Quantitative Risk Analysis With Qualitative Statements

by

Karim Elhammady

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2023

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Cybersecurity risk analysis is crucial for organizations to assess, identify, and prioritize possible threats to their systems and assets. Organizations seek to assess the potential costs of risks in order to determine how to invest in mitigating those risks. Risk analysts rely on qualitative methods to analyze risks. However, qualitative approaches do not produce a complete idea of the loss. The current methods lack efficacy in enabling analysts to make informed decisions. It is crucial to support analysts in their decision-making process by offering means to quantify risks. For this reason, recent studies introduced quantitative risk analysis (QRA) methods to assist organizations in determining risk mitigation strategies and resource allocation. Organizations must use QRA methods to identify and prioritize risks rather than relying on qualitative methods. However, risk analysts tend to prefer quantitative methods since they do not require precise probability estimations.

This thesis proposes a spreadsheet-based QRA method based on verbal likelihoods. Our approach relies on tables constructed by experts that map linguistic likelihood to probability ranges. Using linguistic terms to estimate risk's probability of occurrence will help experts apply quantitative estimation. We eliminate the need to assign exact probabilities by providing a tool that accepts natural language words as input. In modern approaches, Monte Carlo simulation is an important step in QRA to estimate the total loss for risks. For each risk's probability, we will estimate a continuous distribution to use in the simulation. Users will define their own linguistic terms to use them in the risk estimation process. The key benefit of our tool lies in its adaptability across various industries, empowering risk analysts to apply it according to their distinct needs. The tool grants analysts the flexibility to define estimation terms, enhancing precision in their analyses. Finally, we conducted experiments with real examples to validate our approach's accuracy, statistical significance and reliability. We compared our results with those obtained from other methods in the literature. Also, we conducted tests to measure our model's performance and robustness. Our study demonstrates the effectiveness of our approach and its potential to apply it in real-world applications.

Acknowledgements

I extend my heartfelt appreciation to all those who contributed to the realization of this thesis. In particular, I wish to convey my deep thanks to my family for their unwavering support. I also want to express my gratitude to Prof. Fischmeister for his invaluable guidance and professional assistance.

Dedication

This is dedicated to the one I love.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

In an era of evolving technology gradually integrating into our lives, the demand for safe and secure systems has become a critical aspect that organizations should consider. Safety-critical systems within sectors like transportation, medical, automotive, infrastructure, etc., have a profound impact on human lives should they malfunction or face compromise. For this reason, it is vital to analyze and understand cybersecurity risks to plan, specify, and implement risk mitigation actions in a system.

Recent studies introduced risk assessment techniques to identify and analyze cybersecurity risks [12,13]. Quantitative and qualitative risk analysis are two ways to classify risk assessment techniques [21]. Qualitative risk analysis is a subjective method to identify and prioritize based on their likelihood and impact. This method heavily relies on expert judgments and qualitative data to categorize risks using ordinal scales. Although qualitative analysis is a simple and relatively fast approach to analyzing cybersecurity risks, it does not provide the precise insights essential for decision-making. The textbook "How To Measure Anything in Cybersecurity Risk" (HTMA) [13] discussed problems with using qualitative risk analysis methods and how it results in an inaccurate and misleading risk analysis.

On the other hand, quantitative risk analysis is an objective method that leverages numerical data to quantify risks based on probability and impact. This method uses quantifiable and statistical data to construct mathematical models for precise risk assessment. Quantitative methods provide quantifiable estimation for well-informed decision-making. In HTMA [13], the authors introduced examples of quantitative methods that provide insights for more robust decision-making.

Although many methods exist to quantify safety risks, several significant organizations rely primarily on qualitative assessment methods despite the problems associated with these methods. Prior work pointed out significant problems with using ordinal scales [6,17]; moreover, qualitative evaluation is always subjective, and it does not give an idea of the actual loss associated with the risks. Given these problems with qualitative methods, it is crucial to adopt quantitative analysis methods to have a solid foundation for decision-making.

One significant advantage of using quantitative methods is that they can express the impact of cybersecurity in monetary terms, which is important in many applications. For example, embedded systems are price sensitive as they are designed for particular roles in various products. The cost of embedded devices is crucial in the total price of products because they are often manufactured in large quantities (e.g., automobiles or communi-

cation devices). Increasing the cost per unit by just a tiny fraction will primarily affect the total manufacturing cost. Consequently, system designers should carefully analyze the cost/security trade-off to guarantee that the product matches the requirements and is cost-effective. One example of a company mass-producing products is Texas Instruments. Designers must carefully consider the cost of the manufactured microcontroller [14]. A 0.01$ change per unit to implement a security measure will enormously affect the overall cost of the tens of billions of chips produced each year. Relying on qualitative methods will prevent the company from knowing the amount of money it will lose if cybersecurity risks happen. Using a quantitative approach will enable the systems designers to make informed decisions on trade-offs between cost and security.

This work contributes the following to the state-of-the-art:

1. A quantification model that uses linguistic likelihoods to estimate loss for individual security threats.
2. A method for merging different results on words of estimative probability and thus support for continuous integration of new findings.
3. A comparative case study for quantitative assessment of risks for current and future work.

The remainder of the paper is structured as follows: Section 1.2 introduces the problem statement. Section 1.4 describes our approach. Section 2.1 explains the implementation details for our approach. Section 2.2 demonstrates a case study to verify our model. Section 3.1 shows tests that we ran to measure the performance and robustness of our model. Section 4.2 shows our approach's related work and possible future work. In section 4.1, we discuss how to design good tables for our model. Finally, Section 4.3 concludes our paper.

1.2 Problem Statement

The following problem statement defines the expected cost of loss for a system given a set of risks, where each risk has an associated impact cost and verbal likelihood:

Problem 1: Given a set of risks $R = \{r_1, r_2, \ldots, r_n\}$, a set of associated costs $C = \{c_1, c_2, \ldots, c_m\}$ with $c = \langle \text{min cost}, \text{max cost} \rangle$, and a set of associated verbal likelihoods $L = \{l_1, l_2, \ldots, l_o\}$ with l as a member of a standardized dictionary (e.g., the Words of Esimative Probability introduced in Section 1.4), compute the expected cost for a loss Υ.

Let R be a set of n risks, where each risk r_i has an associated impact cost c_j and verbal likelihood l_k. The functions $RC : R \to C$ and $RL : R \to L$ define the mapping from risk to cost and risk to likelihood, respectively.

The expected loss $\upsilon : R \times C \times L \to \mathbb{R}$ is a function that computes the anticipated cost for risks happening. The total expected loss $\Upsilon : 2^R \times 2^C \times 2^L \to \mathbb{R}$ is a function computed as

$$\Upsilon = \sum_{r_i \in R} \upsilon(r_i, RC(r_i), RL(r_i))$$

As an example, consider a vehicle with the following three risks $R = \{$Data breach, Airbag failure, Vehicle leak$\}$. The verbal likelihood $L := \{$*certain*, *certain*, *improbable*$\}$ and impact cost C for each risk are $\{\$500\,000, \$2\,000\,000, \$10\,000\}$ respectively. The expected loss υ for each risk would be $\{\$500\,000, \$2\,000\,000, \$0\}$. Therefore, the total cost Υ for the vehicle is $\$2\,500\,000$.

The challenge with the problem statement is to compute υ, specifically converting verbal likelihoods into quantitative statements. The risk r has an associated cost $RC(r)$ and an associated verbal likelihood $RL(r)$. The verbal likelihood is traditionally not quantifiable. Once we quantify the verbal likelihood $Q : L \to \mathbb{R}$ with a co-domain of $0 \leq Q \leq 1$, then solving υ simply becomes $\upsilon(r) = RC(r) \cdot RL(r)$ [1] or in more recent literature a Monte-Carlo simulation with $\upsilon(r) = MC(RC(r), RL(r))$ [5, 22].

1.3 Preliminary Knowledge

1.3.1 Kernel Density Estimation

Kernel Density Estimation (KDE) [28] is a non-parametric method that estimates a Probability Density Function (PDF) for a dataset from an unknown distribution. KDE utilizes a window function (kernel) to make inferences about the underlying population of the data sample.

