Innovative Supply Chain Cyber Risk Analytics: Unsupervised Clustering and Reinforcement Learning Approaches

by

Benjamin M. Siegel

B.S., United States Military Academy (2021)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Author . . . .
Sloan School of Management
May 12, 2023

Certified by . . . .
Retsef Levi
J. Spencer Standish (1945) Professor of Operations Management
Thesis Supervisor

Accepted by . . . .
Georgia Perakis
William F. Pounds Professor of Management Science
Co-Director, Operations Research Center

Innovative Supply Chain Cyber Risk Analytics: Unsupervised Clustering and Reinforcement Learning Approaches

by

Benjamin M. Siegel

Submitted to the Sloan School of Management
on May 12, 2023, in partial fulfillment of the
requirements for the degree of
MASTER OF SCIENCE IN OPERATIONS RESEARCH

Abstract

The increasing frequency and severity of cyberattacks has made reliable cyber risk assessment a critical concern for organizations worldwide. Traditional cyber risk methodologies focus on the enterprise's level of cyber maturity. Moreover, several commercial companies provide cyber ratings using information about the organization accessible by outside parties, often called outside-in ratings. However, merely focusing on the enterprise's own cyber maturity may be insufficient given the increasing number of cyberattacks that exploit vulnerabilities in the organization's supply chain. This thesis presents innovative approaches to cyber risk assessment that incorporate attributes of the digital supply chain.

Chapter 2 is motivated by recent cyberattacks that relied on compromising software companies as a vector to attack their customers, illustrating the importance of going beyond the enterprise's vulnerabilities and assessing potential threats from the supply chain. Taking into account this observation, the chapter presents a data-driven approach to identifying high risk software companies based on their relative position in the supply chain. The newly proposed approach is based on unsupervised clustering techniques applied to intuitive supply chain features of the respective software companies. The clustering approach is applied to a self-constructed dataset of over 4,600 software companies, and the model partitions the software companies into two clusters. Historical breach data that was not used in the clustering suggests that the second cluster, despite being smaller, has a significantly higher proportion of breached companies. Furthermore, feature differences between clusters reveal that the risky software companies tend to have many more customers and suppliers, particularly in the Technology and Business Services sectors. These findings highlight the importance of specific supply chain features as risk drivers in assessing the cybersecurity posture of software companies.

In Chapter 3, we propose a novel approach to cyber risk assessment that directly incorporates an attacker model and in so doing are able to better predict enterprises' vulnerabilities. We develop a theoretical attacking agent to randomly target a company and explore neighboring nodes in the supply chain graph. Deep reinforcement learning algorithms are used to train the attacker over time, identifying rewarding paths throughout the supply chain network. The fully trained attacker then simulates attacks, yielding a risk score for each individual company in the network. This score corresponds to the relative number of breaches the company experiences in simulation. This approach is empirically validated using a dataset of over 13,000 companies in the Retail sector, and the results are highly statistically significant when compared to real-world breach incident data and an existing outside-in ratings model. Because the theoretical attacker approach is validated by existing breach data and holds predictive power, this methodology can contribute to the development of more effective risk assessment strategies to combat the growing threat of cyberattacks.

Thesis Supervisor: Retsef Levi
Title: J. Spencer Standish (1945) Professor of Operations Management

Acknowledgments

I would like to thank my advisor, Professor Retsef Levi, for his invaluable mentorship. From the very first time we met, Retsef has gone out of his way to ensure my success and find research opportunities that aligned perfectly with my interests in the field of cybersecurity. During our research projects, he always provided unique insights and creative solutions that greatly influenced the direction of this thesis, particularly in uncovering supply chain weaknesses. His exceptional work ethic has been an inspiration to me and many other students, and I would like to thank him for being an outstanding role model throughout my academic journey over the past two years.

Several other individuals also had a significant impact on this thesis. I am immensely thankful to Kevin, Ghali, and Rafi for their continuous support and direction, ranging from insightful cybersecurity discussions to invaluable modeling assistance. I would also like to extend my gratitude to Lincoln Laboratory for providing me with the opportunity to study at MIT, as well as the members of my Lincoln team for expanding my knowledge of cyber events and reinforcement learning in general.

I am also deeply appreciative of the many individuals who have continuously impacted my life. My colleagues and friends at the Operations Research Center are some of the most interesting and intelligent individuals I have had the privilege to know. To my new friends in Boston and connections from the past, I am grateful for your friendship and hope to see you again soon. Finally, a special thank you to my family for their unwavering support, especially my brothers Ethan and Noah, who are a constant source of motivation to strive for excellence in all that I do.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The increasing frequency and severity of cyberattacks (9) has made reliable cyber risk assessment a critical concern for organizations worldwide. Traditional cyber risk methodologies focus on the enterprise's level of cyber maturity. This information can be obtained by evaluating an organization's cyber controls, processes, and procedures with respect to accepted security principles (39). To obtain an even more thorough assessment, organizations sometimes also conduct exercises such as penetration testing and red teaming that simulate real-world attacks on internal networks to identify vulnerabilities that attackers might exploit (2). However, performing a thorough assessment of an organization's network design or conducting simulation exercises require a significant amount of time from highly trained cybersecurity professionals and cannot be performed at scale (36). To address this challenge, several commercial companies have emerged offering outside-in ratings as an alternative approach. Outside-in ratings are based on information that can be obtained from the public domain about an organization's digital footprint, such as its vulnerability patching frequency and open port policies. Companies that offer outside-in ratings often use machine learning algorithms to process this data and generate a rating that represents an organization's overall cyber maturity (7). Several studies provide some evidence that suggest outside-in ratings are correlated with the probability of experiencing a cyberattack (11, 5). However, merely focusing on the internal cyber maturity of companies may not provide a comprehensive assessment of risk, since an increasing number of cyberattacks have exploited trusted relationships in the digital supply chain to create innovative attack vectors even against otherwise strongly defended companies. In other words, even if an organization has the most robust cybersecurity practices in place, their data, customers, and reputation may still be at risk due to the actions or vulnerabilities of a company in their supply chain (21). This observation has been underscored by recent high-profile events. For example, the SolarWinds incident was a massive cyberattack in 2020 that demonstrated the significant risk that third-party suppliers can pose to organizations in the digital supply chain, as the attackers inserted malicious code into SolarWinds' IT management software, affecting thousands of organizations including U.S. government agencies and private companies (13). To complement the existing risk assessment approaches and address the existing gaps, this thesis presents innovative approaches and methodologies of cyber risk assessment that directly account for potential risk drivers emerging from the enterprise's digital supply chain attributes.

1.1 Thesis Results

Chapter 2

Chapter 2 is motivated by the SolarWinds incident, whereby a prominent software company was used as a vector to attack its customers, showing that traditional risk assessment methodologies are insufficient should they not examine vulnerabilities throughout an enterprise's supply chain. To provide quantitative evidence of this observation, the chapter presents a data-driven approach to identifying high risk software companies based on their relative position in the supply chain and related attributes. The approach is based on unsupervised clustering techniques applied to intuitive supply chain features of the respective software companies. The clustering approach is applied to a self-constructed dataset of over 4,600 software companies, and the model partitions the software companies into two clusters. Historical cyber breach data that was not used in the clustering suggests that the second cluster, despite being smaller, has a significantly higher proportion of breached companies, including SolarWinds. Furthermore, an examination of the most significant feature differences between clusters reveals that software companies in the risky cluster tend to have many more customers and suppliers, particularly in the Technology and Business Services sectors. These findings highlight the importance of specific supply chain attributes as risk drivers in assessing the cybersecurity posture of software companies, and provide regulators several key performance indicators for assessing their own company's internal enterprise risk.

Chapter 3

Motivated by the fact that supply chain vulnerabilities can pose significant cybersecurity risks to an organization, Chapter 3 introduces a novel approach to a scalable cyber risk assessment methodology that incorporates the global digital supply chain to model attacker behavior. A theoretical attacking agent is developed, which randomly targets a company and explores neighboring nodes in the supply chain graph. The attacker gains utility based on the size of any newly breached companies but must operate under certain realistic constraints. For example, successful breach probability depends on the size of the targeted company, with smaller companies potentially having weaker defenses. The directionality of the traversed edge and type of product or service provided between companies also influence the spread of attacks. Deep reinforcement learning algorithms are used to train the attacker over time, identifying rewarding paths throughout the supply chain network. The fully trained attacker then simulates attacks, yielding a risk score for each individual company in the network. This score corresponds to the relative number of breaches the company experiences in simulation. This approach is empirically validated using a dataset of over 13,000 companies in the Retail sector, and the results are highly statistically significant when compared to real-world breach incident data. Comparison with an existing outside-in ratings model also demonstrates similar out-of-sample cyberattack detection power. Furthermore, when the simulated risk scores are added as an additional feature to the existing model, the combined model shows improved performance, indicating that the risk scores contribute valuable supply chain risk information beyond the outside-in ratings. The validated theoretical attacker approach presented in this chapter offers a new tool for assessing supply chain cyber risk that does not require propriety risk ratings or disclosure of censored historical breaches and can contribute to the development of more effective risk assessment strategies to combat the growing threat of cyberattacks.

Chapter 2

Detecting High Risk Software Companies via Unsupervised Clustering Based on Supply Chain Features

2.1 Introduction

On December 13, 2020, thousands of American companies and government organizations experienced one of the most sophisticated and widespread computer hacks in history. Over 18,000 known entities were affected, including several high-profile companies such as Microsoft, Intel, Nvidia, Cisco, and FireEye (45). Additionally, the U.S. government reported that the attack affected federal, state, and local governments across the country, as well as at least nine federal agencies (18). Although the full extent of the attack is still unknown even today, it can be traced back to a Texas-based company called SolarWinds, a large-scale software company that provides a range of IT management and monitoring software solutions to organizations of all sizes. Its customer base spans over 190 countries and includes more than 320,000 clients, among which are 499 companies featured in the Fortune 500 list (13). The attack involved the insertion of a malicious code into the SolarWinds Orion software update, which is widely used by organizations for network management. When customers downloaded their Orion software update, they unknowingly installed malware, giving hackers access to sensitive data and systems.

Although the SolarWinds cyberattack has unprecedented scale and impact, it is perhaps more important because of the manner in which the attack was carried out. The attackers compromised SolarWinds' trusted system, allowing them to insert malicious code into software updates that were distributed to customers. This novel attack technique leveraged the digital supply chain and is particularly insidious because it highlights new cyberattack vectors that pose risks even to otherwise strongly defended organizations. Since SolarWinds, there have been a plethora of publicly known cyber incidents including Kaseya (27), Dependency Confusion (16), and Codecov (40) where attackers gained access to their targets by exploiting trusted digital supply chain connections.

