. The review function allows a subject matter expert to examine new matches generated by the Matcher. The SME can confirm the match or indicate that the given threat-mitigation pair is NR. It is not required that all generated matches must be reviewed. Initially that may be the practice, but as experience is gained in practical use, it may be that some new matches can be confirmed based on the model's confidence in the match leaving only the least confident matches for SME review. Regardless of how review is handled, we think it is important to present the review status of each returned match to CSE who requests a list of mitigations for a given threat. This will help the CSE to compensate for errors in precision where a mitigation that is not relevant may be erroneously presented.

APPENDIX D. DESIGN TRIALS

This appendix discusses details for each design iteration. At the outset, we had three design concepts for the threat-mitigation matcher artifact: classification, ranked retrieval, and a hybrid of the two. We explored a number of designs, including various classifiers, feature sets, and feature reduction techniques.

Classification

In this section we discuss the iterative process for applying classification in the design of our artifact. Following the medical SR literature discussed above in the Literature Review, we started with a single threat and some labeled mitigation data that contains instances that are relevant and not relevant to the threat. We designate this as the "per-threat" approach. In order for the "per threat" approach to solve the problem at hand, we would have to eventually train a classifier for each existing threat and likewise for new threats that come along; however, this does not seem like an unreasonable requirement. New threats do come along, but the library of known threats is relatively stable over time. In the ten months since we started this research, the CAPEC dataset has undergone two subsequent releases but only two new standard threats have been added to CAPEC. Tables D.1 and D.2 summarize several design iterations on the "per threat" approach, each of which is discussed in more detail following the tables. Later on, we discuss several trials where we experimented with a "one for all" approach.

Table D.1. "Per Threat" Summary of Classification Iterations – Full Text

#	Trial	Class	P	R	FP	F	C	I
1	Threat 49, one row for each mitigation Features: • Mitigation text, • R/NR indicator Filter: StringToWordVector • TFIDF • Lower case • Word tokenization (removes punctuation)	R	0.25	0.11	0.01	0.15	1	8
		NR	0.99	0.99	0.89	0.99	601	3

#	Trial	Class	P	R	FP	F	C	I
	• Stemming • Eliminate stop words • Retain 1,000 words Classifier: Weka SMO							
2	Same as trial 1 except Attribute selection: top 50 attributes based on information gain	R	0.50	0.11	0.00	0.18	1	8
		NR	0.99	0.99	0.89	0.99	603	1
3	Threat 49, one row for each mitigation Features: • Mitigation text • R/NR indicator Filter: StringToWordVector • TFIDF • Lower case • Word tokenization (removes punctuation) • Stemming • Eliminate stop words • Retain 1,679 words Classifier: Weka SMO	R	0.92	0.48	0.00	0.63	12	13
		NR	0.98	0.99	0.52	0.99	611	1
4	Same as trial 5 plus attribute selection: top 200 attributes based on information gain	R	0.83	0.39	0.002	0.53	5	8
		N	0.99	0.99	0.62	0.99	599	1

Trial 1

In the first trial, we made a training data set consisting of one row for each mitigation, where each row contained the mitigation text and an attribute to indicate if the mitigation is or is not relevant (R/NR) to threat 49. This data set was extremely unbalanced, containing 9 items in the R class and 604 (>99%) in the NR class. The input dataset was preprocessed in Weka by applying a StringToWordVector filter using TFIDF weighting, lower case, word tokenization, stemming, and stop word elimination, retaining 1,000 words. We trained a SMO model from the filtered data set. The only good thing to be said about this model is the false positive rate for the R class is low. Precision and recall for the R class (0.25/0.11) were worse than the flip of a coin; hence, unacceptable.

Trial 2

In the second trial, we made a training data set and applied the filter as described in trial 1, then selected the top 50 attributes using information gain. We trained a SMO model using the filtered and reduced data set. This resulted in improved precision (0.50) for the R class and maintained the low false positive rate, but recall was still poor (0.11).

As expected, the models in trials 1 and 2 were both very good at correctly classifying non-relevant instances due to the class imbalance in the data, but they were not good at correctly classifying relevant instances, likely for the same reason. It became apparent that it was necessary to do something about the class imbalance. In addition, note that this approach did not utilize any information from the threat; thus, such an approach may not generalize to other threats. This ultimately led us to try the keyword approach described later in trials 5 and 6.

Trial 3

In trial 3, we followed the method described in trial 1, except we retained 1,679 words from the StringToWordVector filter. We selected the number 1,679 to facilitate comparison with the LSA ranked retrieval results in trials 9 and 10 discussed later (1,679 was the number of unique words identified during the LSA transformation). We did not perform any attribute reduction. We trained a SMO model from the filtered data set, achieving precision of 0.92, recall of 0.48, and minimal false positives for the R class. This is an improvement over the prior trials and suggests that retaining more words is better. Precision and recall for the NR class were 0.98 and 0.99, respectively. As mentioned previously, the dataset is highly imbalanced in favor of the NR class but we are primarily interested in the R class. For the R class, precision in this trial was good (0.92), but recall was not good enough. There are only a small number of relevant mitigation documents for a given threat and at 50% recall, we would be failing to recommend over half of them.

Trial 4

In trial 4, we made a training dataset similar to the one in trial 3, but used attribute selection to choose the top 200 attributes based on information gain. We trained a SMO model from the filtered and reduced data set, achieving precision and recall (0.83/0.39) for the R class and (0.99/0.99) for the majority NR class. Recall and precision here were worse than trial 3,

suggesting that the additional words do add some information to the model, which is lost during the information gain reduction.

Table D.2. "Per Threat" Summary of Classification Iterations – Keywords

#	Trial	Class	P	R	FP	F	C	I
5	Threat 49, one row for each mitigation, 2/3 undersampling of the NR class Features: • Presence/absence of threat keywords (TextRank + synonyms) in mitigation text • R/NR indicator Filter: • Lower case • Eliminate stop words and punctuation Classifier: Weka SMO	R	0.82	0.67	0.00	0.74	14	3
		NR	0.97	0.99	0.33	0.98	207	7
6	Threat 49, one row for each mitigation, 2/3 undersampling of the NR class instances and 100% SMOTE oversampling of the R class Features: • Presence/absence of threat keywords in mitigation text • R/NR indicator Filter: • Lower case • Eliminate stop words and punctuation Classifier: Weka SMO	R	0.97	0.74	0.00	0.84	31	1
		NR	0.95	1.00	0.24	0.97	211	11

Trial 5

An inspection of the mitigation text for the 9 relevant examples in trial 2 revealed that those which were correctly classified have in common some key words from threat 49 suggesting keywords/phrases as a possible way to introduce information from the threat text into the approach, while also potentially improving the classification results. Table D.3 shows the keywords/phrases automatically extracted by TextRank for threat 49 and its associated

mitigations. Some of these keywords were rather rough, so we decided to clean them up manually. The improved keywords are also shown in the table. While it would be nice in the long run (assuming an approach based on keywords bears fruit) to automate the keyword/phrase extraction, it will suffice to prove the concept if we use expert-assigned keywords.

Table D.3. Keywords for Threat 49

Text Rank	Improved
attack	password policy
adequate password policy	password
brute force attack	policy
dictionary attacks	brute force
effective e	brute
feasible	force
computationally	combination
maximum length	trial and error
password	trial
password brute	length
possible passwords	throttle
possible value	limit
proper enforcement	strong password
mechanism	strong
pure brute force attack	weak password
rainbow tables	weak
strong passwords	user
weak other password	

Next, we investigated techniques to address the class imbalance in the data (Cohen et al., 2006; Miwa et al., 2014; Timsina et al., 2016). The most obvious solution was to add more relevant mitigations, so we extracted about a dozen additional documents relevant to threat 49 from the internet and added them to the data. In addition, we decided to try undersampling of the dominant (NR) class. The danger of undersampling is information loss; however, due to the extreme imbalance, it seemed a risk worth taking. We also decided to try oversampling of the minority (R) class. Oversampling can result in overfitting, but this likewise seemed like a risk worth taking in the given situation.