KDE is widely used in research to provide a method of visualizing data from unknown distributions. The estimation depends on the window function to smoothen the data and estimate the PDF. KDE is optimally used with complex and unstructured data samples. The method highly adapts to the data's characteristics and properties, making it beneficial in data analysis and visualization. This tool [10] provides a good visualization of how KDE works from specific data points.

The following equation shows how KDE is evaluated:

$F(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)$

Where:

- K is the window function that smoothens the data; most likely, it is a symmetric function like Gaussian Kernel.
- n is the number of samples in the dataset
- x_i is a given data point from the dataset.
- h is the smoothing parameter, also called *bandwidth*, which controls the smoothens of the estimated distribution.

We will use KDE in our approach to estimating the underlying distribution for the sample that we collected. Using this method, we can visualize the PDF for the estimated likelihood of risks.

1.3.2 Monte Carlo Simulation

John von Neumann and Stanislaw Ulam introduced Monte Carlo Simulation (MCS) in the 1940s [2]. Since the method has an uncertain aspect, they named it after a famous gambling location in Monaco because of its similarities to roulette games.

MCS [2] is a probabilistic model that depends on uncertainty to predict outcomes for specific events. The model makes its prediction by running a lot of scenarios that consider multiple factors that might affect the outcome. Each scenario may have a different outcome depending on the model's randomness element in this specific scenario. For example, the distance a plane will fly between two countries is constant. A probabilistic model can consider multiple factors that can affect the flight's light, such as air resistance, mechanical issues, etc. The model randomly evaluates the effect of each factor in each scenario to determine the outcome. After running all scenarios, results are visualized and analyzed statistically to determine the analysis output. The model determines the randomness element for all factors based on pre-defined inputs. For example, users can define a probability distribution for each factor in the simulation, so the model can randomly sample from this distribution in each simulation iteration.

Risk analysts commonly use MCS in various fields to analyze hundreds of risk factors that might affect their use case. For example, in analyzing cybersecurity risks, risk analysts use MCS to simulate the risks and predict the outcome in case they occur. The main goal is to predict the outcome of these risks and how to protect against them.

Various tools and programming languages, such as R, Python, Matlab, and Excel, apply MCS. Excel is one of the most used tools because it is simple to use and doesn't require any programming knowledge. Several Excel plug-ins are available to make using MCS easier.

We will use MCS to predict the outcome of a certain number of risks in a particular system. We used Excel's Visual Basic to develop our tool to convert it into an Excel plug-in.

1.3.3 Two-Sample Kolmogorov-Smirnov Test

The Two-Sample Kolmogorov-Smirnov Test (KST) is a statistical method that compares two datasets. In data analysis, it is crucial to determine whether two samples originated from the same distribution. This test provides a robust method to assess the similarity between two datasets without relying on distributional assumptions.

The test starts by defining the null hypothesis that both datasets come from the same distribution with no significant difference. Consequently, the alternate hypothesis will be that both datasets are not from the same underlying distribution. The outcome of the test is a parameter called p, which determines whether we should accept or reject the null hypothesis. After obtaining the p-value, we compare it with a certain significant level α. If the p-value is smaller than α, we should reject the null hypothesis and conclude that both

datasets are significantly different. On the other hand, if it is greater than α, we conclude that there is no significant difference between both datasets.

KST is specifically applicable when standard parametric tests cannot be used because of the absence of the underlying distribution's properties. The test is highly adaptable and can be applied to data of various types and sizes. We will use KST in our method to verify our model's output.

1.3.4 Words Of Estimative Probability

Words Of Estimative Probability (WEP) are words used by experts to show the likelihood of possible future events when performing risk analysis. WEP is an important tool in intelligence analysis and risk assessment. Different governmental agencies, intelligence agencies, and military organizations widely use WEP. Although the early efforts from Sherman Kent to unify the use of WEP [16], entities have not standardized their use of WEP among themselves. Several agencies started standardizing their use of WEP like the National Intelligence Council (NIC) [23], Center for Internet Security (CIS) [11], Government Of Canada [8, 9], Intergovernmental Panel on Climate Change (IPCC) [15], etc.

Despite the efforts to unify the use of WEP, experts can still interpret it differently. For example, a study shows that doctors can analyze likelihood dissimilarly [24]. Scott Barclay et al. [3] conducted an experiment with NATO officers, asking twenty-three officers to assign probabilities for certain estimative words. Although the officers were familiar with Sherman Kent mapping [16], they estimated most of the terms differently, showing that miscommunication in assessing words is inevitable, even among analysts. Individual mistakes in the estimation process make WEP tables suffer from inaccurate mappings, which proves the importance of standardizing the use of WEP within organizations. The Canadian Government outlined the specific terms used by risk analysts to estimate the probability of cybersecurity risks within the electricity sector [8]. Standardizing the use of WEP across an organization will prevent miscommunication among risk analysts, ensuring precise estimation and analysis of system risks. This, in turn, facilitates informed decision-making.

1.4 Our Approach

As illustrated in Section 1.2, the main challenge in our approach is to compute the risk's probability of occurrence using verbal likelihood. The process of estimating the risks' probability is often complex, time-consuming, and has a lot of inaccuracies. Estimating the probability of a specific risk directly is not practical in many cases. As a result, risk analysts heavily rely on qualitative methods utilizing verbal likelihood or ordinal scales. Our solution proposes a method to execute a quantitative risk analysis while still using verbal likelihood in estimating risk. This method will help risk estimators use much easier linguistic words in the estimation process while getting the output of a quantitative risk analysis.

Figure 1.1 shows an overview of the workflow to execute the quantitative analysis. The dotted box in the figure represents the function Q, which maps the verbal likelihood of a particular risk to a probability value that we can use during MCS. The outcome of the quantitative analysis is the expected cost Υ for the system that risk analysts use for further statistical analysis before decision-making.

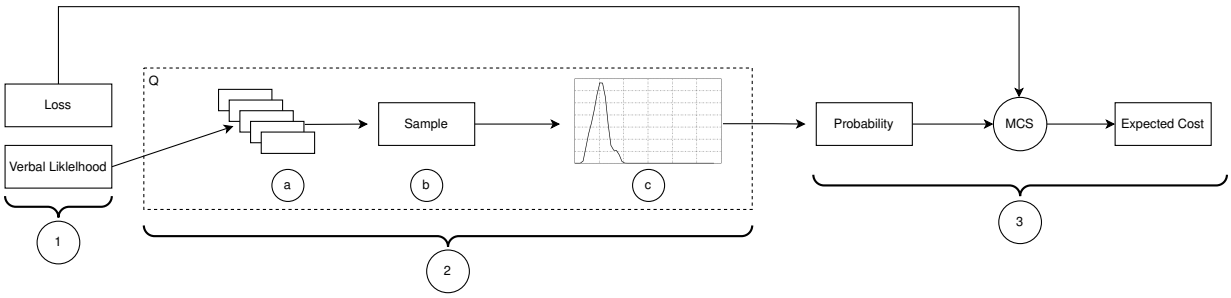


Figure 1.1: Workflow overview

The function Q uses WEP to map verbal likelihood statements to quantitative probabilities.

We use the workflow in figure 1.1 to derive Υ, using Q, for a given set of risks, costs, and verbal likelihood statements. The workflow shows the steps to calculate the expected cost for one risk; it has three main steps. In the first step, we take the user inputs for each risk: verbal likelihood and the loss. The second step is to execute the function Q to map the verbal likelihood to a probability value. The function Q has three main steps:

(a) Each WEP table maps a verbal likelihood to a probability range, so we use the user-inputted likelihood to sample N probability ranges from multiple WEP tables.

(b) We estimate a continuous distribution that describes the density of the sample's values.

(c) We use the estimated distribution to estimate a numeric probability for the risk.

Our approach to making the function Q more robust is to develop sound and complete methods for merging multiple WEP tables into a robust interpretation Q_r. We join tables by sampling N probability ranges randomly to create a sample population from all tables based on a specific likelihood. Taking samples from different sources will make Q_r robust against individual mistakes in WEP mappings.

The output of the function Q_r is the estimated probability for the risk. The third step involves utilizing the output from Q_r along with the loss provided by the user to conduct a MCS and determine the anticipated cost associated with the risk. To calculate the total loss for the system, we execute the workflow for all the user-defined risks. Section 2.1 provides the implementation details for the workflow in Figure 1.1.