Motivated by these recent high-profile cyberattacks that leveraged the digital supply chain to create innovative attack vectors, this chapter takes a data-driven approach to identifying high risk software companies based on their relative location in the supply chain network and related attributes. Specifically, this chapter is based on a comprehensive self-constructed dataset that covers over 4,600 software companies. The dataset includes for each entity their organizational characteristics and publicly available digital supply chain relationships. Using unsupervised machine learning techniques applied to natural digital supply chain features, the companies can be partitioned into two clusters. Analysis based on historical cyber breach incident data that were not included in the clustering shows that there is a 278% relative difference between the two clusters in the proportion of breached companies, and that the risky cluster specifically includes SolarWinds. Furthermore, an examination of the most significant feature differences between clusters demonstrates that software companies in the risky cluster tend to have many more customers and suppliers, particularly in the Technology and Business Services sectors. These findings suggest that supply chain attributes are important risk drivers in assessing risk for software companies.

These findings are of significant practical and academic importance as they provide a valuable tool for assessing the risk of software companies that provide services to an enterprise. Notably, the newly developed approach does not rely on propriety internal data or historical breach disclosure, but still provides companies and organizations with actionable insights to assess risk and make informed decision in proactively managing their own supply chain cybersecurity and most importantly assess potential risk from software providers.

2.2 Literature Review

This chapter describes an unsupervised machine learning clustering approach to assess the risk of software companies being compromised and used as a vector to launch cyberattacks. The history of cyber risk assessment methodologies can be traced back to the early days of computing. In the 1960s and 1970s, computer systems were primarily used by large organizations, governments, and military establishments. The primary focus of security was on physical security and protecting against hardware failures. As computer networks became more prevalent in the following decades, the focus of security shifted to protecting data and information. These concepts led to a 1977 publication by the National Institute of Standards and Technology that introduced the widely popular CIA triad of confidentiality, integrity, and availability as a framework to guide data security (33). At a high level, these principles inform security policies by ensuring data is private, unaltered, and accessible. Many organizations and companies adopted these principles to develop their own corporate and organizational cybersecurity strategies. The industry cybersecurity strategies have evolved over the years, but typically involve a set of best practices such as implementing firewalls, two-factor authentication, and/or employee training programs. Additionally, current cyber risk assessment methods mainly rely on measuring and assessing compliance with respect to the set of accepted best practices. For example, the most recent Verizon Data Breach Investigations Report, which provides an annual analysis on the state of cybersecurity and data breaches around the world, indicates current risk assessment methodologies primarily focus on adherence to internal security protocols (3).

Parallel to the growth of internal defense best practices, some modern organizations have started trying to quantitatively assess cyber maturity. Over the past decade, industry stakeholders such as BitSight started developing cyber ratings to offer data-driven measures of cyber maturity and risk for individual companies. Like other cyber ratings providers, BitSight calculates its ratings using externally observable cybersecurity information about organizations, such as vulnerability patching frequency and open port policies. By aggregating individual ratings across multiple best practices, BitSight ultimately creates numerical "outside-in" risk ratings of individual companies, somewhat similar to the way credit ratings and FICO scores provide a numerical measure of credit risk (6). The study in (5) across thousands of organizations showed there are statistically significant correlations between specific BitSight outside-in ratings and the likelihood of a cybersecurity breach.

Yet, many recent high-profile events such as the SolarWinds incident have illustrated that merely focusing on internal policies is not sufficient to protect against cyberattacks since most companies today are digitally connected and interacting with external organizations which give rise to new vulnerabilities and attack vectors. This is further supported by the authors in (8) who argue that cyber-supply chain management "has emerged as a critical discipline" in the current digital ecosystem and perform a decade-long study on the impact of an organization's adoption of policies outlined in the U.S. National Institute of Standards and Technology cybersecurity framework. The analysis reveals that certain policies and practices are closely linked to more effective control of breaches originating from the supply chain. More recently, Hu et al. (19) developed an integrated dataset with company attributes, BitSight outside-in ratings, and supply chain relationships for more than thirty-eight thousand companies in the major Healthcare, Retail, and Oil and Gas sectors. The main result of the paper is to show that a machine learning model incorporating supply chain features significantly improves out-of-sample AUC compared to baseline model including only company attributes and BitSight outside-in ratings. This emerging research demonstrates supply chain attributes are becoming key factors for assessing cyber risk.

2.3 Data

The dataset used in this chapter comes from two main data sources. The first is data from BitSight (6), and the second is the Veris Community Database (12). The dataset consists of all 4,617 entities in the BitSight database that are classified as software companies and had at least one known digital supply chain relationship with a customer between the time period May 2017 to April 2020. There are 77,923 software companies in the BitSight database that did not have a supply chain relationship with a customer and are excluded from consideration. For each included software company, the dataset contains entity data, supply chain data, and breach incident data.

The entity data consists of internal company information such as employee count and outside-in ratings. The outside-in ratings are generated by BitSight and are meant to reflect internal company security posture based on an analysis of externally observable cybersecurity data. The outside-in ratings contain a component for each internal process such as patching cadence and software updates on a scale from 300 to 820 with higher ratings representing better internal security. The self constructed dataset includes the annual average of each score component for the time period May 2017 to April 2018.

The supply chain data consists of over 12 million digital supply chain relationships detailing the products or services provided between individual companies. These connections are part of the BitSight database. Supply chain features are created by considering the local supply chain network for each software company that consists of all the software company's *customers*, suppliers (denoted *third-party* suppliers), and suppliers of these suppliers (denoted *fourth-party* suppliers). To avoid cycles in the local supply chain network where suppliers are both third-parties and fourth-parties, these companies are only included as third-parties. An example local supply chain network is shown in Figure 2-1 with the software company colored red. Using the respective local supply chain networks, we calculate the number of customers, the number of third-party suppliers, and the number of fourth-party suppliers for each software company. We also define a variable denoted as entity local size which is each software company's total number of customers plus total number of third-party suppliers. In addition, we create sector features based on the local supply chain network by calculating each software company's number of incoming and outgoing edges by sector (sector definitions provided in Appendix A).

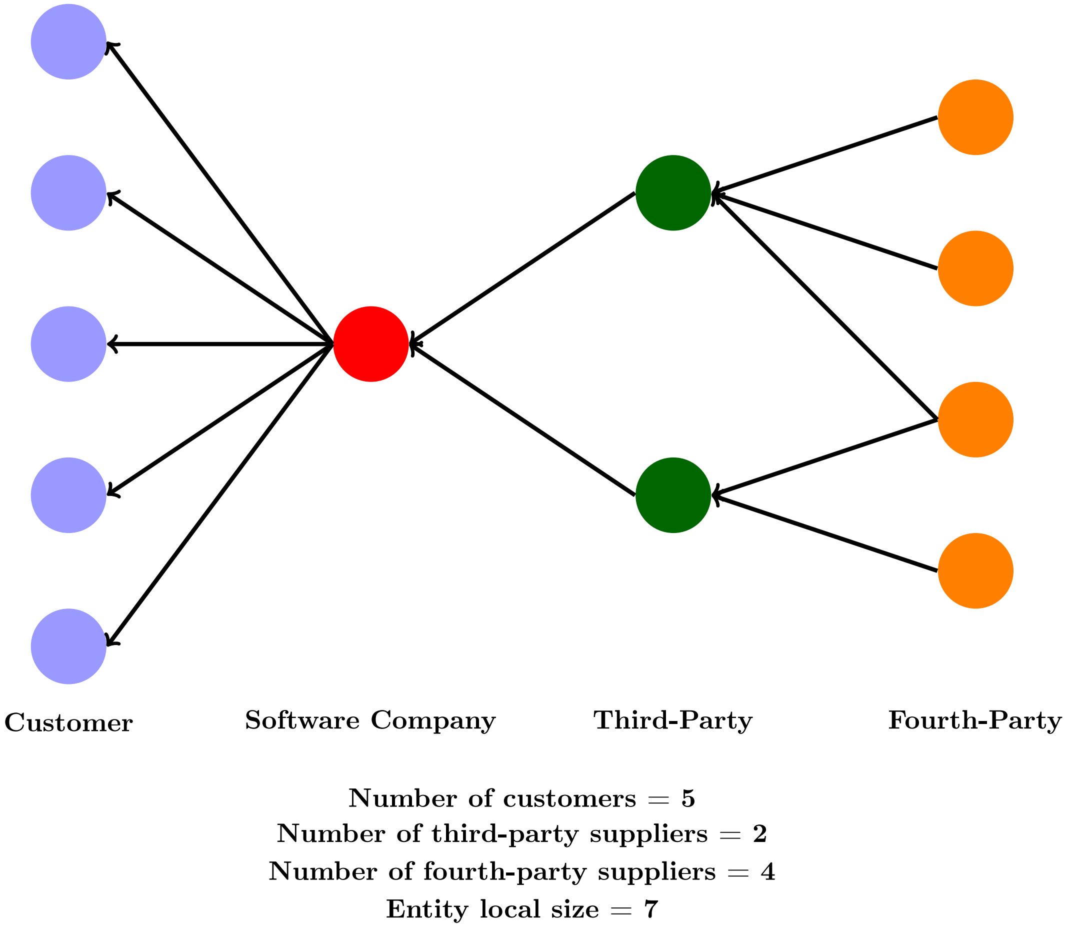


Figure 2-1: Local supply chain network of a software company

Between the entity data and supply chain data there are in total 74 features for each software company, as summarized in Table 2.1 (reference Appendix A for detailed feature definitions).

	Feature	Number (n = 74)
Entity features		
	Employee count	1
	Outside-in ratings	21
Supply chain features		
	Local supply chain features	4
	Customer sector features	24
	Supplier sector features	24

Table 2.1: Summary of features used in clustering algorithms

Finally, the breach incident data reports all documented cyberattacks in the integrated BitSight and Veris databases that occurred between May 2018 to April 2020. During this time period 72 software companies (1.6%) were breached at some point. Although not a part of the clustering algorithm, the breaches provide a benchmark for evaluation. Initial preprocessing revealed that many of the features have heavy-tailed distributions. As a result, we limit extreme values by winsorizing at the 5% and 95% percentiles, and then standardizing the resulting data.

2.4 Methodology

This section employs an unsupervised machine learning clustering to partition the software companies based on the supply chain features described above. Subsequently, the clusters are compared with respect to the proportion of companies in the clusters that were breached during 2018 to 2020. Significantly different proportions would imply that the supply chain features have predictive power with respect to breach risk from software companies. To perform this clustering, we implement a probabilistic approach known as a Gaussian mixture model (GMM) (24). In a GMM, each cluster is modeled as a Gaussian distribution, with its own mean and covariance matrix. The mixture model combines these Gaussian distributions to create a more complex distribution that can better represent the data, and the model parameters are estimated using the expectation maximization algorithm. One of the key advantages of the GMM is its flexibility in modeling complex data distributions, making it particularly useful for problems where the underlying structure of the data may be unknown or difficult to model using other techniques.