In trial 5, we created a dataset with one entry for each mitigation in the corpus, including 12 additional mitigations relevant to threat 49 drawn from the Internet. In this dataset, the features consisted of threat 49 keyword counts plus the R/NR indicator. To reduce class imbalance, we under-sampled by randomly dropping 2/3 of the NR instances, then we trained

a SMO model. The training set contained about 225 instances (slight variances due to random sampling) with about 9% relevant. Although still notable, the class imbalance was not as severe as was the original dataset. In this trial, the SMO model showed improved precision (0.82) and recall (0.67) of the R class, low false positives (0.00), and no appreciable impact to the precision and recall of the NR class. We used several different methods for determining the keyword counts, including a simple count of the times a keyword appeared in the document (TF), TFIDF, TF divided by the total number of words in the document, and 0 or 1 to indicate the keyword is present or absent in the document. Of these, the presence/absence approach yielded the best results, which are reported here.

Trial 6

To further improve balance, in trial 6 we followed a process similar to trial 5, but with 100% Synthetic Minority Oversampling Technique (SMOTE) (He & Garcia, 2009; Liu et al., 2016) based on 5 nearest neighbors to double the number of instances of the R class. The SMOTE technique creates new instances of the minority class by drawing features from the K (e.g. 5) nearest minority instances based on Euclidean distance in the feature space. We trained a SMO model for threat 49. The training set contained about 225 instances, 18% relevant. Although still significant, the class imbalance was less pronounced than the prior trial. With combined undersampling of the NR class and oversampling of the R class, the SMO model achieved precision of 0.97 and recall of 0.74 for the R class with minimal false positives and no appreciable impact to the precision and recall of the NR class. The undersampling of the NR class and oversampling of the R class showed some modest improvement in results over prior trials, especially in regards to precision. However, a recall of 0.74 means we would fail to recommend about a quarter of the available mitigations for threat 49.

We were curious about the potential impact of additional under- and oversampling, so we experimented with 3/4 undersampling of the NR class, and 200% oversampling of the R class for threat 49. When comparing 3/4 undersampling versus 2/3 undersampling of the NR class for the same oversampling percentage (100%) of the R class, the precision, recall, and F-measure for 2/3 undersampling was better. When we increased oversampling of the NR class to 200%, recall of the R class seemed to improve overall but with a small toll on precision. In the 200% oversampling case, the model failed to properly classify test samples. These results

suggest that 3/4 NR undersampling was too much and, when combined with 200% R oversampling, the model was becoming overfit to the training data.

Ranked Retrieval

As a possible alternative to classification, in the spirit of iterative design, we investigated two ranked retrieval (i.e. search engine) approaches to matching relevant mitigations for a given threat similar to (Foltz, 1990; Goldrich et al., 2014; Swanson & Smalheiser, 1997). In trials 7 and 8, we investigated ranking based on a combination of the Boolean and Vector Space models as implemented in Apache Lucene (Apache Foundation, 2013). In trials 9 and 10, we investigated ranking based on Latent Semantic Analysis as implemented in Gensim (Rehurek, 2018). The results are summarized in Table D.4 with details provided after the table.

Table D.4. "Per Threat" Summary of Ranked Retrieval Iterations

#	Trial	Class	P@25	R	FP	F	C	I
7	Threat 49, one row for each mitigation Features: • Full mitigation text • Tokenized, stop words removed, TFIDF Apache Lucene with Standard analyzer similarity to threat keywords (top 25)	R	0.48	0.48			12	13
8	Threat 49, one row for each mitigation Features: • Full mitigation text • Tokenized, stop words removed, TFIDF, stemmed Apache Lucene with Custom analyzer similarity to threat keywords (top 25)	R	0.60	0.60			15	10
9	Threat 49, one row for each mitigation Features:	R	0.92	0.92			23	2

#	Trial	Class	P@25	R	FP	F	C	I
	• Full mitigation text LSA similarity to full threat text (top 25)							
10	Threat 49, one row for each mitigation Features: • Full mitigation text LSA similarity to threat name (top 25)	R	0.84	0.84			21	4

Trials 7 and 8

For trials 7 and 8, we used Apache Lucene, which implements the Vector Space models. Retrieval in Lucene is a two-stage process. First, an index of the document corpus is created; then queries can be run against the index. A Lucene index is an inverted index of terms in documents, where each term consists of a field name and corresponding field token(s). The tokens are, in essence, values of the fields input into the indexing process, except in the case of text inputs they may have been tokenized, lower-cased, stemmed, etc. depending on the Lucene Analyzer chosen. The inverted index supports scoring of results during the search stage such that documents which contain more of the search terms will score higher and thus will be deemed more relevant. Items designated as "TextField" are tokenized by the Analyzer which those designated as "StringField" are captured literally in the index. We indexed the fields from each mitigation as shown in Table D.5. Meanings of the fields are described in the Data Source and One-time Data Preparation section above. We elected not to tokenize the Id and Threat Ids because we included them in the index for diagnostic purposes only (not for searching) and we wanted to preserve their human-readability. We elected not to tokenize the Domain of Attack and Mechanism of Attack because these are metadata which we also wanted to preserve intact. We allowed the remaining fields to be tokenized to improve matching during the search stage.

Table D.5. Fields Indexed for Ranked Retrieval

Field	Type	Index	Store	Rationale
Name	TextField	Yes	Yes	Threat matching
Description	TextField	Yes	No	Threat matching
Keywords	TextField	Yes	Yes	Threat matching

Field	Type	Index	Store	Rationale
Id	StringField	Yes	Yes	Diagnostic
Domain of Attack	StringField	Yes	Yes	Threat matching
Mechanism of Attack	StringField	Yes	Yes	Threat matching
Threat Ids Mitigated	StringField	Yes	Yes	Diagnostic

We experimented with two different analyzers. The StandardAnalyzer is the most commonly used Lucene analyzer. It tokenizes text based on white space, removes stop words, and lower cases the text. We also tried a CustomAnalyzer, in which we added stemming to the other options. We created the search query for each threat by or-ing its respective threat keywords and executed the search over the mitigation text. The query returned the mitigations in rank order by similarity. In a perfect world, the known relevant mitigations should be top-ranked, so we established a relevant/not relevant cutoff at the top 25 for purposes of measuring the efficacy of this approach. At this cut-off, only about half the relevant mitigations were returned, and precision and recall were about equivalent to a coin flip. If we were to use this approach to recommend mitigations, we would not want the cut-off to be much larger than the expected number of relevant results as this would lead to recommending mitigations that are not actually relevant to the threat.

Trials 9 and 10

As mentioned in the Literature Review section, Latent Semantic Analysis has been shown to improve retrieval of relevant documents from a corpus when compared to keyword search because LSA addresses the issue of synonymy inherent in natural language. In trials 9 and 10 we experimented with a ranking approach using LSA. This is also a two-stage process where the corpus must be indexed (i.e. transformed to a semantic space) before it can be queried. We started with a comma-separated-values (CSV) file containing one row for each mitigation, containing the mitigation id, text, and R/NR indicator designating the mitigation's relevance to threat 49. The mitigation text was used to build the semantic space and the other fields were used for evaluation and diagnostic purposes.

For each mitigation text, stop words were removed, then the text was tokenized, lower-cased, and stemmed. Using Gensim, Bag of words (BOW) and TFIDF representations of the

corpus were computed and then TFIDF representation was transformed to a semantic space or Latent Semantic Index (LSI) retaining 200 topics. This is slightly higher than the number of standard threats in CAPEC and fits with optimal LSI dimensionality findings in (Bradford, 2008). Bradford observed favorable results when the number of topics was between 200 and 500 for a corpus with millions of documents. We selected the low end of Bradford's range because our corpus is much smaller than his. The LSI representation was saved for future use in similarity queries. The BOW corpus had 637 documents and 1679 features.