Chapter 2

Implementation Details

2.1 Details

This section proposes a spreadsheet-based model for quantitative risk analysis. We explain our model, show the steps to perform the analysis, and illustrate how a user can use the model.

2.1.1 Collecting WEP tables

Our approach merges multiple likelihood mappings to make the quantification function Q robust against mistakes. Experts from renowned organizations constructed each mapping based on the application where the WEP is used. Mappings can depend on the geographical location, application, or even more factors that affect the experts' judgment for constructing the mapping. As mentioned before, we collect our mappings from different resources to include as much diversity as possible in our model. After collecting the tables, we store and use them during the analysis.

Most qualitative risk analysis methods give the user a particular scale to follow during the estimation process. Following a specific scale requires analysts to abide by certain rules, which might cause inaccurate estimations. Risk estimation tools should allow analysts to freely adjust the scales based on their application, which will help in a more accurate estimation process and more concise decision-making.

Our model allows the analysts to freely add or remove WEP tables to the model based on their needs. Analysts can construct their own WEP mapping that matches their application. Users will also have the choice to include other tables from different resources, making our model exceptionally customizable and flexible.

Generally, the more tables we have, the more our quantification function Q will be robust against mistakes.

2.1.2 Taking User Inputs

The first step in figure 1.1 is taking user inputs. Users must identify the risks and their associated impact's cost and verbal likelihood. We represent the impact as a 90% Confidence Interval (CI), a range with 90% certainty that actual loss will fail within this range. Our approach takes the likelihood of risk occurrence in terms of WEP. We assume that experts will fill the likelihood and impact of each risk.

2.1.3 Collecting Samples

The second step in our approach is to execute the function Q_r. Firstly, the function collects samples for each risk from the tables we collected. Algorithm 1 shows the sample collection for each risk.

Algorithm 1: Sample Across WEP Tables

Input: Likelihood of the selected risk (l), number of samples (N)
Output: Random value sampled from WEP ranges

```
1  R ← empty array
2  for i ← 0 to N − 1 do
3  |   t ← randomly select WEP table containing l;
4  |   (x, y) ← probability range from t considering l;
5  |   Append (x, y) to R;
6  |   i ← i + 1;
7  end
8  S ← empty array
9  for j ← 0 to N − 1 do
10 |   (x_j, y_j) ← R_j;
11 |   n ← generate a random number between x_j and y_j;
12 |   Append n to S;
13 end
14 return array S;
```

Each sample contains N probabilities. Our approach uses the sample to estimate a continuous distribution for each risk. The algorithm uses a uniform random number generator in all random picks.

The algorithm is sound, complete, and always terminates, but is not repeatable. Assuming a correct implementation of random selection, the algorithm is sound as it continually selects a value within the likelihood range in the WEP tables. The algorithm is complete as it samples from a range of values. All loops have a strictly monotonically increasing loop counter with a defined bound; thus, every loop and the algorithm will always terminate. Finally, the algorithm is non-repeatable because every execution can select different tables and ranges randomly.

2.1.4 Estimating a Continuous Distribution

KDE estimates a random variable's PDF by deducing a population's characteristics from sample data. The second step in the function Q_r is applying KDE on the sample we collected to estimate a PDF for each risk. After that, we convert the PDF to a Cumulative Density Function (CDF) that we use for sampling during MCS. The final step in the function Q_r is to generate a probability for each risk. We estimate the probability by randomly sampling from the CDF. Subsection 2.1.6 will further clarify the sampling process. Figure 2.2 & Figure 2.3 show our model's CDF and PDF output.

The bandwidth is a crucial aspect of KDE. It determines the spread and the smoothness of the kernel function used to estimate the PDF. The kernel function evaluates the contribution of each data point to the PDF at any given point. By adjusting the bandwidth, a user can change the size of the area around each data point, which affects the PDF estimation. It is vital to choose the bandwidth carefully, as it might significantly impact the results of the quantitative analysis.

Figure 2.1 shows the effect of applying KDE with different bandwidth values. If the bandwidth is too small, the estimated PDF will capture more noise in the data. On the other hand, if the bandwidth is too large, the estimated PDF will be over-smoothed and fail to capture the patterns in the sample. When using high bandwidth (10), the curve is over-smoothed. When using low bandwidth (0.1), the curve is too noisy and captures a lot of irrelevant patterns. We used 1.5 bandwidth to estimate the curve in Figure 2.3, resulting in a balanced PDF capturing some patterns without over-smoothing or adding noise to the distribution.

We use the estimated distribution for sampling during MCS. Sampling from a noisy distribution will result in inaccurate statistical estimates because of the random fluctuations around the actual value. Moreover, sampling from an over-smoothed distribution will also cause unreliable estimates because of the missing details and features of the actual underlying distribution. We conclude that it is essential to carefully choose the bandwidth values to represent the original data accurately.

2.1.5 Representative Values

Representative values, like mean, standard deviation, median, or mode, are used to give different perspectives on a particular distribution. Frequently, no one value corresponds to the *standard* value of a continuous PDF. Instead, the distribution's mean, median, standard deviation and mode are utilized to give distinct viewpoints on the distribution.

Decision makers use the distribution's representative values to further analyze the risks and understand the risks' distributions. Our tool will provide each PDF's mean, mode, median, and standard deviation. Risk analysts can use these values to analyze the probability of occurrence for each risk, which will undoubtedly aid in decision-making.

2.1.6 Monte Carlo Simulation

Following modern approaches, like HTMA [13], we use the output from Q in MCS to identify potential risks and uncertainties. MCS is used to predict the outcome of uncertain events based on past data. The simulation uses the probability distribution as an input, then simulates the possible scenarios and identifies the estimated loss for each iteration.

In the third and last step in the workflow, our approach executes MCS using the open-source Excel add-in, XLRisk [26]. We use the add-in's control ribbon to define the inputs, the number of iterations, and the output for the simulation. In our case, the inputs will be the probability value and expected loss for each risk, and the output will be the total loss for all risks. In each iteration, our model chooses a random probability p_u by sampling from a uniform distribution function, using the $RiskUniform(min, max)$ function in XLRisk where $min = 0$ and $max = 1$. Then, it samples another probability p_c from the risk's CDF, using the $RiskCumul(minvalue, maxvalue, XValues, YValues)$ function in XLRisk. Our tool extracts the $RiskCumul$ function's parameters from the CDF we computed during step two of the analysis. When comparing both probabilities, if $p_c > p_u$, our model will decide that the simulated event will occur and consider it in the total cost.

Our approach calculates the expected cost for the simulated event by sampling from a LogNormal distribution with a mean and variance evaluated using the lower and upper bound entered by the user. HTMA [13] illustrates this approach. We use the $RiskLogNormal$ $(mean, stdev)$ function from XLRisk to calculate the cost. We use the 90% CI that the user provides to calculate the mean and standard deviation and pass them to the function.

XLRisk handles the sampling for all the functions we use in the MCS using the Latin Hypercube sampling method. Finally, after completing the simulation, XLRisk provides comprehensive simulation statistics and graphical output. Figure 2.4 shows a sample of the graphs produced. The curve shows a cumulative distribution of total loss values and their corresponding percentile. For example, suppose the 25^{th} percentile is 378 813, which means that 25% of the total loss values are below this number. In that case, the y-axis value at 25% in figure 2.4 will be 378 813. Risk analysts can use simulation results to analyze the expected loss for the system and decide any mitigation actions required.

2.1.7 Loss Exceedance Curve (LEC)

A Loss Exceedance Curve (LEC) is a graphical representation of the distribution of losses that an organization could incur due to a specific risk event. The LEC shows the probability of losses being greater than or equal to a particular amount on the x-axis across different probability levels on the y-axis in all the loss scenarios.