We perform GMM clustering on three sets of data features (see Appendix A). The first version, hereafter referred to as the "supply chain clustering", includes 53 features that encompass employee count and local supply chain network features. The second version, hereafter referred to as the "internal posture clustering", includes 21 features that encompass all BitSight outside-in ratings. The third version, hereafter referred to as the "combined clustering", includes 74 features that encompass all features in the supply chain clustering and internal posture clustering.

Since clustering algorithms are known to perform poorly on very high dimensional data, a PCA feature dimensionality reduction is performed on each set of data features, retaining the minimum number of components that explain over 90% of the variance. Using each compressed data as input, we employ the GMM clustering algorithm to produce a set of clusters, where every software company is assigned to a specific cluster. This model only requires a single hyperparameter K indicating the number of desired clusters. To determine the optimal number of clusters, we calculate the Silhouette score for a range of possible cluster sizes K between two and fifteen across twenty random initializations. The Silhouette score is a common metric used to evaluate the performance of clustering algorithms and is defined in Equation (2.1). The Silhouette score can range from -1 to 1, with 1 representing better clustering. As a result, the cluster size K that produced the highest mean Silhouette score across all software companies was chosen as optimal (32).

$$\text{Silhouette Score} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{b(i)-a(i)}{\max(a(i), b(i))}\right) \tag{2.1}$$

where:

n : number of samples

$a(i)$: average distance to all points within the same cluster

$b(i)$: average distance to all points in the nearest neighboring cluster

A common technique to help interpret the results of unsupervised clustering algorithms is to examine the differences between the center locations of each resulting cluster, also known as the cluster centroids. Each centroid represents the average observation within each cluster, and describing the feature differences between these average observations provides insights into the characteristics that separate between clusters.

2.5 Results

The GMM algorithm ultimately selects $K = 2$ clusters as optimal for each of the three sets of data features. The different clustering models are assessed based on the documented data breaches throughout 2018 to 2020, as shown in Table 2.2.

Clustering	Cluster	Companies (№)	Breaches (№)	Breach Proportion Rate (%)
Supply Chain	1	2471	17	0.7
	2	2074	55	2.6
Internal Posture	1	2380	23	1.0
	2	2165	49	2.2
Combined	1	2027	15	0.7
	2	2518	57	2.2

Table 2.2: Gaussian mixture model clustering results for $K = 2$ clusters

For each set of data features, the model finds two clusters of roughly equal sizes, where the first cluster is relatively safe and the second cluster is significantly more risky. The risky cluster of the supply chain version not only has a 278% higher breach proportion rate than the safe cluster but also captures 76% of all breached companies in the dataset between 2018 to 2020. The internal posture and combined versions produce slightly weaker results, with a 131% and 201% higher breach proportion rate that capture 68% and 79% of breached companies, respectively. This provides evidence that supply chain features are the most significant drivers for detecting cyber risk in software companies.

Given that the supply chain version performs best, the results of this model are used to detect the differences between the safe and risky clusters. Figure 2-2 presents the results of the GMM clustering algorithm projected to the first two principal components, where every red "X" denotes a breached software company. As seen in the figure, the model essentially finds a single dense, safe cluster as well as a single sparse, risky cluster. In Figure 2-2 it can also be seen that the infamous SolarWinds attack is part of the risky cluster.

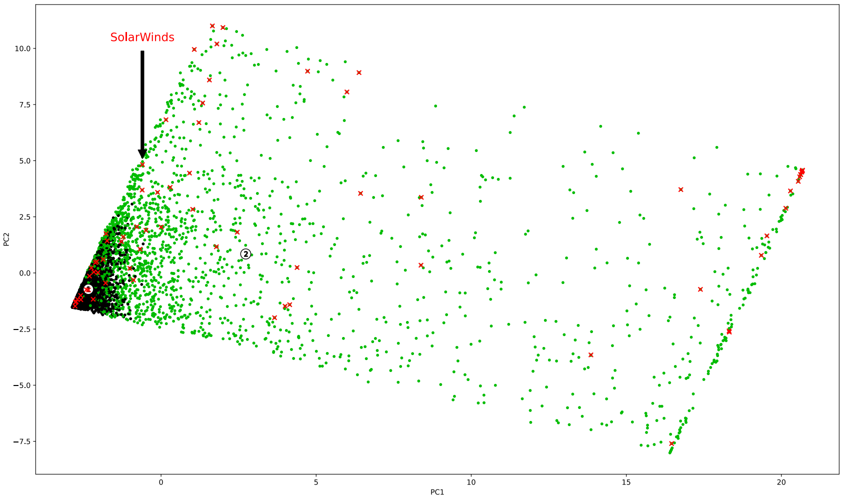


Figure 2-2: Visualization of the supply chain clustering projected to the first two principal components

To gain further insights as to what characteristics the more risky companies have compared to non-risky companies, the two centroids are compared. Table 2.3 shows the top five features with the most significant standardized differences between clusters. Each feature's difference between clusters is highly statistically significant under a standard t-test with p-values of approximately zero. From the table it is noticeable that software companies in the risky cluster tend to have a much higher number of customers and suppliers, particularly in the Technology and Business Services sectors.

Rank	Feature	Cluster 1 Median	Cluster 2 Median
1	№ Third-party suppliers	17	57
2	№ Technology suppliers	15	49
3	№ Business Services suppliers	1	4
4	№ Customers and third-party suppliers	45	476
5	№ Customers	21	418

Table 2.3: Top five features with the most significant standardized differences between clusters. Each feature's median value by cluster is reported.

2.6 Conclusions and Discussion

This chapter builds upon research in the cybersecurity field by leveraging supply chain features to detect high risk software companies. Among several clustering models that were built, the strongest results for this dataset solely rely on supply chain attributes, underscoring their importance in predicting and assessing cyberattack risk. Notably, the SolarWinds company that was involved in a major cyberattack incident belonged to the risky cluster identified through the analysis of supply chain features, which provides an anecdotal evidence that further supports the approach.

Besides outlining a new approach to assess cyber risk of software companies based on digital supply chain attributes, this chapter also highlights several key performance indicators that could be predictive of risk. Furthermore, internal information relevant to the assessment of a company's cyber maturity and risk are hard to obtain at scale, but digital supply chain relationships could be more easily mapped, at least partially. As a result, the framework and derived insights presented in this chapter could inform internal enterprise risk management policies and particularly management and surveillance of digital suppliers. Additionally, it can provide regulators and government organizations risk assessment framework and tools to manage growing national cybersecurity risks.

It is important to note that a potential limitation of this chapter is the exclusion of software companies that lacked supply chain data. This exclusion could limit the generalizability of the results, especially for companies that may not disclose supply chain information. Future research could explore alternative methods for estimating supply chain relationships in order to increase the coverage and accuracy of the analysis.

Chapter 3

A Reinforcement Learning Supply Chain Cyber Risk Model

3.1 Introduction

With the number and severity of cyberattacks rapidly increasing, cybersecurity is a growing priority for companies and organizations around the world. According to the FBI's Internet Crime Complaint Center, the number of cyberattack complaints and the cost of cyberattack losses have significantly increased over the past five years, reaching over $10.3 Billion in total global economic losses last year alone (9). These concerning trends underscore the need to develop reliable cyber risk assessment frameworks and methodologies.

To address this challenge, industry stakeholders have been collecting vast amounts of data on security incidents, vulnerabilities, and other risk factors. One approach that has been promoted by various companies such as BitSight attempts to assess the cyber maturity and risk of individual organizations by developing outside-in ratings (6). These ratings are based on data and information that can be collected on the organization from the public domain and are meant to provide insights on specific internal security postures such as patching cadence and software updates. More recently, Hu et al. (19) have shown that incorporating detailed information on the enterprise's digital supply chain significantly increases out-of-sample predictive power beyond a model that relies on outside-in ratings alone. However, one potential disadvantage of these approaches is that they rely on feature engineering that is somewhat subjective and have no explicit assumptions with respect to attacker behavior. Additionally, training these models requires massive data on historical cyberattacks and related data breaches, which is often "censored" by limited and partial public reporting by companies. In fact, it is fair to assume that only a portion of the actual cyberattacks are ultimately reported (15, 20).

To overcome these weaknesses, this chapter presents an innovative approach for developing a cyber risk assessment methodology that is based on the following elements. The input to the model is a global supply chain network graph, denoted by $G = (V, E)$. Each node $v \in V$ in the graph represents a company and each directed edge $e = (u, v) \in E$ signifies a known digital supplier-customer relationship, where the company corresponding to node u provides a digital service to the company corresponding to node v. Additionally, each node in the graph is associated with features that indicate the size of the corresponding company, and a probability that decreases proportionally with the size, reflecting the likelihood of a successful direct attack on the company. Every directed edge (u, v) in the graph has a product type feature corresponding to the specific digital service provided between the two nodes, and an additional attack probability that exploits the product type offered by node u to node v, assuming that node u was already compromised. Given this supply chain graph, we introduce a theoretical attacker that gains reward proportional to the size of any newly breached company and aims to collect as much reward as possible over a finite time horizon. However, the theoretical attacker does not know the full supply chain graph and is assumed to only see local information such as the set of currently accessible nodes from the current node. By repeatedly exploring accessible edges and attempting to move along them to additional nodes in the graph, the theoretical attacker discovers increasingly rewarding paths throughout the supply chain graph. Once the attacker is fully trained, cyber risk scores are generated by examining which companies the trained attacker chooses to target; for example frequently attacked companies are likely high-value targets.

In this chapter, we empirically implement and validate this approach by leveraging a self-constructed dataset of over 13,000 companies related to the Retail sector, along with each company's associated supply chain relationships. After training a theoretical attacking agent inside this supply chain network using deep reinforcement learning (RL) algorithms, the fully trained attacker then simulates attacks, yielding a risk score for each individual company in the network. This score corresponds to the relative number of breaches the company experiences in simulation. We can evaluate model performance by comparing the generated scores to real world breach incident data. Notably, there is a highly statistically significant difference in the distribution of risk scores between the subset of companies that experienced a cyberattack and the subset of companies that did not. Additionally, the simulated risk scores also have predictive power. The risk scores achieve nearly the same out-of-sample cyberattack detection power as a model that uses outside-in ratings, and outperform the outside-in model when the risk scores are added as an additional feature. Because the general attacker model is validated by real world breach data and holds predictive power, this model provides a new tool for assessing cyber risk and can contribute to the development of more effective risk assessment strategies to combat the growing threat of cyberattacks.