We experimented with two approaches for constructing the threat query. In trial 9, we used the full text of the threat document (tokenized, stemmed, lower-cased, and transformed to the semantic space) as the query and in trial 10 we used the threat name (similarly transformed) as the query. We established the cut-off at the top 25. In trial 9 (precision=0.92, recall=0.92), 22 of the known mitigations earned similarity scores in the top 25, while the others scored 26th, 38th, 40th, and 370th. In trial 10 (precision=0.84, recall=0.84), 21 of the known mitigations ranked in the top 25 and all ranked in the top 82. Trial 9, similarity to full threat text, outperformed trial 10, similarity to threat name. This suggests that a query with more semantic context (i.e. more words) is better.

In terms of precision and recall, the LSA retrieval results are better than the SMO models trained based on words in the mitigation text (trials 1 - 4) but slightly worse than the SMO models trained to emphasize threat keywords in the mitigation text (trials 5 and 6). The LSA results are better than the keyword search trials (7 and 8), which is not surprising given LSA's reputation for improved performance versus keyword search (Deerwester et al., 1990).

Hybrid

Drawing from (Manning et al., 2009), (Nakamoto, 2011), and (Gee, 2003), we experimented with several hybrid approaches that combine ranked retrieval and classification techniques. For these trials we used LSA features in conjunction with the SVM classifier. As mentioned previously, we selected this classifier because support vector machines have been shown to perform favorably for text classification, especially when the number of positive instances per category is small (Platt, 1998). We decided to continue to use SVM in the hybrid trials to facilitate apples-to-apples comparisons with the prior results. The results of these trials are presented in Table D-6 with details following the table.

Table D-6. "Per Threat" Summary of Hybrid Iterations

#	Trial	Class	P	R	FP	F	C	I
11	Threat 49, one row for each mitigation Features: • LSA transform of mitigation text (200 features) • R/NR indicator Classifier: Weka SMO	R	1.00	0.72	0.00	0.76	18	7
		NR	0.99	1.00	0.28	0.99	612	0
12	Threat 49, one row for each mitigation, drop rows not in top 100 similarity scores vs full threat text Features: • LSA transform of mitigation text (200 features) • R/NR indicator (*)The number incorrect does not include the one known relevant mitigation that was ranked outside the top 100. Classifier: Weka SMO	R	0.95	0.75	0.01	0.84	18	6(*)
		NR	0.93	0.99	0.25	0.96	75	1
13	Same as trial 12 except drop rows not in top 100 similarity scores vs threat name (all R samples were in the top 100)	R	0.95	0.76	0.01	0.84	19	6
		NR	0.93	0.99	0.24	0.96	74	1
14	Classifier based on (Gee, 2003) using LSA nearest neighbor and/or majority on mitigation text	R	0.63	0.83	0.03	0.71	5	1
		NR	0.99	0.97	0.17	0.98	92	1 (+2 tie)

Trials 11

In trial 11, we extracted the LSA-transformed representation of each mitigation (200 features) from the semantic space and made a CSV consisting of these features plus the R/NR indicator. We trained a SMO model using this data set. The model in trial 11 achieves very high precision (1.0) and minimal false positives but only mediocre recall (0.72). This suggests that, although this approach would not recommend any errant mitigations, it would fail to recommend nearly 40% of the relevant mitigations.

Trials 12 - 13

Recalling that the dataset is extremely imbalanced in favor of the NR class and that we saw improvement in the results above (trials 5 and 6) when we took steps to achieve better balance in the training data, we decided, in trials 12 - 13, to utilize the LSA similarity scores as a means to balance the training data. That is, we cut the training data off after the top 100 entries based on similarity to the threat text. We intuited that this approach will be better than simply undersampling at random and over-sampling with SMOTE for the following reasons. Undersampling at random could drop relevant entries of which we already have too few. Oversampling with SMOTE adds new instances to the corpus, but no new knowledge. Because the similarity score imparts some knowledge about the semantics of the entries, keeping the most similar entries will keep most of the relevant entries and in addition the non-relevant entries that are most difficult to discriminate.

In trial 12, we used similarity scores resulting from comparing the full threat test against the mitigations in the semantic space up to the cut-off. In trial 13, we used similarity scores resulting from comparing the threat name against the mitigations up to the cut-off. In trials 12 and 13, we trained the models using only the 200 LSA features and the R/NR indicator. Trial 12 (similarity based on full threat text) and 13 (similarity based on threat name) produced similar balance of precision and recall, while keeping false positives low (Trial 12: P=0.95, R=0.75, FP=0.01; Trial 13: P=0.95, R=0.76, FP=0.01), but it is worth noting that the recall number is somewhat optimistic because it does not account for one relevant mitigation that was dropped from the training set because it ranked lower than the cut-off. We cannot afford to omit up to 25% of the relevant mitigations.

Trial 14

In trial 14 we developed a method for classifying mitigations relevant/not-relevant to a given threat inspired by Gee (Gee, 2003) and Foltz (Foltz, 1990). First, LSA was utilized to create a semantic space for a training set consisting of 80% of the existing labeled mitigation documents and an external index was constructed to maintain the known relevance status of the mitigation with regard to the threat. When a new mitigation document was presented, it was used as a query against the semantic space, returning a ranked list of other mitigation documents similar to the query from most similar to least. The trial 14 classifier classifies the new

document in three stages. First, it is classified according to the class of its nearest neighbor in the space (i.e. the existing mitigation document whose similarity score is highest). Next, the new mitigation document is classified according to class of the majority of all results in the ranked list truncated at an arbitrary cut-off N. Finally, if the majority and nearest neighbor stages agree, the new mitigation document is deemed to be of the nearest neighbor's class. If the majority and nearest neighbor stages do not agree, the dispute is settled by the third stage which attempts to detect the skew of the new document towards one class or the other. We implemented the first 2 stages using an arbitrary cut-off of top 5, but for the tie-breaker we took a default where tie equates to an incorrect classification (i.e. for the R class, resulting prediction is NR; for the NR class, resulting prediction is R). We intended to go back and implement a more robust tie-breaker if observations revealed an approach that would be beneficial.

In trial 14, there were 546 mitigations in the training set and 101 (6 relevant and 95 not relevant to threat 49) in the testing set. On the test data, this method yielded precision of 0.63 and recall of 0.83 with 3% false positives on the R class and 0.99/0.97/17% for the NR class. Two ties were encountered in the NR class indicating the need to consider a better tie-breaker before this method could to be viable.

A possible stage 3 algorithm, based on (Gee, 2003) is as follows for arbitrary A, B, and C which Gee set to 0.7, 0.7, and 0.65 respectively:

- If the average of the majority scores > A and the nearest neighbor score < B, use the majority class
- If the average of the majority scores < B and the nearest neighbor score > A use the nearest neighbor class
- If the nearest neighbor score > C use the nearest neighbor class
- If the average of the majority scores > C use the majority class
- If still not determined, result = incorrect classification

Analysis of Text

Success in classifying textual data is heavily influenced by the characteristics of the text itself. Having experimented with a few variations, it made sense to pause and look closely at the text of threat 49 for insights on the matching successes and failures. In the training corpus, there are 25 known relevant mitigations. Using diagnostic tools, we identified 6 mitigations that were commonly misclassified in the trials. One thing the false negative instances had in

common is that they lack any text that helps the reader understand how the mitigation addresses the threat. The false positives fell into two categories: (a) some dealt with password vulnerabilities but not specifically password brute force guessing and (b) others dealt with brute force guessing but not of passwords. We hypothesized that improving the mitigation texts to include an explanation of how each one addresses the threat would improve the match results by reducing the FNs. In some applications of text mining, the text "is what it is" and we have to use what we find (e.g. ratings, surveys, news articles). For threat-mitigation matching, we have influence over the problem space and thus we do have the luxury of recommending improvements to the threat and mitigation documents to better support automated matching in the future. With that in mind, we augmented the text of the FPs and FNs then reran selected trials as shown in Table D-7 and described below the table. A side-by-side comparison of the results for the R class on the original and improved mitigation text for the best trials is provided in Table D-8.