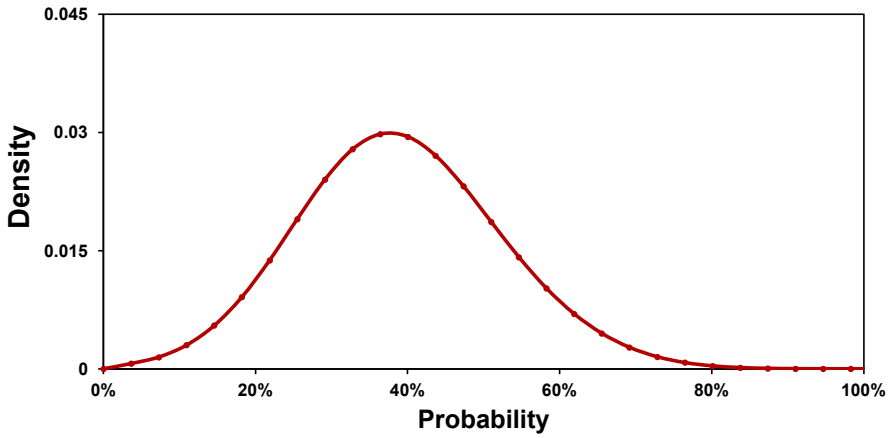
After obtaining the MCS results, we construct the LEC from the loss evaluated in each scenario. Any given point on the curve shows the percentage of all loss values from all scenarios that *exceeds* the corresponding x-axis value. For example, suppose the LEC indicates a 10% probability of a loss exceeding $1 million. In that case, the y-axis value is 10% when the x-axis value is $1 million, representing the percentage of all possible loss values that exceed $1 million.

LEC is a powerful plot for visualizing risk and facilitating risk management decisions, making it a valuable component of risk analysis and management processes.

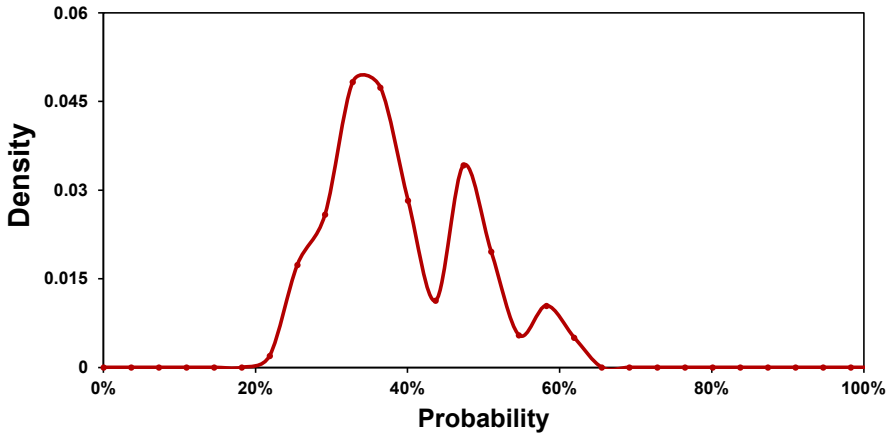
2.1.8 Configuration parameters

The user is able to configure the following parameters for the model:

1. Number of MCS iterations
2. Number of Samples N
3. Bandwidth
4. Steps on the X-axis for the PDF
5. Enabling or Disabling Optimization (discussed in Section 3.2)



(a) PDF with bandwidth = 10



(b) PDF with bandwidth = 0.1

Figure 2.1: PDFs with different bandwidth values

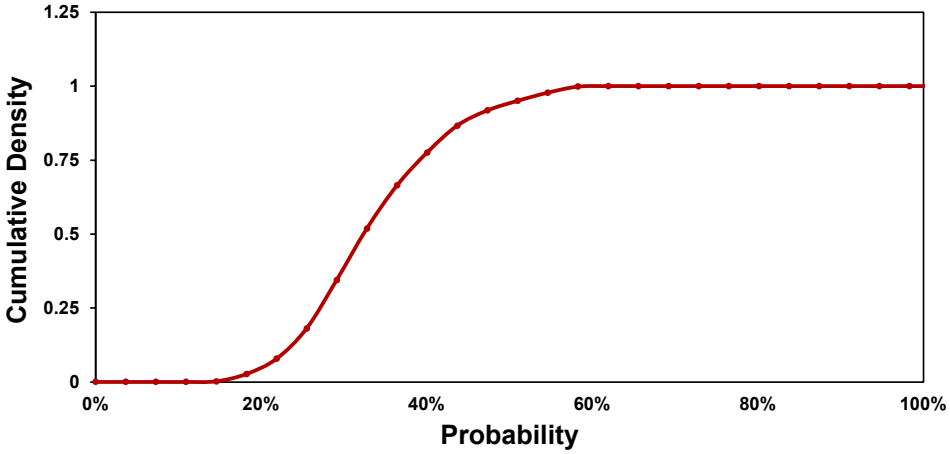


Figure 2.2: Risk's CDF from our model's output

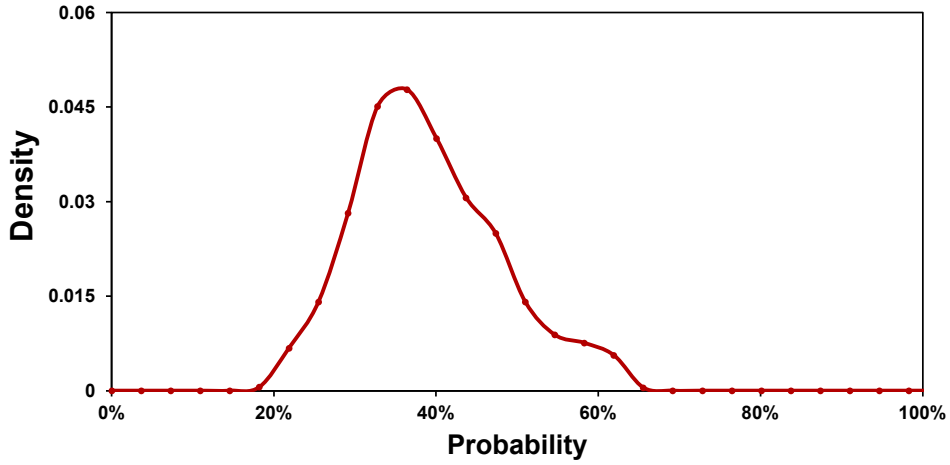


Figure 2.3: Risk's PDF from our model's output

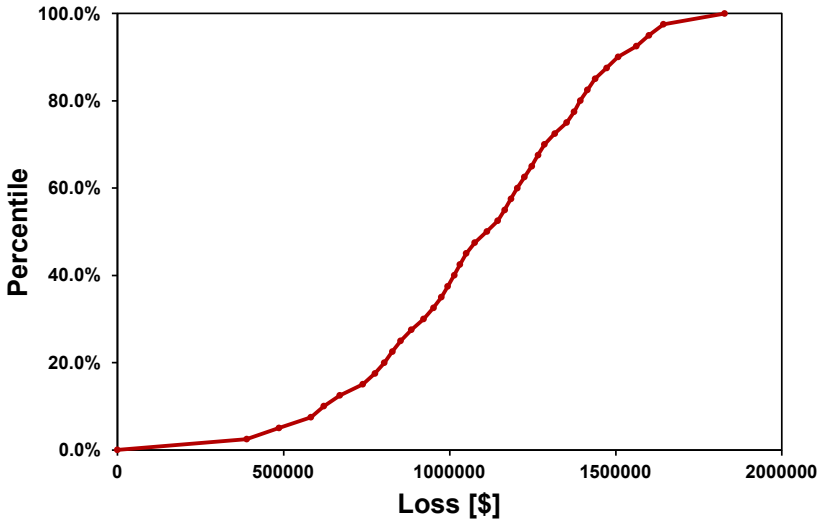


Figure 2.4: XLRisk simulation results - output CDF

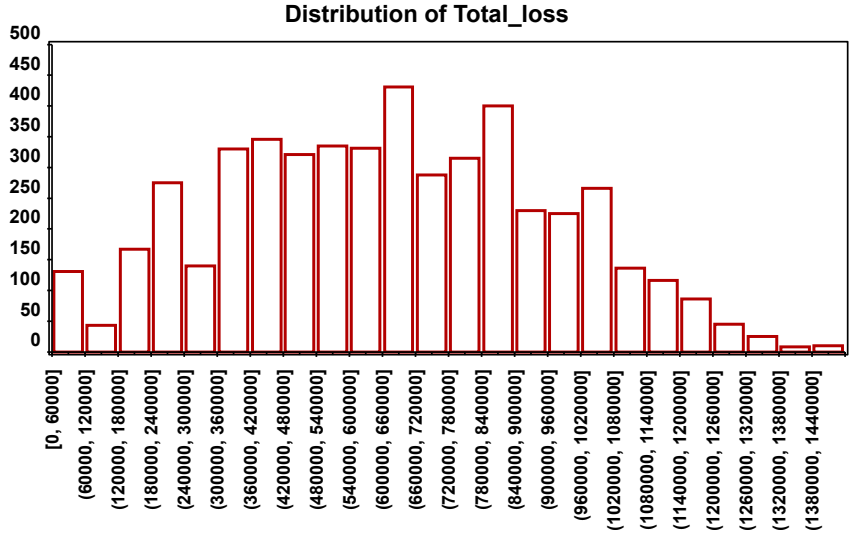


Figure 2.5: XLRisk simulation results - output distribution

2.2 Case Study

This section introduces two case studies with which we compared our model. Firstly, we chose Santini et al. [27] work because it is based on the exact cost calculation method we are using and introduced in HTMA [13]. Secondly, we use the quantitative risk analysis example in HTMA [13] since our cost calculation method is based on this work. In both studies, we adjusted the inputs to fit our model and then compared our model's results with the other work. We focused on comparing the LEC to validate our model because both studies showed it in their results. Both studies aim to evaluate our model's performance, validate its output, and identify possible areas of improvement.