3.2 Literature Review

3.2.1 Cyber Risk Assessment

Historically, researchers have performed cyber risk assessment by creating a variety of supervised and unsupervised machine learning models. For example, BitSight recently performed a study across thousands of organizations comparing its ratings data between the organizations that experienced a cybersecurity incident and those that did not, and found statistically significant correlations between specific BitSight ratings and the likelihood of a cybersecurity incident (5). The research performed by Hu et al. (19) similarly predicts enterprise cyber risk by training on a labeled dataset of breached versus not breached companies. Other models such as the previous chapter use unsupervised clustering methods to detect anomalies and high risk companies.

To the best of our knowledge, there is no work that uses RL approaches to develop predictive risk assessment models with respect to cyberattack breaches. A recent review paper performed a broad survey of RL approaches developed for cybersecurity and found that current applications are generally limited to defending against cyberattacks (28). For example, in one use case, an RL methodology was used to develop an approach to protect against data infusion attacks in autonomous vehicles (31). The model learns to make decisions based on the current state of the vehicle to ensure safety in dynamically changing environments. There are also several applications of successful RL-based autonomous detection systems to monitor network traffic and system activity for suspicious behavior or policy violations (48). Finally, one interesting application of RL is the development of autonomous solutions for network penetration testing (36). Penetration testing a common task in cybersecurity that involves simulating a controlled attack on a computer system to find exploitable vulnerabilities. The paper frames penetration testing as a Markov decision process where network topologies represent states, available exploits represent actions, and breached machines give rewards. The researcher implements a variety of RL algorithms that find optimal attack paths in a small network. Unlike the autonomous penetration testing paper which focuses on generating attack paths in a single computer network, we focus on predicting cyber risk across thousands of companies in a supply chain. Nevertheless, we adopt themes from the penetration testing paper such as framing the general attacker model as a Markov decision process with potential states, actions, rewards, and transitions.

3.2.2 Reinforcement Learning

This section provides a background on the existing RL literature. RL is a type of machine learning that optimizes decision making in an uncertain environment by rewarding good actions and penalizing costly ones (41). The problem is interesting because in most realistic environments, a single action not only affects the immediate reward but also the next state of the environment and therefore all future rewards. As a result, the model must learn through trial-and-error what sequence of actions produces the most long-term reward in a constantly changing environment. In recent years there have been many successful applications of RL ranging from Facebook's push notification algorithm (14), to autonomous navigation of high-altitude balloons (4), to models that demonstrate superhuman proficiency in complicated games such as Chess and AlphaGo (38).

RL problems are typically modeled in the form of a Markov decision process, which is a mathematical framework used for decision-making problems in which outcomes are partly random and partly under the control of a decision-maker. This model is defined by a set of states S, a set of actions A, a reward function $R(s, a, s')$ that specifies the immediate reward for taking an action a in state s and transitioning to state s', and a transition function $P(s' \mid s, a)$ that describes the probability of moving from one state to another after taking an action. Under the Markov decision process tuple (S, A, R, P), an *agent* learns a policy $\pi(s)$ that specifies which action to take when in state s. The goal of the agent is to find an optimal policy π^* which maximizes the expected sum of rewards over a possibly infinite time horizon T:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{T} R_t \mid \pi\right] \tag{3.1}$$

Under the RL paradigm, the agent does not have complete knowledge of the Markov decision process, and its goal is to learn the optimal policy and underlying model dynamics through experience with the environment. Specifically, the agent learns through an iterative process of observing the current state, taking an action, and receiving a reward. This crucial agent-interaction loop is shown in Figure 3-1, where at some time step t an agent performs an action a_t that produces an immediate reward signal r_t and an updated environment state s_t.

Once an agent is trained, it is possible to evaluate performance by initializing the agent at some point in the environment and then calculating how much reward it accumulates over time. A well-trained agent should consistently accumulate a large amount of reward.

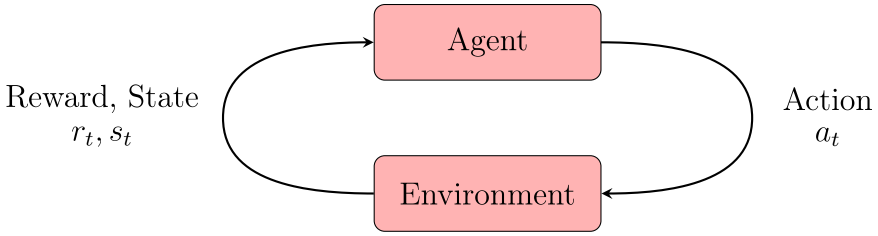


Figure 3-1: Agent-interaction loop

Tabular Methods

An agent must discover through trial-and-error which policy yields the most reward. One of the most foundational algorithms for learning this optimal policy is called Q-Learning (44). This algorithm works by creating a “Q-table” of state-action values denoted as $Q(s, a)$. The Q-table has a row for every state, a column for every action, and each cell contains the estimated value for each state-action combination. At first, the Q-table is randomly initialized, but over repeated interactions with the environment the Q-table converges to the true value of each state-action combination. Once the Q-table has converged, choosing the most rewarding action for any state provides the optimal policy:

$$\pi^* = \max_A Q(s, a) \tag{3.2}$$

Q-Learning is one of the most famous RL algorithms because there are mathematical guarantees the Q-table converges to the true values under certain assumptions such as a steady exploration policy that samples all cells. Although Q-Learning is one of the only algorithms with convergence guarantees, the main downside lies in its requirement to enumerate every single state-action pair in a Q-table. For example, it would be computationally infeasible to use Q-Learning in a complicated environment like chess, where there are approximately 10^{43} possible board configurations, and each configuration has many possible actions (37).

Approximate Value Methods

To address the key limitation of enumerating every state-action combination, approximate value algorithms use a function approximator, such as a neural network, to estimate the Q-values for each action based on the state. To provide a specific example, consider the simple grid environment shown in Figure 3-2. In this environment, the world is represented as a two-dimensional grid of cells where each individual cell has a binary entry (indicated by a red "X"). Since there are 41 cells, and each cell has a binary entry, there are in total 2^{41} possible configurations of states. Furthermore, the agent (indicated by a green stick figure) can perform four possible actions of moving right, left, up, or down.

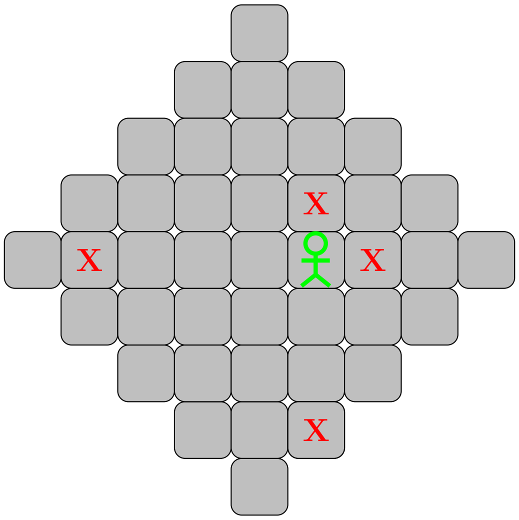


Figure 3-2: Example grid environment

Instead of creating a Q-table with 2^{41} rows and 4 columns, we can introduce a parameter θ for each individual cell in the grid and each individual action. In this new representation of the environment, there are only $41 + 4$ parameters to solve. Once the correct parameters are found the Q-values can be directly recovered from:

$$\pi^* = \max_A Q_\theta(s, a) \tag{3.3}$$

The family of algorithms that iteratively solves the parameterized Q-table is known as approximate value iteration. Unfortunately, vanilla approximate value iteration algorithms are often extremely unstable in practice (41). To improve performance, researchers have developed many subtle algorithmic improvements to tackle various instability issues. For example, the famous Deep Q-Network (DQN) is a popular approximate value iteration algorithm that uses a deep neural network to approximate the Q-values. (26). DQN combines parameterized Q-Learning with experience relay, where the agent stores past experiences in a replay buffer and samples from it to update the Q-network. However, DQN suffers from a phenomenon called overestimation bias, which can result in suboptimal policies. To overcome this limitation, researchers have proposed several variants of DQN, including Double DQN (42), Dueling DQN (43), and Rainbow DQN (17). These variants address the overestimation issue in different ways, such as modifying the Q-Learning update rule to avoid overestimation (Double DQN), using a separate neural network to estimate the value of each action (Dueling DQN), or combining multiple of the proposed variants into a single algorithm (Rainbow DQN). Although DQN and variants do not have theoretical guarantees and can be unstable during training, they have demonstrated impressive empirical performance across a wide range of RL tasks (22).

Policy Space Methods

The previously described algorithms find the optimal policy by randomly initializing a value function, iteratively improving the parameters of the value function until convergence, and then returning the optimal policy by taking the argmax of the value function. Instead of learning a value function and then deriving a policy from it, policy space methods focus on directly learning a single optimal policy. To do so, these algorithms create a parameterized policy function $\pi(a|s;\theta)$ that maps each state to an action, and then updates the parameters θ using gradient ascent on the expected sum of rewards $\mathbb{E}\left[R_t \mid \theta\right]$:

$$\theta_{t+1} = \theta_t + \alpha_t \nabla \mathbb{E}\left[R_t \mid \theta_t\right] \tag{3.4}$$

Figure 3-3 visually demonstrates the difference between approximate value methods and policy space methods. The algorithm on the left side of the figure iteratively learns a value function $Q_\theta(s, a)$ while the algorithm on the right side of the figure iteratively learns a policy $\pi(a|s;\theta)$.