Table D-7. "Per Threat" Summary (Improved Mitigation Text)

#	Trial	Class	P	R	FP	F	C	I
15	Threat 49, one row for each mitigation, enhanced mitigation text with vector space representation and TFIDF (comparable to trial 3)	R	1.00	0.56	0.00	0.72	14	11
		NR	0.98	1.00	0.44	0.99	612	0
16	Same as trial 15 plus attribute selection: top 200 attributes based on information gain (comparable to trial 4)	R	1.00	0.56	0.00	0.72	14	11
		NR	0.98	1.00	0.44	0.99	612	0
17	Threat 49, one row for each mitigation, full corpus, 200 LSA features from enhanced mitigation text (comparable to trial 11)	R	0.95	0.80	0.00	0.87	20	5
		NR	0.99	0.99	0.20	0.99	611	1
18	Threat 49, one row for each mitigation, 200 LSA features from enhanced mitigation text, drop rows not in top 100 (comparable to trial 12), Weka SMO	R	0.95	0.80	0.01	0.87	20	5
		NR	0.94	0.99	0.20	0.96	74	1
18b	Same as trial 18 but retain top 200 rows	R	1.00	0.70	0.00	0.82	14	6
		NR	0.97	1.00	0.30	0.98	180	0
18c	Same as trial 18 but retain top 300 rows	R	0.94	0.64	0.00	0.76	16	9
		NR	0.97	0.99	0.36	0.98	274	1

#	Trial	Class	P	R	FP	F	C	I
18d	Same as trial 18 except using scikit-learn SVM.SVC	R	0.96	0.92	0.01	0.94	23	2
		NR	0.97	0.99	0.08	0.98	74	1
19	Classifier based on (Gee, 2003) using LSA nearest neighbor and/or majority on enhanced mitigation text (comparable to trial 14)	R	0.75	1.00	0.01	0.86	6	0
		NR	1.00	0.99	0.00	0.99	200	2

Trials 15 - 16

We ran trials 15 and 16 to see if the improved mitigation text yielded improved results when classifying the text using the Vector Space Model and TFIDF weights without and with information gain attribute selection. These compare with trials 3 and 4 in Table 3. We saw improvement in recall and precision and reduction in both false negatives and false positives, but recall was still too low for our purposes.

Trials 17 - 18

We ran trials 17 - 18 on the improved text because the corresponding trials in Table 4 showed the best results on the original text. In trial 17, we trained the classifier on the LSA features using the full corpus. In trial 18, we used the top 100 mitigations ranked by similarity to the threat as the training corpus. Trial 17 showed modest improvement in recall but a slight decline in precision over a similar trial (11) and no false positives. Trial 18 showed stable precision and false positive rate and modest improvement in recall over a similar trial (12). This model has good precision and an acceptably low FP rate on the R class, but the recall of 0.80 was concerning because it represents a significant number of relevant mitigations that would not be recommended. We ran alternate versions of trial 18 where we retained the 200 (18b) and 300 (18c) top-ranked mitigations, but the recall of the R class declined as we increased the training dataset, likely because the additional samples were mainly NR samples resulting in increased class imbalance. An alternate version (18d) using scikit-learn SVM.SVC had a modest improvement in precision over the Weka SMO version (18) and a notable improvement in recall.

Trial 19

In trial 19, overall precision and recall for the R class was 0.75/1.0 with 1% FP and for the NR class was 1.0/0.99 with no false positives. For the NR class, in 200 instances, the predicted and actual labels agreed, 2 resulted in a tie (reaffirming the need to more fully investigate a tie-breaker) where nearest neighbor class predicted NR but top 5 majority predicted R, and none were incorrectly classified. The nearest neighbor similarity range for the R class was 0.47 to 0.92 and for the NR class was 0.36 to 1. The majority mean similarity range for the R class was 0.45 to 0.65 and for the NR class was 0.31 to 0.89. With such large ranges, tie-breaker cut-offs similar to those in Gee's algorithm were not obvious. The majority and nearest neighbor similarities for the two ties, both of the NR class, were 0.41 and 0.38 respectively.

Table D-8. "Per Threat" Results Before and After Text Improvement

#	Trial	Class	P	R	FP	F	C	I
3	Threat 49, one row for each mitigation, mitigation text with vector space representation and TFIDF	R	0.92	0.48	0.00	0.63	12	13
15		R	1.00	0.56	0.00	0.72	14	11
4	Same as trial 3/15 plus attribute selection: top 200 attributes based on information gain	R	0.83	0.39	0.002	0.53	5	8
16		R	1.00	0.56	0.00	0.72	14	11
6	Threat 49, one row for each mitigation, 2/3 undersampling of the NR class instances and 100% SMOTE oversampling of the R class, presence/absence of threat keywords in mitigation text (Note: No after improvement trial)	R	0.97	0.74	0.00	0.84	31	1
11	Threat 49, one row for each mitigation, full corpus with 200 LSA features	R	1.00	0.72	0.00	0.76	18	7
17		R	0.95	0.80	0.00	0.87	20	5
12	Threat 49, one row for each mitigation, drop rows not in top 100, 200 LSA features	R	0.95	0.75	0.01	0.84	18	6 + 1(*)
18d		R	0.96	0.92	0.01	0.94	23	2
14	Ensemble classifier based on (Gee, 2003)	R	0.63	0.83	0.03	0.71	5	1
19		R	0.75	1.00	0.01	0.86	6	0

Table D-8 provides a comparison of results for "per threat" matching approaches for selected trials before and after improvement of the mitigation text. Per common practice, we used the precision, recall, and false positive rates of the R class (based on cross-validation statistics generated during training) to compare the models. From these results, we decided to advance the designs in trials 12/18d and 14/19. These have the best balance of precision and recall on cross-validated training data. We left the designs in trials 3/15, 4/16, and 11/17 behind due to unacceptably low recall. We shelved the design in trial 6 for two reasons. First, its recall lags behind the other retained designs. Second, the automated keyword/phrase extraction was only moderately successful, leaving us with required manual SME intervention to perfect the keywords. Note also that our intuition that improving the mitigation text to describe how the mitigation addresses the threat would yield better matching results is buoyed by these initial results, especially in regards to precision.

Extensibility to Other Threats

Having seen promising results from some "per threat" designs, we wanted to know if these results would extend to other CAPEC standard threats. Tables D-9, D-10, D-11, and D-12 show results for threats 268, 593, 66, and 134 respectively for the designs in trials 13/14 for the unimproved text and 18d/19 for the improved text.

Table D-9. "Per Threat" Comparison for Threat 268

#	Trial	Class	P	R	FP	F	C	I
	Unimproved Text							
13	One row for each mitigation, drop rows not in top 100, 200 LSA features (*) The number incorrect does not include the one known relevant mitigation that was ranked outside the top 100	R	1.00	0.90	0.00	0.95	18	2+ 1(*)
		NR	0.98	1.00	0.00	0.99	80	0
14	Ensemble classifier based on (Gee, 2003)	R	0.66	0.50	0.02	0.57	2	2
		NR	0.96	0.98	0.50	0.97	49	1
	Improved Text							
18d	One row for each mitigation, 200 LSA features, drop rows not in top 100	R	0.95	0.95	0.01	0.95	20	1
		NR	0.99	0.99	0.05	0.99	78	1
19	Ensemble classifier based on (Gee, 2003)	R	0.80	1.00	0.02	0.89	4	0
		NR	1.00	0.98	0.00	0.99	49	1

Table D-10. "Per Threat" Comparison for Threat 593

#	Trial	Class	P	R	FP	F	C	I
	Unimproved Text							
13	One row for each mitigation, 200 LSA features, drop rows not in top 100	R	0.73	0.69	0.12	0.71	22	10+ 4(*)
		NR	0.86	0.88	0.31	0.87	60	8
14	Ensemble classifier based on (Gee, 2003)	R	0.30	0.75	0.04	0.75	3	1
		NR	0.99	0.96	0.25	0.99	167	1 (+6 tie)
	Improved Text							
18d	One row for each mitigation, 200 LSA features, drop rows not in top 100	R	0.94	0.86	0.03	0.90	30	5+ 1(*)
		NR	0.93	0.97	0.14	0.95	63	2
19	Ensemble classifier based on (Gee, 2003)	R	0.45	1.00	0.03	0.91	5	0
		NR	1.00	0.97	0.00	0.99	167	1 (+5 tie)