2.2.1 Case study 1

In [27], Santini et al. proposed a quantitative method to estimate the loss for organizations. The authors' approach is based on empirical data collected from various sources. First, their approach uses specific threats and their corresponding likelihood from [19], Table 2.1 shows the threats that the authors considered and their corresponding probability of occurrence. We extracted Table 2.1 from Santini's work.

After knowing the likelihood of the threats, the authors chose three organizations for their case study. They customized the likelihood for each one by collecting answers from stakeholders and analyzing the implemented controls in each organization. Following this, they aggregated the obtained results with empirical data measuring control effectiveness against threats. This aggregation helped calculate a vector t, subsequently utilized to customize the organization's likelihood assessment. Finally, the authors determined the 90% CI based on the organization's operating sector and size.

To compare Santini's results with ours, we had to make some adjustments so the inputs could fit our model, as we are using linguistic likelihoods. Firstly, we calculated each organization's customized threats' likelihood using the vector t from Table 4 in Santini et al.'s work and the values from table 2.1. We used the equation [27]:

$$p_i = (xp_i)^{2/3} \tag{2.1}$$

to calculate the customized likelihoods, where $x = \max\{1 - t_i, 0.06\}$ and p_i is the i^{th} threat's probability in Table 2.1. Table 2.2 shows the customized likelihood for each organization, namely O1, O2, and O3. Secondly, we mapped the values in Table 2.2 to linguistic likelihood by fitting them into a range. For example, the probability *0.62* falls in

the range *0.55–0.80*, which maps to *likely* likelihood in most of the WEP tables. Table 2.3 shows the mapped linguistic likelihoods.

After obtaining the linguistic likelihoods and the 90% CI for impact, we ran our model to quantify the loss using the following parameters:

1. Number of WEP tables = 6
2. Number of Samples $N = 300$
3. MCS iterations = 5000
4. KDE bandwidth = 1.5
5. Steps on the X-axis = 50

Figure 2.6 shows the LECs for all organizations. We compared our LECs with the ones in Figure 1 in [27]. Upon visual inspection, we realized the following key points between each organization's curves:

1. All curves have the same starting point on the y-axis.
2. All curves have the same trend and shape.
3. Our O3's LEC captured some fluctuation that appeared on the curve for O3 appears in [27]
4. All curves have different endpoints on the x-axis. However, we identified that our curves show the correct end value. The loss for a given organization cannot be greater than the sum of the upper bounds in the 90% CI for each threat. For example, when we calculated all the upper bounds for all threats for O1, the value was 1 898 530. Our LEC shows that the loss did not exceed 1 798 777, which is very close to the maximum value. On the other hand, the loss for O1 in Santini's work exceeded 2 000 000. Therefore, we identify that our results are more accurate and match the 90% CI that we input into the model.

Unfortunately, we could not compare the results with a method other than visual inspection due to the absence of the detailed curve's values in Santini's work. If we had more data, we could have used a statistical method like the Two-Sample Kolmogorov-Smirnov Test to prove there is no significant difference between curves.

Table 2.1: Threats with their correspondent likelihoods used in [27]

ID	Threat	Likelihood
1	Malware	0.98
2	Web-based attacks	0.67
3	Denial of services	0.53
4	Malicious insiders	0.40
5	Phishing and social eng.	0.69
6	Malicious code	0.58
7	Stolen devices	0.43
8	Ransomware	0.27
9	Botnets	0.63

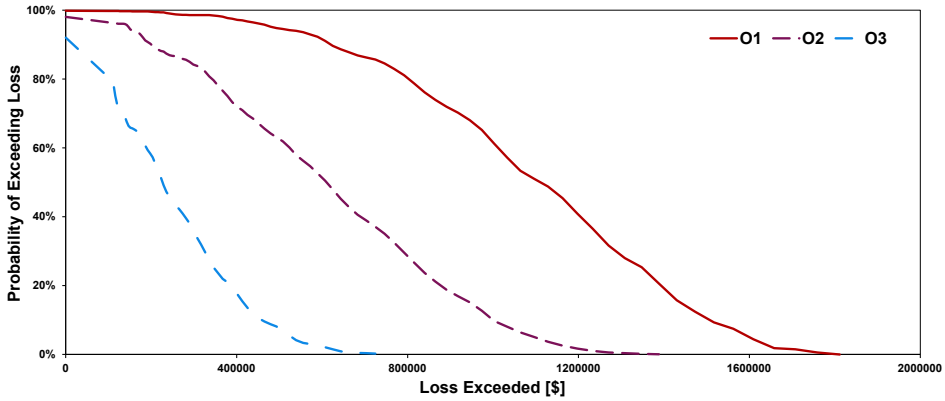


Figure 2.6: LEC output from case study 1

2.2.2 Case Study 2

In HTMA [13], the authors provided an example for their quantitative analysis method. They showed three risks in the analysis, $\{DataBreach, Malware, LostLaptops\}$, each with a probability of loss over 1 year {40%, 15%, 35%} respectively. Moreover, they defined the 90% CI for each risk as lower and upper bounds and used MCS to estimate the total expected loss.

To fit the inputs in our model, we replaced the probability in the example with linguistic likelihoods using the WEP tables we collected. Table 2.4 shows each threat with its corresponding probability and linguistic likelihood. We used the exact 90% CI of impact

Table 2.2: Customized likelihoods for threats for each organization used in the case study [27]

Threat ID	O1	O2	O3
1	0.62	0.53	0.28
2	0.64	0.27	0.28
3	0.33	0.27	0.10
4	0.50	0.41	0.26
5	0.67	0.20	0.17
6	0.62	0.40	0.28
7	0.51	0.48	0.32
8	0.20	0.17	0.08
9	0.15	0.19	0.23

given in the example.

After determining the inputs, we ran our model to quantify the loss using the following configuration:

1. Number of MCS iterations = 5000
2. Number of Samples $N = 300$
3. Bandwidth = 0.1

We ran our model and plotted the LEC to compare it with HTMA's results. Fig. 2.7 shows the LEC from our model and HTMA example. We deduce that the results are almost identical to the original example from HTMA as both have the same starting position (67%) and trend. Some fluctuations in the HTMA curve are not found in the curve from our method, but both curves exactly match in most regions.

Besides the LECs, we compared the mean for total loss in all MCS iterations. As expected, the difference between HTMA's mean and the non-approximated method is negligible; \$1 930 925 and \$1 992 508 respectively. The approximated method's mean is higher (\$2 156 253), which matches the pattern from the LECs. Finally, we used KST to verify the results statistically. We compared the values for the LEC from our model and HTMA to prove that both curves don't have any significant differences. The test yielded a p-value more significant than our predetermined significance level (alpha = 0.05), indicating

Table 2.3: Linguistic likelihoods for threats for each organization used in the case study

Threat ID	**O1**	**O2**	**O3**
1	likely	even chance	unlikely
2	likely	unlikely	unlikely
3	unlikely	unlikely	highly unlikely
4	even chance	even chance	unlikely
5	likely	unlikely	highly unlikely
6	likely	even chance	unlikely
7	even chance	even chance	unlikely
8	highly unlikely	highly unlikely	highly unlikely
9	highly unlikely	unlikely	unlikely

a lack of evidence to support the hypothesis that the two approaches produce statistically different results. We conclude that our results are reliable and similar to the one evaluated in HTMA.