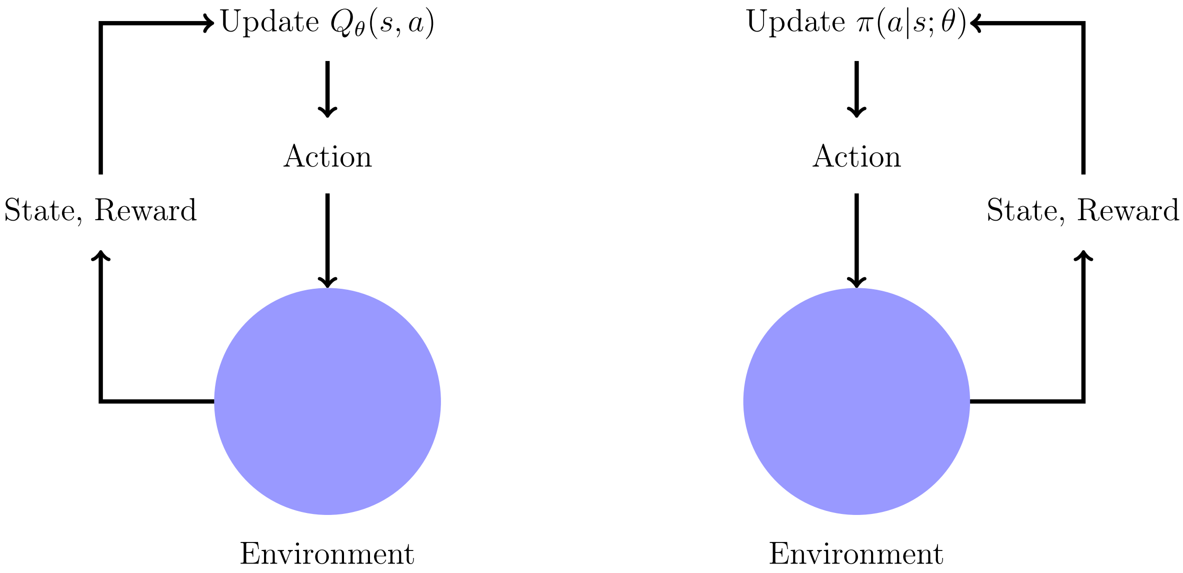


Figure 3-3: Value-based algorithm (left) and policy-based algorithm (right)

Once the parameters of a policy space method converge, the optimal policy directly follows. This algorithm is the most foundational policy space method and is known as REINFORCE (46). A REINFORCE algorithm is easy to compute and simpler to understand, but the gradient ascent process often creates a very large variance that requires many samples to converge (47). To solve this problem effectively, researchers have combined value-based methods and policy-based methods into a single algorithm known as the "actor-critic" framework. In actor-critic algorithms, the critic function approximator estimates the parameterized value function and the actor function approximator solves the policy gradient ascent problem in the direction suggested by the critic. This algorithm is shown in Figure 3-4. Overall, actor-critic algorithms offer a balance between the advantages of value-based and policy-based methods, and in practice often offer better performance than either method alone. Additionally, there are many variants such as advantage actor-critic (A2C) and asynchronous actor-critic (A3C) that improve upon the basic actor-critic framework by using a parallelized architecture (25).

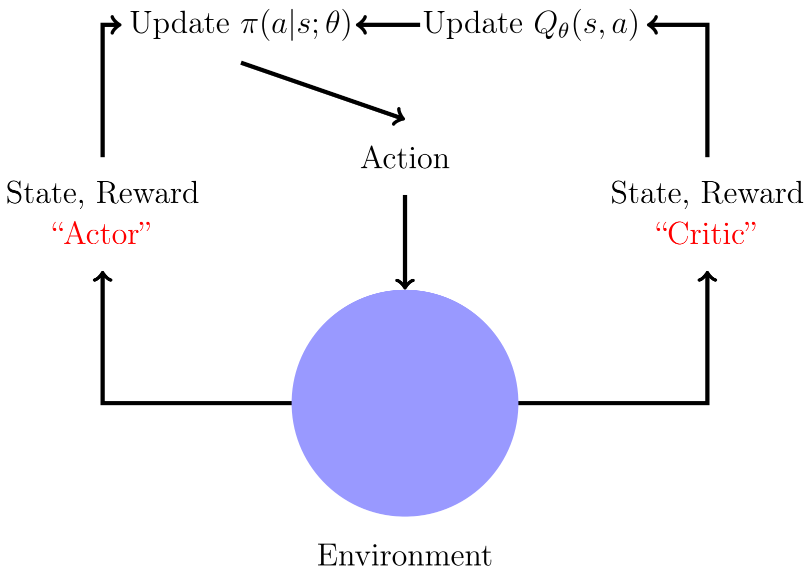


Figure 3-4: Actor-critic algorithm

State-of-the-art RL algorithms improve on the actor-critic framework by optimizing a seemingly small hyperparameter. Recall policy space parameters update through the gradient ascent Equation (3.4). At each iteration, the policy is updated by a learning rate α_t. If the learning rate is too small the algorithm converges slowly, and if the learning rate is too large the algorithm converges prematurely. Finding the correct learning rate schedule is a difficult problem, but many recent algorithms such as trust region policy optimization (TRPO) (34) and proximal policy optimization (PPO) (35) efficiently balance the learning rate to ensure smooth, stable training.

3.3 Methodology

The input data for the general attacker model consists of a global supply chain network graph, denoted by $G = (V, E)$. The graph consists of individual nodes $v \in V$, where each node corresponds to a company. Additionally, each node is associated with a feature that signifies the size of the respective company, and a probability that represents the difficulty of directly attacking the company. This probability is referred to as the *internal penetration probability* and decreases proportionally to the size of the company. Each directed edge $e = (u, v) \in E$ signifies a known digital supplier-customer relationship, where the company corresponding to node u provides a digital service to the company corresponding to node v. Every directed edge has a product type feature that corresponds to the specific digital service provided, as well as a probability indicating the ease of attack propagation between the two nodes, denoted as the *edge propagation probability*. The edge propagation probability is influenced by two factors. First, the directionality of the edge plays a role, with attacks targeting downstream customers having a higher success rate than those targeting upstream suppliers, due to differences in trusted cybersecurity relationship controls. Second, the product type feature for each edge also affects the probability, as riskier types of digital services have less secure systems, resulting in a higher probability of successful attacks. To summarize, the input data for the general attacker model consists of a global supply chain graph, where every node corresponds to an individual company with size and internal penetration probability features, and every edge corresponds to a digital supplier-customer relationship between two nodes with directionality, product type, and edge propagation probability features.

We introduce a theoretical attacking agent that operates within this supply chain graph, and develop a Markov decision process to model its behavior. In the context of a Markov decision process, (S) represents the set of states in the environment, (A) represents the set of actions available to the agent, (R) represents the reward function that measures the desirability of a state-action pair, and (P) represents the transition probabilities that determine the likelihood of moving from one state to another after taking an action.

A state $s \in S$ is defined as the collection of all known information for each node in the network. Specifically, the state includes for each node if the node is compromised, accessible through supply chain connections, or neither compromised nor accessible. Under this definition, the agent does not know the full supply chain graph and is assumed to only see local information for the set of currently compromised and accessible nodes. Importantly, Equation (3.5) indicates the state space grows exponentially according to the number of nodes in the network, suggesting deep RL algorithms are necessary.

$$|S| \in O(4^n) \tag{3.5}$$

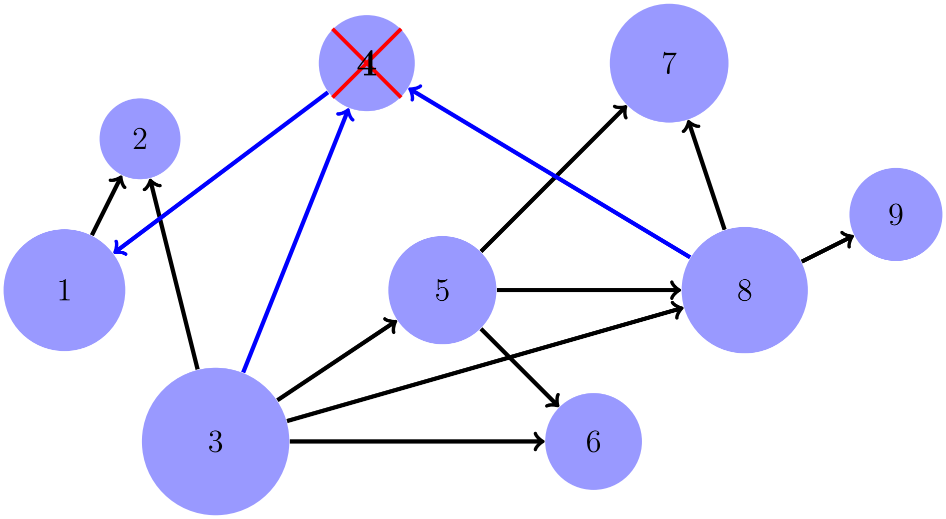
At each time step, an agent can perform an attacking action against any accessible node in the network. Every attack incurs a fixed cost, but each successful attack gains value representing reward for breaching the new company, which is linearly proportional to the size of the newly breached company. Therefore, the reward for any attacking action is determined by the value of the resulting state transition minus the cost of an action.

$$R(s', a, s) = \text{value}(s', s) - \text{cost}(a) \tag{3.6}$$

Finally, the stochastic transition function reflects supply chain dynamics and determines how the environment evolves over time. Specifically, if the agent is currently at node u and attacks node v, the internal penetration probability associated with node v and the edge propagation probability associated with edge (u, v) combine to determine the likelihood of a successful attack and subsequent state transition. If the attack is successful, the attacked node becomes compromised, and the agent gains access to all nodes accessible from the newly compromised node. Conversely, if the attack is unsuccessful, the environment remains in the same state at the next time step. Given enough time, an attacker will compromise every company in the network, so we restrict the agent to maximize reward over a finite time horizon T.

$$P(s' \mid s, a) = \text{internal penetration}(v) \cdot \text{edge propagation}(u, v) \tag{3.7}$$

Under this Markov decision process, the theoretical attacking agent aims to learn a policy that maximizes the expected cumulative reward over a finite time horizon T. To achieve this, the agent employs a deep RL algorithm that repeatedly interacts with the environment by selecting actions, observing state transitions, and receiving rewards. The learning process consists of multiple episodes, where each episode begins with a random company being chosen as compromised, its immediate supply chain connections labeled as accessible, and every other node labeled as neither compromised nor accessible. Figure 3-5 demonstrates the start of an episode in a small supply chain network. Notice the agent begins at node 4 which is compromised, the immediate supply chain connections (nodes 1, 3, and 8) are labeled as accessible in blue, and every other node is labeled as neither compromised nor accessible. After the episode initialization, the agent takes actions according to its current policy until the time horizon T is reached or all nodes in the network are compromised. At each time step, the agent updates its estimate of the policy using the observed reward and state transition. During the training process, the attacker learns to explore the supply chain graph, exploit the learned policy to make increasingly rewarding decisions, and adapt its strategies based on the feedback from the rewards received. Overall, the training process of the theoretical attacking agent involves iteratively updating its policy based on observed state transitions and rewards, until it converges to the optimal policy that maximizes the expected cumulative reward.



$$P(4 \rightarrow 1) = \text{internal penetration}(1) \cdot \text{edge propagation}(4{,}1)$$
$$P(4 \rightarrow 3) = \text{internal penetration}(3) \cdot \text{edge propagation}(4{,}3)$$
$$P(4 \rightarrow 8) = \text{internal penetration}(8) \cdot \text{edge propagation}(4{,}8)$$

Figure 3-5: Episode initialization in an example supply chain network

Once the attacker is fully trained, we calculate risk scores by simulating agent actions under a Monte Carlo approach. That is, risk scores are generated by randomly initializing a compromised company in the network and examining which companies are ultimately compromised during an episode. This process is shown in Algorithm 1. After the simulation episodes, each company's final cyber risk score is defined as the proportion of all simulation episodes in which a breach event occurs.