Table D-11. "Per Threat" Comparison for Threat 66

#	Trial	Class	P	R	FP	F	C	I
	Unimproved Text							
13	One row for each mitigation, 200 LSA features, drop rows not in top 100	R	0.50	0.33	0.06	0.40	5	10+ 5(*)
		NR	0.89	0.94	0.67	0.91	80	5
14	Ensemble classifier based on (Gee, 2003)	R	0.00	0.00	0.03	0.00	0	3 (+1 tie)
		NR	0.96	0.97	1.00	0.98	96	1 (+2 tie)
	Improved Text							
18d	One row for each mitigation, 200 LSA features, drop rows not in top 100	R	0.86	0.90	0.04	0.88	18	2
		NR	0.97	0.96	0.10	0.97	77	3
19	Ensemble classifier based on (Gee, 2003)	R	0.80	1.00	0.01	1.00	4	0
		NR	1.00	0.99	0.00	1.00	98	(+1 tie)

Table D-12. "Per Threat" Comparison for Threat 134

#	Trial	Class	P	R	FP	F	C	I
	Unimproved Text							
13	One row for each mitigation, 200 LSA features, drop rows not in top 100	R	0.86	0.75	0.01	0.80	6	2
		NR	0.98	0.99	0.25	0.98	91	1
14	Ensemble classifier based on (Gee, 2003)	R	0.00	0.00	0.00	0.00	0	1 (+1 tie)
		NR	0.98	1.00	1.00	1.00	111	0
	Improved Text							
18d	One row for each mitigation, 200 LSA features, drop rows not in top 100	R	1.00	0.88	0.00	0.93	7	1
		NR	0.99	1.00	0.13	0.99	92	0
19	Ensemble classifier based on (Gee, 2003)	R	1.00	0.50	0.00	1.00	1	0 (+1 tie)
		NR	0.99	1.00	0.50	1.00	111	0

Table D-13 shows a summary of the cross-validation statistics for the R class for models trained for threats 49, 66, 134, 268, and 593. Note that precision, recall, and false positive rates are better for the improved text when compared to models trained with the unimproved text. Of the two, the SVM classifier based on LSA features and top 100 most similar documents (Trial 18d) has the best precision, recall, and false positive rate when compared to the ensemble classifier (19).

Table D-13. "Per Threat" Models Summary for R Class

#	Trial	Threat	P (Mean)	R (Mean)	FP (Mean)	#C	#I
	Unimproved Text						
13	One row for each mitigation, 200 LSA features, drop rows not in top 100	134	0.86	0.75	0.01	6	2
		49	0.95	0.76	0.01	19	6
		268	1.00	0.90	0.00	18	3
		593	0.73	0.69	0.11	22	14
		66	0.50	0.33	0.06	5	15
			(0.81)	**(0.69)**	**(0.04)**	**(64%)**	**(36%)**
14	Ensemble classifier based on (Gee, 2003)	134	0.00	0.00	0.00	0	1
		49	0.63	0.83	0.03	5	1
		268	0.66	0.50	0.02	2	2
		593	0.30	0.75	0.04	3	1
		66	0.00	0.00	0.03	0	4
			(0.32)	**(0.42)**	**(0.02)**	**(53%)**	**(47%)**

#	Trial Improved Text	Threat	P (Mean)	R (Mean)	FP (Mean)	#C	#I
18d	One row for each mitigation, 200 LSA features, drop rows not in top 100	134	1.00	0.88	0.00	7	1
		49	0.96	0.92	0.01	23	2
		268	0.95	0.95	0.01	20	1
		593	0.94	0.86	0.03	30	6
		66	0.86	0.90	0.04	18	2
			(0.94)	**(0.90)**	**(0.02)**	**(89%)**	**(11%)**
19	Ensemble classifier based on (Gee, 2003)	134	1.00	0.50	0.00	1	1
		49	0.75	1.00	0.01	6	0
		268	0.80	1.00	0.02	4	0
		593	0.45	1.00	0.03	5	0
		66	0.80	1.00	0.01	4	0
			(0.76)	**(0.90)**	**(0.01)**	**(95%)**	**(5%)**

"One for All" - Beyond the Per Threat Approach

So far, we have discussed matching approaches that are implemented on a "per threat" basis. This approach is derived from the medical SRs research discussed in the Literature Review. It is based on the premise that each threat has its own pattern or semantics. A "per threat" solution is not unreasonable and would work for our purposes as described in the Architecture section. However, we wondered if there was a way to implement a "one for all" approach where a single matcher would determine relevant mitigations for any threat contained in the corpus. In the next paragraphs, we discuss two trials towards a "one for all" approach as summarized in Table D-14. We used the unimproved text for these trials because it was not practical to improve the text of the entire CAPEC dataset.

Table D-14. "One for All" Trials

#	Trial	Class	P	R	FP	F	C	I
20	All threat-mitigation combinations, up to 200 LSA features of each, unimproved text, Weka SMO	R	0.00	0.00	0.00	0.00	0	593
		NR	0.99	1.00	1.00	0.99	86915	0
21	Ensemble classifier based on (Gee, 2003)	R					3	8

Trial 20

In Trial 20, we used LSA to create a semantic space representing all the standard threats and a separate semantic space representing all the labeled mitigations for these threats. Then for each combination of a threat and a mitigation, we made a training dataset consisting of the 200 LSA factors representing the mitigation from the corresponding semantic space, the threat id, 163 LSA factors representing the threat from the corresponding semantic space, and a label indicating whether the mitigation was relevant or not relevant to the threat. (Although we specified 200 features when building both semantic spaces, the threat space yielded only 163 features.) The dataset consisted of 364 attributes and 87,000 instances. We trained a SMO model which we hoped might be able to answer for given threat (T) and mitigation (M), is M relevant to T? The results shown in Table 14 indicate that this model will not be able to distinguish relevant T-M pairs from non-relevant ones. Intuitively, this result makes sense. It is simply a hodge-podge of features tagged either R or NR. When the threat features and the mitigation features are comingled, the model does not know which features represent the threat and which represent the mitigation. Also, there is no reason to expect that, for example, a relevant T-M pair for Threat 49 will have anything in common with a relevant T-M pair for Threat 268 to indicate that they are both of class R since they express totally different concepts.

Trial 21

In Trial 21, we constructed a model based on (Gee, 2003)[6] to try to select the threat T to which a new mitigation M is relevant from among all threats in the corpus based on M's similarity to labeled mitigations already known to be relevant to T. We used LSA to create a semantic space of the mitigations mapped to all the standard threats in the corpus and we also created an index of which mitigations are labeled relevant to each threat. In this classifier threat id is the dependent variable. When a new mitigation document is presented, it is used as a query against the semantic space, returning a ranked list of other mitigation documents similar to the query from most similar to least. The trial 21 model classifies the new document in three stages. First, it is classified according to the class of its nearest neighbor in the space (i.e. it is assigned the threat id associated with the existing mitigation document whose similarity score is highest).

[6] Note this model is not the same as the one discussed in the "per threat" section, trials 14 and 19.