The main difference between our method and HTMA's is the probability p_c. In our method, we sample p_c in each iteration in the MCS from either the CDF in the non-approximated method or the N samples in the approximated method. However, HTMA's method uses the same probability p_c in all iterations. Sampling different probabilities in each iteration introduces diversity in the model and makes it robust against using individual mistakes in estimating probabilities.

Table 2.4: Linguistic likelihoods for threats used in HTMA case study

Threat Name	**Probability**	**Linguistic Likelihood**
Data Breach	40%	even chance
Malware	15%	highly unlikely
Lost Laptops	35%	unlikely

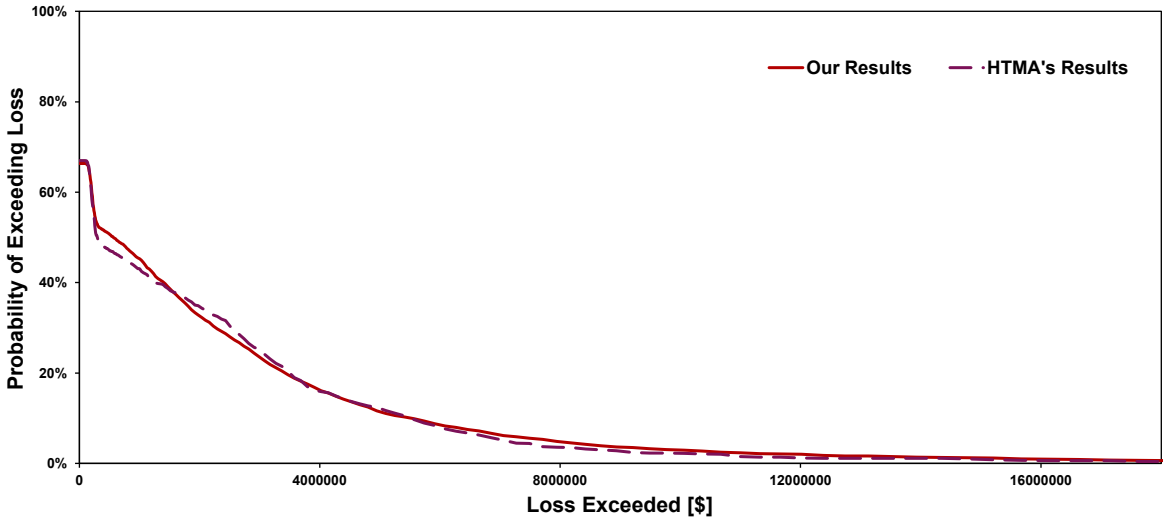


Figure 2.7: HTMA case study LEC results

Chapter 3

Experiments

3.1 Experiments

This section introduces the Performance and Robustness tests we executed on our mode. We demonstrate the tests' results and show our deductions from analyzing the tests' output.

3.1.1 Performance Testing

To evaluate the performance of the proposed model, we conducted a series of experiments that measured the execution time of our model. We recognized two factors that affect the overall execution time of the model: the number of samples N and the number of risks R. To run the experiments, we incrementally increased both factors and measured the execution time of the model at each step, then plotted the results to analyze the system's performance. We describe each experiment separately. We conducted the experiments on a Dell XPS 15 9510 laptop with an Intel Core i7-11800H clocked at 2.3 GHz, 64 GB of DDR4 RAM, and a 1536 GB solid-state drive. The machine runs Windows 11 and NVIDIA GeForce RTX 3050 with 4 GB of GDDR6 memory.

Number of Samples N

We tested the system's performance when incrementing the samples collected from WEP tables, as illustrated in Algorithm 1. First, we define the following constants in the system during all trials:

1. Number of WEP tables = 5
2. Number of Risks = 5

Furthermore, we ran our model using the following values of N: {1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192}. At each value, we noted the execution time of the model, then plotted the number of samples N Vs—execution time. Fig. 3.1 shows the results of the experiment.

As expected, the execution time for the model increases as the number of samples N increases. The relation is almost linear, as Fig. 3.1 shows. During the experiment, we found the number of samples N is bounded by the maximum number of columns in Excel, where N should not exceed the number of available columns.

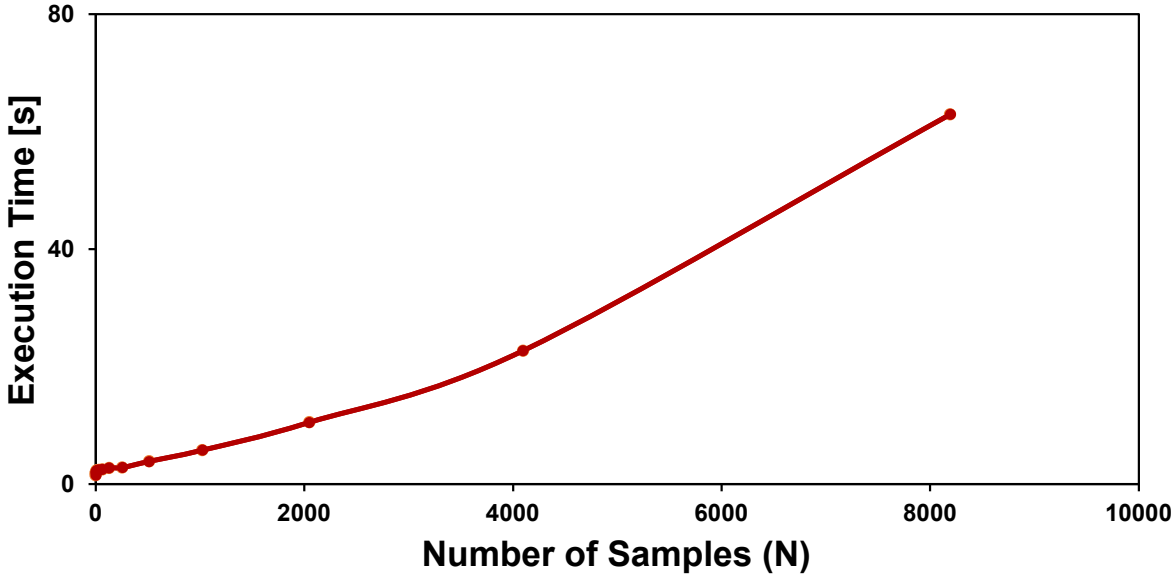


Figure 3.1: number of samples (N) Vs Execution time in seconds

Number of Risks R

We executed the same experiment to evaluate the effect of incrementing the number of risks in the system. First, we define the following constants in the system during all trials:

1. Number of WEP tables = 5
2. Number of Samples = 500

We ran our model using the following values of R: {1,2,4,8,16,32,64,128,256}. At each value, we noted the execution time of the model, then plotted the number of risks R Vs—execution time. Fig. 3.2 shows the results of the experiment.

From the plot, we deduce that the execution time increases as the number of risks R increases. The graph shows increasing the risks from 128 to 256 significantly affected the execution time; the gradient increased.

Finally, we highlight that the number of WEP tables used for sampling does not affect the performance of our model. During sampling, our model samples from a random table from the available tables; moreover, regardless of the number of tables, there is no extra

overhead in choosing a random table. Therefore, we conclude that our model's performance is impacted only by the number of risks R and the number of samples N.