Algorithm 1 Monte Carlo risk simulation

```
for $episode = 1, 2, \ldots, N$ do
    Randomly breach a starting company.
    Simulate agent actions for $T$ time steps.
    return which companies were breached during the episode.
end for
```

3.4 Empirical Validation

We implement the general attacker model using an integrated dataset from two data sources: BitSight (6) and the Veris Community Database (12). The dataset contains all 8,265 companies that are classified as belonging to the Retail sector within the BitSight database between the time period May 2017 to April 2020. In addition, we consider all companies inside the Retail sector's immediate supply chain network. More specifically, for each company in the Retail sector, all of the company's suppliers as well as suppliers of these suppliers are added to the dataset, reference Figure 3-6. After adding these suppliers to the original Retail companies, there are in total 13,832 companies in the final dataset.

For each company in the dataset, we have information including employee count, digital supply chain relationships, and historical breach incidents. The digital supply chain relationships between companies are part of the BitSight database. For each digital supply chain relationship, the BitSight database also contains information on the types of products supplied between supplier and customer. There are 74 total

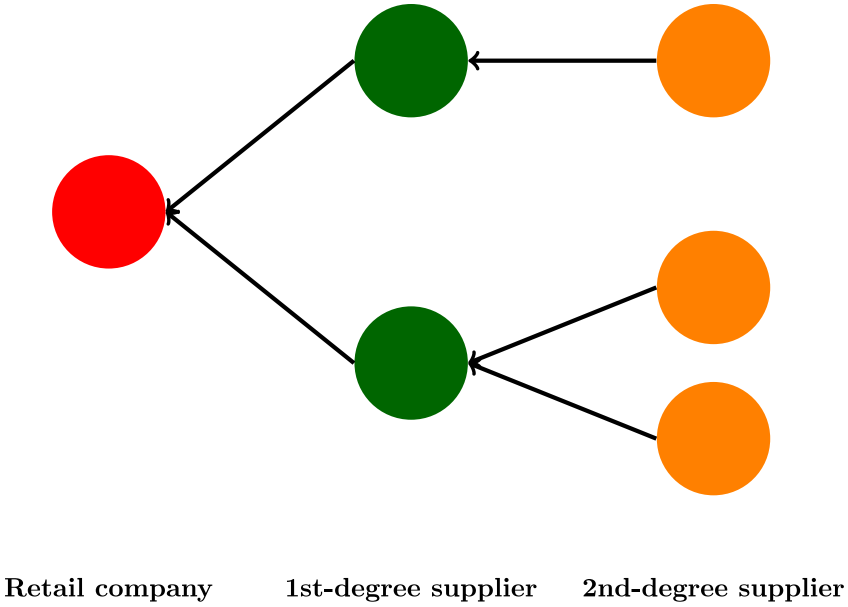


Figure 3-6: Adding a Retail company and its suppliers to the dataset

product types, including products ranging from payment processing, to medical systems, to web application services (see Appendix B for more details). The historical breach incident data reports all documented cyberattacks against the companies in the dataset. In particular, 388 companies (2.81%) were breached at some point over the entire three-year time horizon of interest. Although the historical breach incident data are not features of the general attacker model, the data will be used to evaluate performance. Table 3.1 describes some summary statistics of the dataset.

Feature	**Data (n = 13,832)**
Employee Count, Mean (SD)	7,160 (44,002)
№ Total Connections, Mean (SD)	58 (251)
№ Outgoing Connections, Mean (SD)	30 (237)
№ Incoming Connections, Mean (SD)	30 (59)

Table 3.1: Summary of data used in the attacker model

Using the employee count and digital supply chain information as input, we generate a directed graph $G = (V, E)$. Every node $v \in V$ corresponds to a company with a size feature that is estimated through the company's number of employees. The internal penetration probability feature for each node is modeled with a power law function proportional to the size of a company. That is, a larger company is significantly more difficult to directly attack than a smaller company, as shown in Equation (3.8).

$$\text{Internal Penetration} = \min\left(.25, \text{Size}^{-.25}\right) \tag{3.8}$$

Every edge $e = (u, v) \in E$ corresponds to a digital supply chain connection directed from a supplier node to a customer node. The edge propagation probabilities for each edge are defined in the following manner. First, the directionality feature plays a role, as attacks targeting downstream supply chain customers have a tenfold higher probability of success than attacks targeting upstream suppliers. Second, the product type feature is modeled by binning the BitSight product types into three categories: software edges, risky sector edges, and safe sector edges (reference Appendix B). Software edges compromise all product types related to software including products such as Domain Name System and database hosting. Risky sector edges compromise all product types related to historically dangerous cyber sectors such as Finance and Healthcare. Safe sector edges compromise all remaining product types including Education and Construction. Compared to the baseline safe sector edges, the risky sector edges have a 50% higher edge propagation probability and the software edges have a 100% higher edge propagation probability. In the end, the final edge propagation probability for each edge in the graph is simply the directionality feature multiplied by the product type feature.

It is important to emphasize that the internal penetration and edge propagation probabilities were not manipulated to improve the results of our experiments. Instead, they were predetermined using general domain knowledge and established as part of a transparent and reproducible methodology. The power law function that models the internal penetration probability is based on the observation that larger companies tend to have more robust cybersecurity measures, making them less susceptible to direct attacks. We set a maximum internal penetration probability of *0.25*, which represents the potential vulnerability of a very small company. Similarly, we model the edge propagation probabilities such that the maximum combination of any internal penetration probability and edge propagation probability is *0.5*. Therefore, these probabilities were not designed to skew our results, but rather to enhance the reliability and validity of our findings.

Using the supply chain graph as input, we implement the previously described Markov decision process as an OpenAI gym environment (29). OpenAI gym is a popular RL toolkit with many benchmark environments and algorithms, and the framework allows easy integration with popular open source RL algorithms. In particular, this section integrates the proximal policy optimization (PPO) algorithm from the Stable Baselines3 package (30) due to its stability and high performance during training. Hyperparameter tuning is performed using the Optuna package with a budget of one-hundred trials and a total time steps limit of one million during each individual trial (1). After selecting the hyperparameters with the highest average reward over one-hundred evaluation episodes, a final model is trained and monitored. This model trained until average episode reward converged (23). As seen in Figure 3-7, training stabilizes quickly after approximately 5 million steps, and the termination point is selected as the highest performing point around 30 million steps.

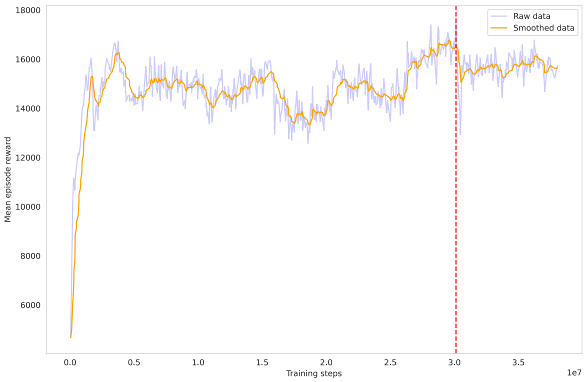


Figure 3-7: Average episode reward across training

Using this termination point, Monte Carlo simulations are run to generate cyber risk scores for each individual company in the dataset. We set the number of Monte Carlo episodes to $N = 100,000$ and create landing probabilities for randomly initializing each simulation. The landing probabilities for each company i are defined in Equation (3.9) and reflect how supply chain cyberattacks typically initialize in larger companies. After all Monte Carlo episodes have run, each company's final risk score is defined as the proportion of simulations in which a breach event occurs.

$$\text{Landing Probability}_i = \frac{\text{Size}_i^{.25}}{\sum\limits_{j=1}^{13{,}832} \text{Size}_j^{.25}} \tag{3.9}$$

3.5 Results

3.5.1 Distribution of Risk Scores

Because the integrated dataset contains historical breach information, we can evaluate the theoretical attacker approach by comparing the distribution of simulated risk scores between the subset of companies that experienced a cyberattack and the subset of companies that did not experience a cyberattack. As a preliminary exploration of the results, Figure 3-8 demonstrates the distributions of risk scores between the two subsets of companies, with the breached subset displayed in red. As seen in the figure, most companies have a very low risk of supply chain cyberattack, while only a few companies have a very high risk of supply chain cyberattack. This result reflects empirical evidence where only a small proportion of companies are attacked every year. Furthermore, the average risk score (depicted as vertical dotted lines) for the subset of breached companies is higher as expected.

Although Figure 3-8 visually shows a difference in risk scores between the two subsets of entities, it would be beneficial to understand if this difference is significant. To do so, we perform two statistical tests. The first test is a standard independent t-test comparing two independent distributions. The test evaluates whether the two means are statistically different from each other when the dependent variable is normally

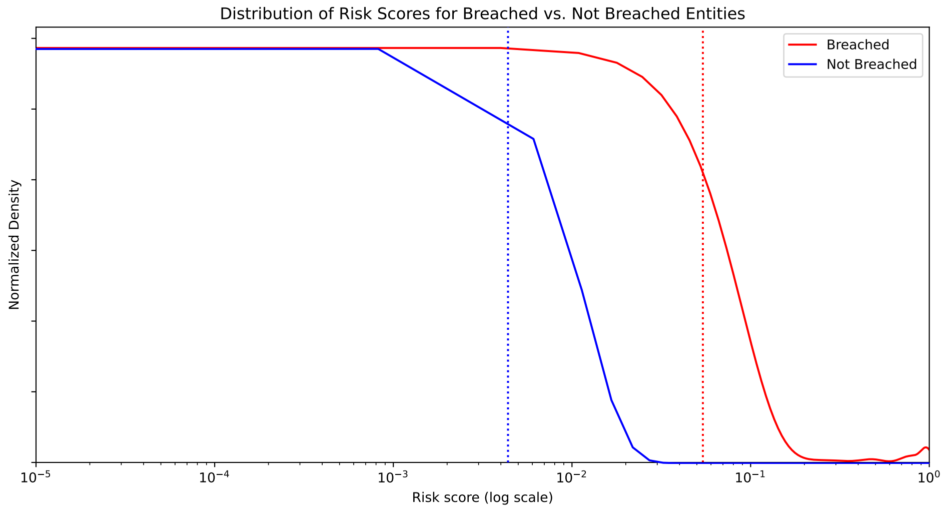


Figure 3-8: Distribution of risk scores between breached and not breached companies

distributed. We also perform the Mann-Whitney U test, which is the nonparametric version of the independent t-test. Unlike the independent t-test, the Mann-Whitney U test does not assume that the data is normally distributed or that the variances of the two distributions are equal. Therefore, it is used when the assumptions of the independent t-test are not met. The results of both tests, which are displayed in Table 3.2, demonstrate there are extremely significant differences (p-values approximately zero) in risk scores between the two subsets of companies regardless of the statistical test.