Next, the new mitigation document is classified according to the class (threat id) associated with of the majority of all mitigations in the ranked list truncated at an arbitrary cut-off N. Finally, if the majority and nearest neighbor stages agree, the new mitigation document is deemed to be relevant to the threat to which its nearest neighbor is relevant. If the majority and nearest neighbor stages do not agree, the dispute is settled by the third stage which attempts to detect the skew of the new document towards one class or the other. We implemented the first two stages (but not the tie-breaker) and tested the results with 11 representative mitigations. Of these, the model classified 3 mitigations as relevant to the correct threat, 7 to an incorrect threat, and 1 resulted in a tie, which we count as incorrect in the absence of tie-breaker logic. These results were so poor that we did not invest any time in developing a tie-breaker, since it would only come into play a small percentage of the time. This result was more of a brain teaser than the prior trial, but in the final analysis it also made intuitive sense. Given a threat, for example, breach of physical access, we may have mitigations that describe a fence, a wall, a moat, and drone surveillance and each of these mitigations will furthermore describe how they mitigate the threat. If we present a new mitigation, for example, an armed guard, which also describes how it mitigates the threat, the mitigation itself (armed guard) is not very similar to any of the other mitigations (fence, wall, moat, drone) for the threat. Even though all the listed mitigations may present as similar to the threat, the inverse is not necessarily true.

APPENDIX E: CYBER RISK ASSESSMENT

A number of cyber risk assessment methodologies are described in the literature and in use today. These include:

- Carnegie Mellon Software Engineering Institute Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) (Caralli et al., 2007)
- ISACA Risk IT Framework based on Control Objectives for Information and Related Technologies (COBIT) (ISACA, 2009; Schmittling, 2010)
- Johns Hopkins University Applied Physics Laboratory Mission Information Risk Analysis (MIRA) (Llanso et al., 2012) (Llanso et al., 2013)
- Johns Hopkins University Applied Physics Laboratory BluGen (Llanso et al., 2017)
- Mitre Crown Jewels Analysis (CJA) and Threat Assessment and Remediation Methodology (TARA) (MITRE, 2015)
- US National Institute of Standards and Technology (NIST) Special Publication 800-30: Guide for Conducting Risk Assessments (National Institute of Standards and Technology, 2012)
- Automated Risk and Utility Management (AURUM) (Fenz et al., 2011)

These were selected because they are representative of approaches in use by organizations that employ formal cyber risk assessment processes and because descriptions are available in open literature. Note that this is not an exhaustive survey of such methodologies, and in particular does not include proprietary and other closed-source methodologies.

OCTAVE is an eight-step process, as follows. First, impact areas (e.g. financial, productivity, reputation, health, etc.) are identified and ranked. Next critical information assets are identified as well as IT and non-IT locations where critical information is processed and stored. Then situations that could affect the critical information are enumerated and threat scenarios (including asset, actor, access, motive, and outcome) are identified. The consequences of identified threat scenarios are assessed to point out risks. An aggregate score is derived for each identified threat/consequence by assigning qualitative impact values (e.g. high, medium, low) to each identified threat/consequence for each impact area, multiplying by the rank of the impact area, and summing the products. Finally, a relative risk matrix is developed based on

probability of occurrence (high, medium, low) and score ranges, then mitigation approaches are selected based on the risk matrix.

Similarly, Risk IT, advocates identifying risk scenarios derived from understanding of business objectives, where each scenario considers the potential actor (insider, competitor, etc.), threat type (malicious, accidental, etc.), event type, asset or resource affected, and time. Likewise, CJA+TARA considers mission priorities and potential impacts due to cyber, identifying the potential threats faced by each individual asset based on common attack patterns cataloged in CAPEC (MITRE, 2017a), scoring (on a scale of 1-5) each threat in multiple dimensions, and aggregating to produce a risk score per asset. Additionally, the NIST risk assessment process is a 5-step process, including: (1) identify possible threat sources and events, (2) identify inherent vulnerabilities and predisposing conditions present in the system, (3) determine likelihood of occurrence of events, (4) determine magnitude of impact of each event occurrence, and (5) determine risk as a combination of likelihood of occurrence and impact. The AURUM Framework follows the NIST risk assessment process and also includes automated control recommendations.

In MIRA, two sets of risk scores are expert-generated. First, experts judge mission impact for each viable combination of mission, system asset, data type, and cyber effect (confidentiality, integrity, or availability). Also, expert input for adversary level of effort (LOE), the amount of effort and/or resources an adversary would have to apply to realize the effect, is required for each viable combination of asset, data type, cyber effect and attack vector. Risk is then visualized by plotting the mission contexts on an x-y plot such that those with the highest mission impact (x) and lowest LOE (y) are the highest priority candidates for mitigation.

BluGen takes a capability-centric approach based on an expert-constructed reusable knowledge resource called the Reference Catalog. In this catalog, threats are mapped to asset types in a taxonomy and mitigations are mapped to threats. Consistent with event-centric approaches, like MIRA, OCTAVE, CJA, Risk IT, and AURUM, BluGen intakes a description of the system being assessed, including assets, data types, and mitigations already present. BluGen requires a set of raw criticality scores, one for each viable combination of mission, asset, data type, and cyber effect. BluGen estimates risk from these scores and the threat-asset type mappings. To the extent that threat-mitigation mappings exist in the Reference Catalog, BluGen is the only method discussed here that recommends mitigations; however, the catalog

is still in its infancy. Ongoing construction of the BluGen Reference Catalog could benefit from an automated approach to mapping mitigations to threats. Table E.1 summarizes the risk assessment methods discussed above.

Table E.1. Asset-based, Threat-informed Cyber Risk Assessment Methods

Method	**Characterize System**	**Characterize Mission**	**Characterize Threat**	**Assess Risk**
AURUM (Fenz et al., 2011)	Assets	Magnitude of impact of adverse events	Threat sources and events; inherent vulnerabilities; likelihood of occurrence	Aggregation of combined likelihood of occurrence and impact
BluGen (Llanso et al., 2017)	Assets, data, existing mitigations	Mission weights, criticality scores per mission/asset/ data/cyber effect	Adversary's anticipated offensive capabilities	Asset exposure based on existing mitigations and Reference Catalog mappings, asset criticality based on aggregation of individual criticality scores
CJA+TARA (MITRE, 2015)	Assets	Mission priorities	Potential threats by asset, scored based on common attack patterns	Aggregation of scores

Method	Characterize System	Characterize Mission	Characterize Threat	Assess Risk
MIRA (Llanso et al., 2012, 2013)	Assets, data, connectivity	Mission impact per asset/data/ cyber effect	Adverse events scored by required adversary LOE per asset/data/ cyber effect/attack vector	x-y plot of assets by mission impact and LOE
NIST SP 800-30 (National Institute of Standards and Technology, 2012)	Assets	Magnitude of impact of adverse events	Threat sources and events; inherent vulnerabilities; likelihood of occurrence	Aggregation of combined likelihood of occurrence and impact
OCTAVE (Caralli et al., 2007)	Assets, locations, information	Areas of impact, consequences	Threat scenarios (asset, actor, access, motive, and outcome)	Aggregation of scores for each identified threat/ consequence
RISKIT (ISACA, 2009; Schmittling, 2010)	Assets	Business objectives	Threat scenarios (asset, actor, motive, time), frequency and magnitude of impact of occurrences	Aggregation of magnitude of impact

APPENDIX F: MITIGATION OPTIMIZATION APPROACHES

The CSE faces two main problems when selecting a security control portfolio to address an organization's cyber risk. First, there may be multiple conflicting objectives to be considered (e.g. cost, ease of use) making it impossible to arrive at a single optimal solution. At the same time, the number of combinations of viable alternatives presents an overwhelmingly large search space, requiring strategies to winnow it down to a tractable scope. These decisions are complex and inexact, involve multiple stakeholders with diverse interests, and require trade-offs between conflicting objectives. Moreover, information environments, risk tolerance levels, and the threats they face vary widely from one organization to the next. (Kiesling et al., 2016) Hence, compromise solutions must be sought. There is a large body of research which applies multi-criteria decision-making (MCDM) techniques to solve the mitigation optimization problem. In addition, a few authors have applied game theory to the problem. We discuss these below and summarize them in Table F-1.