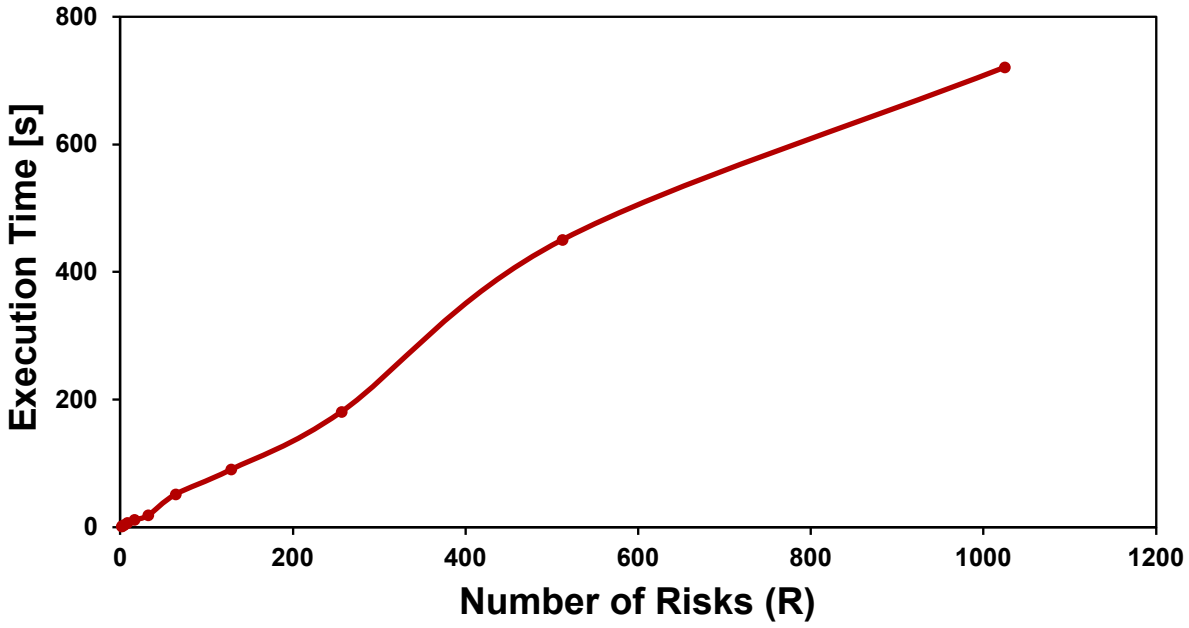


Figure 3.2: Number of risks (R) Vs Execution time in seconds

3.1.2 Robustness Testing

This testing aims to evaluate our model's resilience to mistakes. We consider two mistakes in our tests: inaccurate likelihood estimation for specific risks and bad WEP mappings within a table. The main idea of the tests is to measure how our model will recover from mistakes as we use more WEP tables for sampling; for this reason, we had to generate 2048 WEP tables randomly.

We used a Python script to generate the tables randomly, where all tables have a randomly generated range for each likelihood. The ranges vary significantly for each table to include more diversity in our tables, which corresponds to more use cases. Table 3.2 shows an example of two random WEP mappings. In the *Certainly* likelihood in both WEP mappings, mapping 1 has a range of $0.75 - 1$, while mapping 2 has a range of $0.92 - 1$. Both ranges suggest different applications to which these mappings can be applied; for

example, mapping 1 can be estimated for safety-critical applications because it has low fault tolerance while mapping 2 can be estimated for applications resilient to failures.

Moreover, we run our model with all correct parameters and then use the results as ground truth for the tests. More specifically, we use the loss mean and standard deviation for all simulated scenarios to compare the ground truth and the test values by calculating the absolute difference. We use the following equation to calculate the difference:

$$|mean_{trial} - mean_{truth}| \tag{3.1}$$

The same equation is applied to the standard deviation.

Finally, we discuss our model's robustness against these two mistakes and analyze the results to show our conclusions.

Likelihood Estimation Mistake

In this experiment, we intentionally mapped a specific risk r_A to a wrong estimative word. We aim to measure the effect of the wrong likelihood as we increase the number of tables used in the analysis. Initially, the correct estimation for risk r_A is $EvenChance$; however, during our experiments, we changed it to $HighlyLikely$ in all our trials. We incrementally increased the number of tables in each test; we used the following values for the number of tables: $\{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048\}$. In all trials, we initialized the model with 5 risks, the number of iterations in MCS to 1500 and the number of samples $N = 2500$. Then, we changed the mapping of risk r_A to the wrong likelihood in each trial.

Furthermore, we repeated the same experiment using two different types of WEP tables; Type 1: tables with only two words $(Certainly, NoChance)$ and Type 2: tables with seven words *(Certainly, HighlyLikely, Likely, EvenChance, Unlikely, HighlyUnlikely, NoChance)*. The main idea is to observe how a mistake in estimating r'_As likelihood propagates with different types of tables.

Table 3.1 shows the result of the experiment. The results show the ratio of the deviation to the ground truth value, which we calculate using the following equation:

$$\frac{|mean_{trial} - mean_{truth}|}{mean_{truth}} \tag{3.2}$$

We deduce that the calculated ratio is the same regardless of the number of tables the model used. This means a mistake in estimating likelihoods will undoubtedly affect the analysis results. Upon comparing the results for each type, we find that the ratios for type

1 are much higher than type 2, which suggests that the deviation is more elevated in type 1 because the difference to ground truth ratio is high.

We conclude that our model could be more robust against inaccurate likelihood estimation regardless of the number of tables used. Moreover, we deduce that using WEP tables with more likelihood mappings makes the model less prone to likelihood estimation mistakes.

Table 3.1: Mean results for likelihood estimation mistake test

Number of tables	**Type 1 tables**	**Type 2 tables**
1	0.75	0.19
2	0.59	0.16
4	0.53	0.15
8	0.46	0.21
16	0.30	0.20
32	0.31	0.18
64	0.35	0.19
128	0.38	0.18
256	0.37	0.20
512	0.36	0.20
1024	0.34	0.19
2048	0.35	0.19

Table 3.2: Randomly generated WEP mappings

Likelihood	**Mapping 1**	**Mapping 2**
Certainly	$0.75 - 1$	$0.92 - 1$
Highly Likely	$0.60 - 0.75$	$0.80 - 0.92$
Likely	$0.52 - 0.60$	$0.65 - 0.80$
Even Chance	$0.27 - 0.52$	$0.34 - 0.65$
Unlikely	$0.12 - 0.27$	$0.11 - 0.34$
Highly Unlikely	$0.02 - 0.12$	$0.03 - 0.11$
No Chance	$0 - 0.02$	$0 - 0.03$

WEP Mappings Mistake

In this experiment, we inserted a bad table in the group of WEP tables that we used during sampling. Table 3.3 shows an example of a bad WEP mapping. The problem with this mapping is that *Certainly* likelihood has an extensive range $(0.13 - 1)$ and does not provide a clear indication of the level of confidence; furthermore, the rest of the mappings have a very narrow range of $(0 - 0.13)$, which makes us lose the real benefit behind WEP, and make it hard to differentiate between estimative words. Inaccurate mappings lead to misinterpretations, inaccurate estimations, misunderstandings, and incorrect decision-making. Thus, it is crucial to carefully design WEP mappings to ensure effective use during risk analysis.

Moving on with the experiment, we incrementally increased the tables in our trials; we used the following values for the number of tables: $\{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048\}$. In all trials, we initialized the model with 5 risks, the number of iterations in MCS to 1500 and the number of samples $N = 2500$. Then, we inserted Table 3.3 to the tables used in each trial. Our goal is to measure the effect of the bad table as we increase the number of tables used in the analysis.

Fig. 3.3 shows the mean and standard deviation difference between each trial and the ground truth. The overall trend of the graphs shows that the difference decreases when the number of tables increases. Initially, the difference was more than 100 000 because our model only used the bad table during sampling, which gave inaccurate results. However, the effect of the bad table dramatically decreased when we added just one more table to the model before gradually diminishing until reaching its minimum difference at 2048 tables.

We deduce that our model is robust against inaccurate WEP mappings. As more tables are added to the model, the effect of wrong mappings will decline.