Statistical Test	**P-Value**
Independent t-test	4.0×10^{-48}
Mann-Whitney U test	7.3×10^{-64}

Table 3.2: Results of statistical tests

As a final comparison of distributions, we can also investigate the cumulative distribution function (CDF) of risk scores between the two subsets of companies. We observed that the CDF of non-breached breached companies dominates at every point, as depicted in Figure 3-9. This strong statistical ordering property is recognized as first-order stochastic dominance (49).

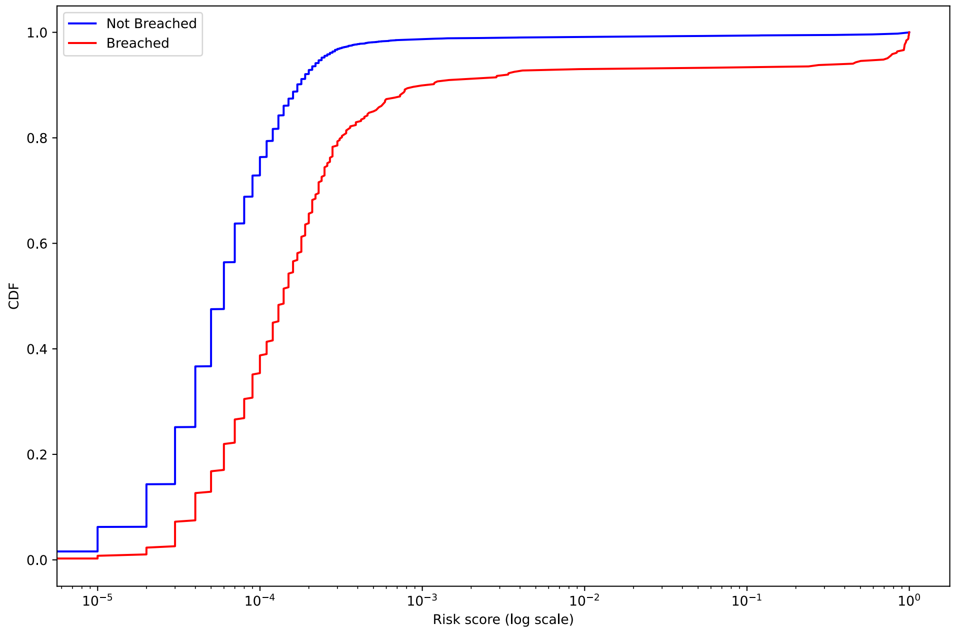


Figure 3-9: First-order stochastic dominance

3.5.2 Cyberattack Detection Power

To determine the predictive power of the risk scores we employ the methodology outlined in Hu et al (19). Three machine learning models are created to predict the likelihood of a cyberattack between the one-year time period May 2019 to April 2020. The first model (Model 1) includes as features for each company basic entity information, specifically sector and number of employees, as well as the newly developed cyber risk scores. The second model (Model 2) includes as features for each company basic entity information as well as BitSight outside-in ratings. The BitSight outside-in ratings are meant to offer data-driven measures of cyber risk for individual companies based on externally observable information, and contain component ratings for specific company internal processes such as patching cadence and software updates on a scale from 300 to 820 with higher ratings representing better internal security. We leverage the outside-in ratings by calculating the average rating of each component over the two-year time horizon May 2017 to April 2019. The third model (Model 3) includes as features for each company basic entity information, BitSight outside-in ratings, and the newly developed cyber risk scores. Appendix C describes all the features in the three models. For each model, we split the dataset into stratified 70% training and 30% testing sets and evaluate the area under the curve (AUC) metric with respect to whether or not each company in the test set experienced a cyber breach event. The algorithm for this binary classification problem is XGBoost (10) and 5-fold cross validation on the training set is performed to find the optimal model hyperparameters. This process is repeated on 1000 random splits of the dataset to ensure stable performance. The results of all three models are shown in Table 3.3.

	AUC
Model 1:	77.9%
Basic entity information and risk scores	77.7% - 78.1%
Model 2:	79.0%
Basic entity information and BitSight ratings	78.8% - 79.2%
Model 3:	79.5%
Basic entity information and BitSight ratings and risk scores	79.4% - 79.7%

Table 3.3: Out-of-sample performance for three cyberattack detection models

As seen in the table, the first model that only includes as features basic entity information and the cyber risk scores performs competitively with the second model, which represents a traditional outside-in ratings model. Furthermore, the third model which includes the cyber risk scores as an additional feature outperforms the traditional outside-in ratings model at the 95% confidence level. To gain further insights into the contribution of each feature in the third model, a Shapley plot is utilized. The Shapley plot quantifies the marginal contribution of the top 10 features towards prediction performance, and reveals that the simulated cyber risk scores that come from the general attacker model are the most important feature, as demonstrated in Figure 3-10.

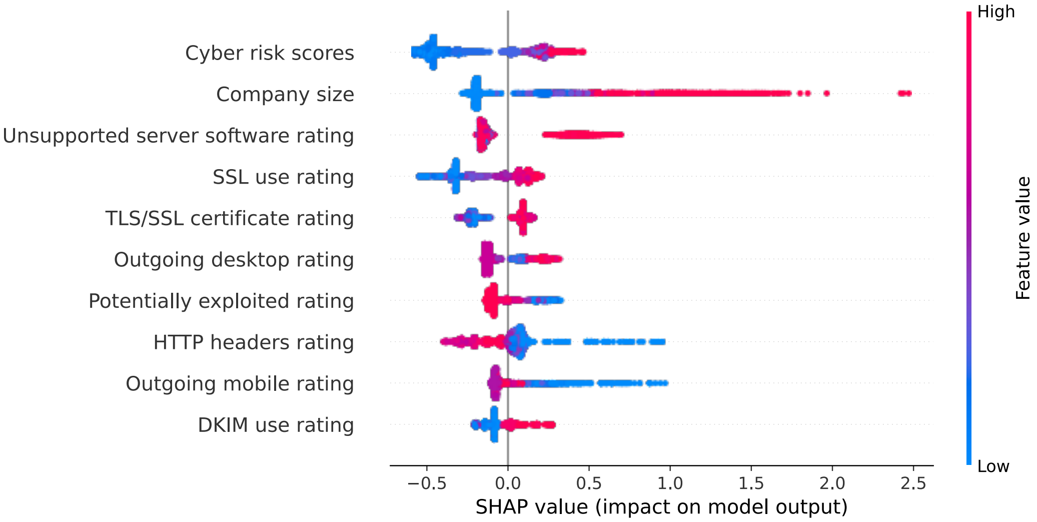


Figure 3-10: Shapley feature importance and impact for the third model

3.6 Conclusions and Discussion

In this chapter, we propose a novel cyber risk assessment approach that is based on a generic attacker model. This type of model is interesting because it represents how an intelligent attacker might target companies in the supply chain. Furthermore, unlike existing supervised learning approaches to cyber risk assessment, the general attacker approach does not require proprietary outside-in ratings or disclosure of censored historical breaches. After empirically implementing this approach on a dataset of over 13,000 companies related to the Retail sector and training the attacker using deep RL algorithms, we compare the distribution of simulated risk scores between the subset of companies that experienced a real-world cyberattack and the subset of companies that did not, and find that the results of the analysis are highly statistically significant. Additionally, the simulated risk scores hold predictive power by not only performing competitively with a traditional outside-in cyber ratings model despite using data that is much more accessible to individual companies but also outperforming the outside-in model when including the risk scores as an additional feature.

Although the model is the first of its kind, there are many potential developments for future work. Theoretically, the model could benefit from algorithmic improvements. For example, the current implementation uses fully connected neural networks following the conventions of Stable Baselines3. However, it is likely more sophisticated approaches such as graph neural networks might improve model performance. Practically speaking, it would also be beneficial to fine tune the internal penetration, edge propagation, and landing probabilities. These probabilities were created using general domain knowledge, but a more refined and data-driven approach would likely lead to even better model performance.

Appendix A

Features used in clustering algorithms

Entity features

- Employee count (1)
- BitSight outside-in ratings (21)
  - Overall entity rating
  - Botnet infections rating
  - Breach history rating
  - TLS/SSL certificate rating
  - DKIM use rating
  - DNSSEC use rating
  - Outgoing communications from mobile devices rating
  - Outgoing communications from desktop devices rating
  - HTTP headers rating
  - Insecure systems rating
  - Malware servers rating
  - Open ports rating
  - Potentially exploited rating
  - Unsupported server software rating
  - Spam rating
  - SPF use rating

  - SSL use rating
  - Unwanted application rating
  - Compromised system rating
  - Patching cadence rating
  - File sharing behavior rating

Supply chain features

- Local supply chain network features (4)
  - Number of customers
  - Number of third-party suppliers
  - Number of fourth-party suppliers
  - Entity local size (number of customers + third-party suppliers)
- Sector features (48)
  - Number Finance customers / suppliers
  - Number Energy customers / suppliers
  - Number Legal customers / suppliers
  - Number Business Services customers / suppliers
  - Number Healthcare customers / suppliers
  - Number Insurance customers / suppliers
  - Number Real Estate customers / suppliers
  - Number Education customers / suppliers
  - Number Technology customers / suppliers
  - Number Tourism customers / suppliers
  - Number Retail customers / suppliers
  - Number Telecommunications customers / suppliers
  - Number Engineering customers / suppliers
  - Number Media customers / suppliers
  - Number Transportation customers / suppliers
  - Number Manufacturing customers / suppliers

- Number Consumer Goods customers / suppliers
- Number Utilities customers / suppliers
- Number Aerospace customers / suppliers
- Number Food Production customers / suppliers
- Number Government customers / suppliers
- Number Credit Union customers / suppliers
- Number Nonprofit customers / suppliers
- Number Unknown customers / suppliers

Appendix B

Product type risk categories

Each digital supply chain connection between a supplier and customer is further classified according to its product type. There are 74 total product types in the BitSight database including the following:

> *"Shipping", "Academic and Education", "Email", "Construction / Industrial", "Quality Management", "Help Desk", "Sustainability / Green Enterprise", "Performance Management", "Commerce", "Order Management", "Hardware", "Audio / Video Delivery", "Inventory Management", "Nonprofit / Fund Management", "Expense Management", "Change Management", "Manufacturing / Engineering", "Procurement Solutions", "Video Platform", "SCM (Supply Chain Management)", "IT Governance", "Payment Processor", "Security Services", "Business Solutions", "Productivity Solutions", "Financial Analytics", "Relationship Management", "Property Management", "HR Management", "Service and Field Support", "Mapping", "BPM (Business Process Management)", "Reporting", "Medical / Healthcare", "Legal and Professional Services", "Ad Network", "Media", "GRC (Governance Risk Compliance)", "Retail", "Analytics", "Telephony", "Enterprise Resource Planning", "Call Center", "Marketing Performance Management", "Business Intelligence", "CMS (Content Management System)", "Virtualization Software", "Remote Server Solutions", "Back-Up and Recovery", "Mobile Technologies", "Network Management", "Virtualization Hosting", "Analytics and Monitoring", "Disaster Recovery", "Software Configuration Management", "Search Engines", "Social Media", "Operating Systems and Languages", "Enterprise Mobility Management", "CDN (Content Delivery Network)", "Application Management", "SIEM (Security Information and Event Management)", "Networking", "Middleware", "IT Operations", "Web Application", "Hosting", "Database", "Enterprise Applications", "Mainframe", "Electronic Data Exchange", "IT Management", "DNS (Domain Name System)", "Server Technologies"*

To create the risk categories that are used in the final model, we bin the product types into three categories based on domain knowledge: safe sector edges, risky sector edges, and software edges. The first 19 product types compromise the safe sector edges, the next 26 product types compromise the risky sector edges, and the final 29 product types compromise the software edges.