Multi-Criteria Decision-Making Approaches

Multi-criteria decision-making (MCDM), also known as multiple-criteria decision analysis (MCDA), is widely applied to security portfolio selection (Fenz et al., 2011; Llansó et al., 2019; Patterson et al., 2013; Sawik, 2013; Schilling & Werners, 2016; Weishäupl, 2017; Yevseyeva et al., 2015). MCDM is discipline for evaluating multiple conflicting criteria. It is used to analyze problems where these are some measures of costs and benefits which can be traded off to arrive at the best solution under the given constraints. Researchers investigate a number of MCDM techniques for this problem, some of which include or are based on fuzzy set theory (Otero, 2014), multi-attribute utility theory (i.e. value functions, knapsack strategy) (Fielder et al., 2016; Panaousis et al., 2014; Shapasand et al., 2015; Smeraldi & Malacaria, 2014), evolutionary multi-objective optimization (EMO) also known as genetic algorithms (Gupta et al., 2006; Kiesling et al., 2016, 2012; Rees et al., 2011; Sarala et al., 2016; Viduto et al., 2012), analytic hierarchy process (AHP) (El-Gayar & Fritz, 2010), grey relational analysis (GRA) (Breier & Hudec, 2013), simple additive weighting (SAW) (Llanso, 2012; Llansó et al., 2019), the technique for order preference by similarity to ideal solution (TOPSIS) (Breier &

Hudec, 2013), and preference ranking organization method for enrichment evaluation (PROMETHEE) (Lv et al., 2011).

(Fenz et al., 2011) describe an automated approach to mitigation selection that requires as input an enumeration of relevant potential controls, risk level of the protected asset, and control attributes, such as cost and effectiveness. Their method defines mitigation selection in terms of a multi-objective combinatorial optimization problem which seeks to select controls by analyzing alternatives in consideration of the stakeholder's objectives, such as risk reduction, cost, availability, and reliability to choose Pareto-efficient combinations. They provide a user interface where each objective is represented by a slider, allowing the stakeholder to tune the upper and lower bounds of his objectives and obtain immediate feedback.

(Patterson et al., 2013) describe a method for optimizing security control decisions for critical infrastructure systems. Given a fixed budget, the method balances costs and benefits of improving three dimensions of cybersecurity, intrusion prevention, detection, and response by posing the selection as an optimization problem. The goal of the optimization is to select the investment strategy that yields the smallest residual probability of successful attack, i.e. the best security portfolio for the budget. This optimization problem requires models of the system under analysis, cost and performance of applicable security controls, and risk. The authors note that creating the models presents a large challenge for future work.

Given an enumeration of threats and potential mitigations, (Sawik, 2013) describes a bi-objective mixed integer trade-off model to select an optimal countermeasure portfolio balancing expected and worst-case losses. The model applies conditional value-at-risk (CVaR) and scenario-based analysis to select controls by considering desired confidence, expected loss, budget, and risk tolerance.

(Schilling & Werners, 2016) present a combinatorial optimization model for optimal selection of security controls. Unlike most models, which are based on cost minimization, this model minimizes the number of controls as a proxy for cost. The authors decided to do this because it eliminates the need to collect cost data on all the candidate solutions before selecting a solution. Their idea is to cost out the selected solution and if the cost is too high, rerun the model after reducing the number of controls.

(Weishäupl, 2017) describe a multi-objective optimization model for control selection which seeks to minimize control cost while maximizing security level. Overall security level is

computed as the sum of the security levels of individual assets weighted by importance. Each asset's security level is inversely proportional to the severities of the vulnerabilities by which it is affected accounting for probability of occurrence. Cost is the sum of initial costs (e.g. purchase, set-up), operating costs (e.g. annual fees and ongoing maintenance), and costs associated with security breaches (e.g. disruption of business, damage, reputation, decline in stock price).

(Yevseyeva et al., 2015) present two formulations of security control selection based on quadratic integer programming based on a traditional risk vs return model common in financial portfolio selection. A multi-objective formula seeks to minimize risk (based on probability of successful attack) and maximize return (by minimizing expected losses due to cyber breach) while simultaneously satisfying a budget constraint. A single-objective formula is derived from the multi-objective formula by assuming that both the return and the budget are constrained.

(Yevseyeva, Fernandes, Van Moorsel, Janicke, & Emmerich, 2016) seek to apply the Sharpe ratio common in financial analysis to security control selection based on a fixed budget and two objectives, risk and return. Maximizing the Sharpe ratio supports computation of efficient portfolios while balancing the objectives in an optimal way.

In his doctoral dissertation, (Otero, 2014) describes creation of an artifact based on fuzzy set theory and constructed using the MATLAB Fuzzy Logic Toolbox. Taking four input variables for each security control under consideration - estimated implementation cost, scope (number of assets protected), extent of compliance with laws and regulations, and effectiveness in addressing the risks - Otero's artifact includes fuzzy "if-then" rules and membership functions defining objectives and constraints developed in consultation with cybersecurity experts and based on the literature. Execution of the rules results in a set of selected controls. The design of the rules and functions in the artifact is based on expert responses to a survey that asks experts to identify the existing controls in place in their organization, rank the 11 ISO/IEC 2702 information security areas by order of importance to the organization, rate the detailed list of security controls in their top three security areas on cost, scope, compliance, and effectiveness.

(Panaousis et al., 2014) model the cybersecurity posture of an organization and then present a series of non-cooperative control-games where each game is between the defender (a single control) and the attacker. The Nash Equilibria of the games is derived in consideration

of organizational preferences such as costs, anticipated threats, and asset importance. A multi-objective, multi-choice knapsack approach is then used to optimize investment in controls within the organization's budget.

(Shapasand et al., 2015) apply a knapsack model for control selection with budget as the constraint. Consts considered in this model include cost of maintaining desired levels of C/I/A, profit reduction due to C/I/A compromise, and penalty cost (e.g. fines, reputation) due to C/I/A compromise. (Smeraldi & Malacaria, 2014) describe a combinatoric optimization algorithm based on variations of the knapsack problem that can also account for mitigations that benefit more than one asset and mitigations that, when applied together, provide more benefit than the sum of their individual benefits.

(Kiesling et al., 2012) describe a decision support framework for security control selection consisting of three stages. In the modeling stage assets, threats, and available controls are identified. In the second stage, a baseline risk assessment is determined through simulation. Finally, Pareto-efficient control portfolios are computed via multi-objective optimization.

(Kiesling et al., 2016) describe Multi-Objective decision Support in Efficient Security Safeguard Selection (MOSES3), a collaborative decision support process that enables cybersecurity professionals and strategic decision makers to "bridge the gap between strategic security investment and operational implementation decisions." After describing the system architecture (assets, data, access), identifying threats and attacker skill level, and enumerating existing controls, assets are valued according to their criticality by C/I/A and candidate mitigations per asset and are specified. An attack-based simulation seeks to estimate a set of Pareto-efficient security control portfolios, optimizing via a genetic algorithm while minimizing the specified objectives (cost, C/I/A impact, undetected rate, target reached rate). Each portfolio is evaluated by initializing the system model with the given set of controls then simulating attacks and aggregating attack outcomes.

(Gupta et al., 2006) present a genetic algorithm approach for selecting a security profile that minimizes cost while also minimizing the number of unmitigated vulnerabilities. (Rees et al., 2011) present a decision support system which uses a genetic algorithm to determine an optimal combination of countermeasures by trading off cost versus residual risk where risk is calculated as the sum for all anticipated threats of the number of occurrences expected annually and the expected cost of each occurrence.

(Sarala et al., 2016) describe an approach to optimizing control selection where solutions must observe a budgetary constraint and solution cost must not exceed the anticipated losses if threats were left unmitigated. Their approach solves a multi-objective problem by applying TABU search combined with genetic algorithm. The objectives, to maximize the number of vulnerabilities addressed while minimizing the cost of the solution, are first processed via the TABU search to arrive at a set of Pareto-efficient solutions. These serve as the initial input to a genetic algorithm

(Viduto et al., 2012) apply the evolutionary algorithm known as Multi-Objective Tabu Search (MOTS) for selecting security controls as a multi-objective optimization problem balancing financial costs (purchase, operational, training, and labor) and residual risk. The MOTS algorithm was shown to arrive at a Pareto-efficient set more rapidly than the exhaustive search method with similar quality solutions.