Appendix C

Features used in cyberattack detection models

Model 1

1. Employee count
2. Sector
3. Cyber risk scores (general attacker model output)

Model 2

1. Employee count
2. Sector
3. Botnet infections rating
4. TLS/SSL certificate rating
5. DKIM use rating
6. DNSSEC use rating
7. Outgoing communications from mobile devices rating
8. Outgoing communications from desktop devices rating
9. HTTP headers rating
10. Insecure systems rating

11. Malware servers rating
12. Open ports rating
13. Potentially exploited rating
14. Unsupported server software rating
15. Spam rating
16. SPF use rating
17. SSL use rating
18. Unwanted application rating
19. Compromised system rating
20. Patching cadence rating
21. File sharing behavior rating

Model 3

1. Employee count
2. Sector
3. Cyber risk scores (general attacker model output)
4. Botnet infections rating
5. TLS/SSL certificate rating
6. DKIM use rating
7. DNSSEC use rating
8. Outgoing communications from mobile devices rating
9. Outgoing communications from desktop devices rating
10. HTTP headers rating
11. Insecure systems rating
12. Malware servers rating
13. Open ports rating
14. Potentially exploited rating
15. Unsupported server software rating
16. Spam rating

17. SPF use rating
18. SSL use rating
19. Unwanted application rating
20. Compromised system rating
21. Patching cadence rating
22. File sharing behavior rating

Bibliography

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[2] Andy Applebaum, Doug Miller, Blake Strom, Chris Korban, and Ross Wolf. Intelligent, Automated Red Team Emulation. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, ACSAC '16, pages 363–373, Los Angeles, California, 2016. Association for Computing Machinery.

[3] Gabriel Bassett, C. David Hylender, Philippe Langlois, Alex Pinto, and Suzanne Widup. 2022 Data Breach Investigations Report. Technical report, Verizon, 2022.

[4] Marc G. Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C. Machado, Subhodeep Moitra, Sameera S. Ponda, and Ziyu Wang. Autonomous Navigation of Stratospheric Balloons Using Reinforcement Learning. *Nature*, 588:77–82, 2020.

[5] BitSight. The Marsh McLennan Cyber Risk Analytics Center Study Finds Statistically Significant Correlation between Bitsight Analytics and Cybersecurity Incidents. https://www.bitsight.com/resources/the-marsh-mclennan-cyber-risk-analytics-center-study-finds-statistically-significant-correlation-between-bitsight-analytics-and-cybersecurity-incidents, 2022. Accessed: 2023-04-17.

[6] BitSight. BitSight: The Standard in Security Ratings. https://www.bitsight.com/, 2023. Accessed: 2023-04-17.

[7] BitSight. What Are Security Ratings? A Complete and Authoritative Guide. https://www.bitsight.com/blog/what-is-a-security-rating, 2023. Accessed: 2023-04-17.

[8] Sandor Boyson, Thomas M. Corsi, and John-Patrick Paraskevas. Defending Digital Supply Chains: Evidence from a Decade-Long Research Program. *Technovation*, 118:102380, 2022.

[9] Internet Crime Complaint Center. Internet Crime Report 2022. Technical report, Federal Bureau of Investigation, 2021.

[10] Tiangi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, 2016. ACM.

[11] Sung J Choi and M Eric Johnson. The Relationship between Cybersecurity Ratings and the Risk of Hospital Data Breaches. *Journal of the American Medical Informatics Association*, 28(10):2085–2092, 2021.

[12] Veris Community. Veris Community Database. http://veriscommunity.net/index.html, 2023. Accessed: 2023-04-17.

[13] Senator Joni Ernst and Senate RPC Policy Papers. The Solarwinds Cyberattack. https://www.rpc.senate.gov/policy-papers/the-solarwinds-cyberattack, January 2021. Accessed: 2023-04-17.

[14] Jason Gauci, Edoardo Conti, Liang Yitaoi, Kittipat Virochsiri, Yuchen He, Zachary Kaden, Vivek Narayanan, Xiaohui Ye, Zhengxing Chen, and Scott Fujimoto. Horizon: Facebook's Open Source Applied Reinforcement Learning Platform. *arXiv preprint*, 2018.

[15] Info Security Group. IT Pros Believe Cyberattacks Are Under-Reported. https://www.infosecurity-magazine.com/news/it-pros-believe-cyberattacks-are/, 2015. Accessed: 2023-04-17.

[16] Jessica Haworth. Researcher Hacks Apple, Microsoft, and Other Major Tech Companies in Novel Supply Chain Attack. https://portswigger.net/daily-swig/researcher-hacks-apple-microsoft-and-other-major-tech-companies-in-novel-supply-chain-attack, February 2021. Accessed: 2023-04-17.

[17] Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining Improvements in Deep Reinforcement Learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applica-*

tions of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'18/IAAI'18/EAAI'18, pages 3215–3222, New Orleans, Louisiana, USA, 2018. AAAI Press.

[18] White House. Press Briefing by Press Secretary Jen Psaki and Deputy National Security Advisor for Cyber and Emerging Technology Anne Neuberger, February 17, 2021. Press Briefing, February 2021.

[19] Kevin Hu, Retsef Levi, Raphael Yahalom, and El Ghali Zerhouni. Supply Chain Characteristics as Predictors of Cyber Risk: A Machine-Learning Assessment. *arXiv preprint*, 2022.

[20] The Wall Street Journal. Why Some of the Worst Cyberattacks in Health Care Go Unreported. https://www.wsj.com/articles/why-some-of-the-worst-cyberattacks-in-health-care-go-unreported-1497814241, 2017. Accessed: 2023-04-17.

[21] Omer F. Keskin, Kevin Matthe Caramancion, Irem Tatar, Owais Raza, and Unal Tatar. Cyber Third-Party Risk Management: A Comparison of Non-Intrusive Risk Scoring Reports. *Electronics*, 10(10):1168, 2021.

[22] Yuxi Li. Deep Reinforcement Learning: An Overview. *arXiv preprint*, 2018.

[23] Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the Arcade Learning Environment: Evaluation Protocols and Open Problems for General Agents. *J. Artif. Int. Res.*, 61(1):523–562, 2018.

[24] Geoffrey J. McLachlan. *Finite Mixture Models*. Wiley, New York, 2000.

[25] Volodymr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1928–1937, New York, NY, 2016. JMLR.org.

[26] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-Level Control through Deep Reinforcement Learning. *Nature*, 518:529–533, 2015.

[27] BBC News. US Companies Hit by Colossal Cyber-Attack. https://www.bbc.com/news/world-us-canada-57703836, July 2021. Accessed: 2023-04-17.

[28] Thanh Thi Nguyen and Vijay Janapa Reddi. Deep Reinforcement Learning for Cyber Security. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–17, 2021.

[29] OpenAI. Gym. https://openai.com/blog/openai-gym-beta/, 2023. Accessed: 2023-04-17.

[30] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[31] I. Rasheed, F. Hu, and L. Zhang. Deep Reinforcement Learning Approach for Autonomous Vehicle Systems for Maintaining Security and Safety Using LSTM-GAN. *Vehicular Communications*, 26:100266, 2020.

[32] Peter J. Rousseeuw. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[33] Zella G. Ruthberg, editor. *Proceedings of the NBS Invitational Workshop Held at Miami Beach, Florida, March 22-24, 1977*, Audit and Evaluation of Computer Security. U.S. Department of Commerce, October 1977.

[34] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust Region Policy Optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 1889–1897, Lille, France, 2015. PMLR.

[35] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint*, 2017.

[36] Jonathon Schwartz and Hanna Kurniawati. Autonomous Penetration Testing Using Reinforcement Learning. *arXiv preprint*, 2019.

[37] Claude E. Shannon. *Programming a Computer for Playing Chess*, pages 2–13. Springer New York, New York, NY, 1988.

[38] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen,

Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the Game of Go without Human Knowledge. *Nature*, 550:354–359, 2017.

[39] Mariusz Stawowski. The Principles of Network Security Design. *ISSA*, pages 29–31, 2007.

[40] Nimrod Stoler. Breaking Down the Codecov Attack: Finding a Malicious Needle in a Code Haystack. https://www.cyberark.com/resources/blog/breaking-down-the-codecov-attack-finding-a-malicious-needle-in-a-code-haystack, April 2021. Accessed: 2023-04-17.

[41] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning an Introduction.* MIT Press, Cambridge, MA, 1998.

[42] Hado van Hasselt, Arthur Guez, and David Silver. Deep Reinforcement Learning with Double Q-learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), 2016.

[43] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling Network Architectures for Deep Reinforcement Learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1995–2003, New York, NY, 2016. JMLR.org.

[44] Christopher Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[45] Marcus Willett. Lessons of the SolarWinds Hack. *Survival*, 63(2):7–26, 2021.

[46] Ronald J. Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8:229–256, 1992.

[47] Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M Bayen, Sham Kakade, Igor Mordatch, and Pieter Abbeel. Variance Reduction for Policy Gradient with Action-Dependent Factorized Baselines. In *International Conference on Learning Representations*, 2018.

[48] X. Xu and T. Xie. A Reinforcement Learning Approach for Host-Based Intrusion Detection Using Sequences of System Calls. *International Conference on Intelligent Computing*, pages 995–1003, 2005.

[49] Shlomo Yitzhaki. Stochastic Dominance, Mean Variance, and Gini's Mean Difference. *The American Economic Review*, 72(1):178–185, 1982.