(El-Gayar & Fritz, 2010) describe a collaborative multi-perspective decision support system (DSS) based on AHP and stakeholder input. The decision model is comprised of assets, threats, and controls expressed as a set of vectors and analysis subspaces representing the pairwise interactions, e.g. threat-asset, threat-control, and asset-control. Stakeholders may be assigned unequal weights. They express judgments of the pairwise interactions. Judgments are aggregated using the weighted arithmetic mean to provide a ranked list in order of importance.

(Breier & Hudec, 2013) describe a quantitative prioritization of security controls based on asset valuation and the threats identified by an a priori risk assessment. Their method uses GRA combined with the TOPSIS, taking as inputs asset importance (financial values), threat data (impact, to which assets, probability of occurrence), and potential security controls (purchase price, difficulty of implementation, maintenance cost, efficiency, applicable to which threats). The security control alternatives are evaluated based on cost, efficiency, and protection against the most significant threats and the top n are selected.

Cyber Investment Analysis Methodology (CIAM) (Llanso, 2012) combines data about the infrastructure to be protected (key hardware, software, people, and processes), incident data (vulnerabilities, attack steps, and frequency) related to the infrastructure, potential security controls including cost to install and maintain, possible business impacts of cyber events, and control weightings (effectiveness) to compute an initial selection of security controls and investment prioritization. A SAW algorithm combines the incident data, effectiveness scores,

control costs, and impact data to compute a list of controls in relative priority order. The list can be used to select controls in the content to an overall cyber security budget.

(Lv et al., 2011) describe a multi-criteria ranking model based on PROMETHEE method. Its accepts a finite set of security controls and a set of evaluation criteria (e.g. purchase cost, operating/maintenance cost, effectiveness, alignment with standards) as inputs, then ranks security controls quantitatively. Evaluation criteria must be numeric, but they can have various units and some may be minimized while others are maximized in order to identify a set of controls that optimizes all the criteria.

(Llansó et al., 2019) describe a SAW-based mitigation selection approach that uses a set of weighted criteria and a a capability-based representation for cybersecurity mitigations. The security engineer sets the weights based on organizational priorities and constraints and the algorithm recommends a candidate set of mitigations representing a "practical middle ground between completely ad hoc mitigation selection approaches" and "approaches whose computational complexity requires the use of sophisticated heuristic algorithms."

Game Theoretic Approaches

Several authors apply game theory to security portfolio selection in combination with MCDM techniques. (Fielder et al., 2016) employs a pure game theoretic approach in a single massive two-person non-cooperative zero-sum static game where the defender (person in charge of choosing controls) competes against an attacker who chooses among various attack targets. The Nash equilibrium of the game represents the best control portfolio. Recognizing that the organization may not have sufficient budget to implement the equilibrium of the pure game, they also discuss a hybrid approach combining game theory with a knapsack strategy. (Panaousis et al., 2014) model the cybersecurity posture of an organization and then present a series of non-cooperative control-games where each game is between the defender (a single control) and the attacker. The Nash equilibria of the games are derived in consideration of organizational preferences such as costs, anticipated threats, and asset importance. A knapsack approach is subsequently used to optimize investment in security controls within the organization's budget. Finally, (Wang & Zhu, 2016) used evolutionary game theory to investigate long-term cybersecurity investment strategy finding that firms will invest as long as either the cost to invest is low or the cost of a breach is high.

Table F.1. Selected Mitigation Optimization Approaches

Method	**Inputs**	**Analysis Approach**
(Barnard & von Solms, 2000)	Business analysis Security requirements / policy **Potential security controls** Evaluation criteria	Flow-based control selection model
(Breier & Hudec, 2013)	Asset financial values Threats to assets **Potential countermeasures** Countermeasure cost, efficiency	GRA combined with TOPSIS
(El-Gayar & Fritz, 2010)	Assets Threats **Controls** Weighted stakeholder judgments	Analytic hierarch process
(Fielder et al., 2016)	Threats **Controls** Degrees of control implementation	Game theory: two-person non-cooperative zero-sum static game combined with MCDM knapsack strategy
(Fenz et al., 2011)	**Potential controls** Risk level of the protected asset Control attributes such as cost and effectiveness	MCDM multi-objective combinatorial optimization (Pareto efficiency)
(Gupta et al., 2006)	**Controls** Cost Unmitigated vulnerabilities	Evolutionary multi-objective optimization / genetic algorithms
(Kiesling et al., 2012)	Assets Threats **Controls**	MCDM: Pareto efficiency

Method	Inputs	Analysis Approach
(Kiesling et al., 2016)	System architecture (assets, data, access) Threats and attacker skill level **Existing controls** Assets valued according to their criticality by C/I/A Candidate mitigations per asset	Evolutionary multi-objective optimization / genetic algorithms
(Llanso, 2012) Cyber Investment Analysis Methodology (CIAM)	Assets to be protected Incident data related to the assets **Potential security controls** Installation and maintenance cost Control effectiveness Possible business impacts of cyber events	Simple additive weighting: cost/benefit algorithm
(Lv et al., 2011)	**Potential security controls** Evaluation criteria (cost, effectiveness, organizational priorities)	Multi-criteria ranking, PROMETHEE
(MITRE, 2017c) Cyber Risk Remediation Analysis (RRA)	**Table of countermeasures per threat** Cost of countermeasures	High to low ranking by cost
(Otero, 2014)	**Potential security controls** (implementation cost, scope, extent of compliance, effectiveness)	MCDM: Fuzzy logic / fuzzy set theory
(Panaousis et al., 2014)	**Potential controls** Organizational preferences such as costs, anticipated threats, and asset importance	Game theory non-cooperative control-games combined with MCDM multi-attribute utility theory

Method	Inputs	Analysis Approach
(Patterson et al., 2013)	Model of the system under analysis **Applicable security controls** Control cost and performance Risk Budget	MCDM
(Rees et al., 2011)	Potential controls **Control cost** Residual risk after control Anticipated threats and annual rate of occurrence	Evolutionary multi-objective optimization / genetic algorithms
(Sarala et al., 2016)	**Potential controls** Budgetary constraint (maximum acceptable control portfolio cost) Anticipated financial loss if threats left unmitigated Vulnerabilities	Evolutionary multi-objective optimization / genetic algorithms
(Sawik, 2013)	Threats **Potential mitigations** Expected loss Budget Risk tolerance Potential mitigations	MCDM: bi-objective trade-off model
(Schilling & Werners, 2016)	**Potential controls** Number of controls as a proxy for cost	MCDM: combinatorial optimization

Method	Inputs	Analysis Approach
(Shapasand et al., 2015)	**Potential controls** Cost of maintaining desired levels of C/I/A, Profit reduction due to C/I/A compromise Penalty cost (e.g. fines, reputation) due to C/I/A compromise	MCDM: multi-attribute utility theory, knapsack model
(Smeraldi & Malacaria, 2014)	**Mitigations** Applicability to multiple assets	MCDM: multi-attribute utility theory, knapsack model
(Viduto et al., 2012)	**Potential mitigations** Financial costs (purchase, operational, training, and labor) Residual risk	Evolutionary multi-objective optimization / genetic algorithms
(Wang & Zhu, 2016)	**Potential controls** Control cost Anticipated losses due to unmitigated cyber breach (including reputation)	Evolutionary game theory
(Weishäupl, 2017)	**Potential controls** Control costs Security level provided by controls Asset importance Vulnerability severity per asset	MCDM: multi-objective optimization

Method	Inputs	Analysis Approach
(Yevseyeva et al., 2015)	**Potential controls** Risk (probability of successful attack) Anticipated losses due to cyber breach Control effectiveness in reducing loss Budget constraint	MCDM: quadratic integer programming
(Yevseyeva et al., 2016)	**Potential controls** Risk Return (anticipated loss minus control effectiveness) Budget constraint	Sharpe ratio