Table 3.3: Bad WEP mappings

Likelihood	**Mapping**
Certainly	$0.13 - 1$
Highly Likely	$0.11 - 0.13$
Likely	$0.09 - 0.11$
Even Chance	$0.07 - 0.09$
Unlikely	$0.05 - 0.07$
Highly Unlikely	$0.02 - 0.05$
No Chance	$0 - 0.02$

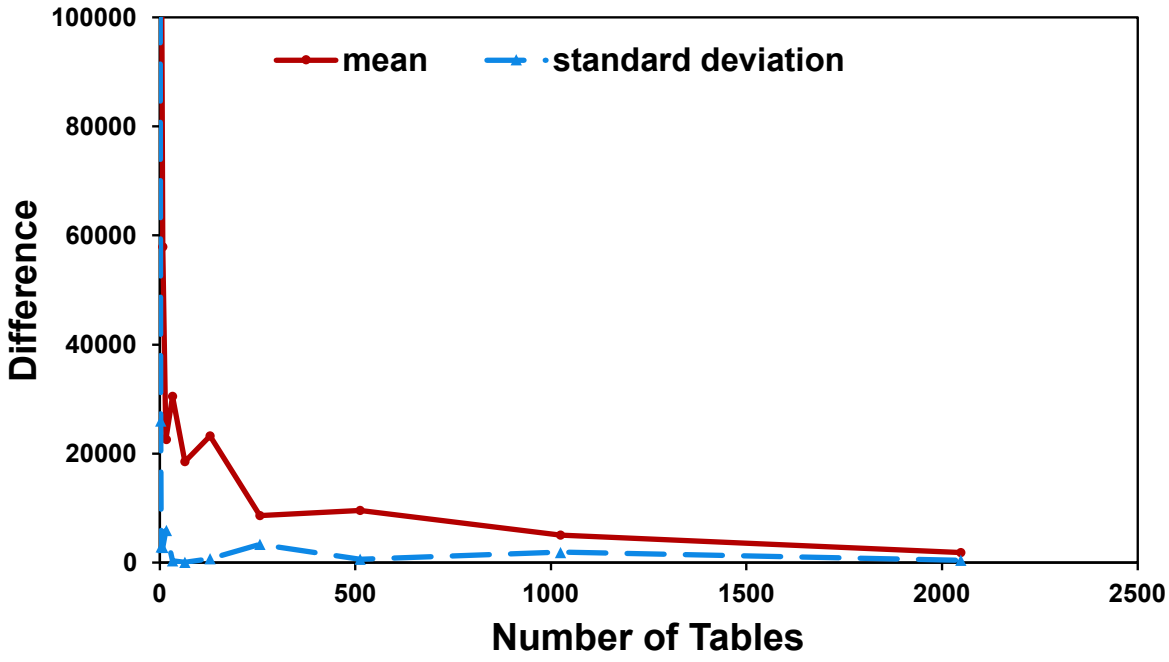


Figure 3.3: number of tables vs. difference in mean and standard deviation between ground truth and trials

3.2 Approximation Technique

This section will introduce an approximation technique to our approach to improve our tool's performance. We approximate the function Q by resampling the N samples we collected from the WEP tables. The primary motivation is to provide a method for an initial assessment with approximated values for the risks without running the entire model and computing all the formulas, which will significantly save the computation time for the model. This technique will help risk analysts get preliminary results in a shorter execution time. In this section, we will explain the method, validate the results and show the speedup in execution time for using this technique.

3.2.1 Technique overview

In Section 2.1, we explained that we sample the probability p_c from the CDF to compare it with the random probability p_u during MCS. In this technique, instead of sampling from the CDF, we will uniformly pick the probability p_c from the N samples we collected from the tables and then compare it with p_u to complete the MCS. Figure 3.4 shows the workflow using the approximated method. The significant difference between Figures 1.1 and 3.4 is part C in step 2; we completely discarded the KDE calculations in the optimized approach. Although the difference between both figures is only one step, it will dramatically speed up the execution time because of the high computation cost of KDE.

Using the approximation technique, we will completely discard all the formulas used during the KDE and generate the PDF, significantly decreasing the computational cost and improving our model's performance. Since it is an approximation technique, we expect to obtain slightly different results and LECs. However, we will prove that both curves have no statistically significant difference. We will also show the performance gain from using this method.

3.2.2 Validating the results

We used the case study from HTMA [13] to validate the results. We will evaluate our results using both the approximated and non-approximated methods and plot them along with the actual results from the case study on the same graph. Figure 3.5 shows the diagram containing three curves: optimized, non-optimized, and results from HTMA. Upon visual inspection, we deduced that the three curves have the same trend and pattern. The curve for HTMA's results has some fluctuations that are not visible in the other two

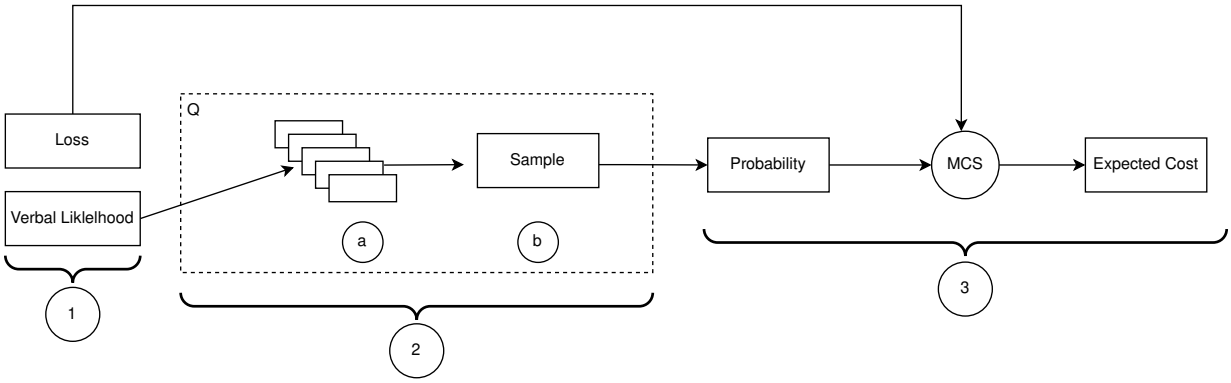


Figure 3.4: Optimized Workflow overview

curves; however, these fluctuations are insignificant, and the curve matches the different two curves in most regions. The curves for the optimized and non-optimized methods are almost identical and follow the same pattern; however, the optimized curve has slightly different values, which we expect because of the approximation technique that we used to evaluate the results.

3.2.3 Evaluating speedup

One of the benefits of having the approximation method is to improve the model's performance. As we illustrated, the approximation technique will discard the KDE in our original method, significantly reducing our model's computational cost. Leveraging the speedup that the approximation technique introduces, users can obtain reliable results for initial analysis without running the entire model in a much faster execution time. We evaluated the speedup by dividing the execution time (e) for both methods using the following equation:

$$e_{approximated}/e_{non-approximated} \tag{3.3}$$

We will showcase the speedup gained from using the approximation method in our model.

In section 3.1, we evaluated our model's performance for the number of samples (N) and the number of risks (R) in the system. We repeated the experiments using the approximation technique to evaluate the speedup. We ran all experiments using the same parameters in section 3.1. Figures 3.7 and 3.6 show the model's execution time using the approximated and non-approximated methods for different values of R and N. We will discuss each one separately.

Figure 3.7 shows the performance gain for using the approximation technique for different values of N. The graph shows that the value of e using the approximation technique is faster than the standard method. Initially, the value of e was close between both methods; however, as we increased the number of samples N, the approximated method's execution time was faster. We calculated the speedup at each value of N and evaluated the average for all values. The average speedup was 1.2.

Moreover, Figure 3.6 shows the performance gain for using the approximation technique for different values of R. The curve indicates that the value of e for the approximated method is faster than the normal one. We calculated the speedup for each value of R and then evaluated the average for all values. The average speedup was 2.2.

3.2.4 Statistical Signficant

We compared the outcomes obtained from the approximated and non-approximated approaches to verify our approach further. To rigorously assess the similarity or dissimilarity of these results, we employed the KST. This statistical test, which evaluates the maximum absolute difference between the cumulative distribution functions of two datasets, provides a robust means of determining whether the results from these approaches significantly differ.

We used the evaluated values to plot the LEC for the HTMA case study in section 2.2. We ran the experiment using both approaches and used the values from the LEC for the test. The test yielded a p-value more significant than our predetermined significance level (alpha = 0.05), indicating a lack of evidence to support the hypothesis that the two approaches produce statistically different results. In other words, our investigation's results suggest no significant difference between the outcomes generated by the two methods. This finding provides valuable insights into the reliability and comparability of the two approaches under consideration, supporting the notion that they produce consistent results within the context of our study.

3.2.5 Limitations

Implementing the approximation technique introduced some limitations on the model because of the Excel limitation. The main difference in the approximation technique is that we are sampling from a set of numbers instead of directly sampling from a CDF. The function $RiskDUniform$ in XLRISK [26], which samples uniformly from a list of numbers, requires to pass the entire array to it. Excel has a memory limitation on the size

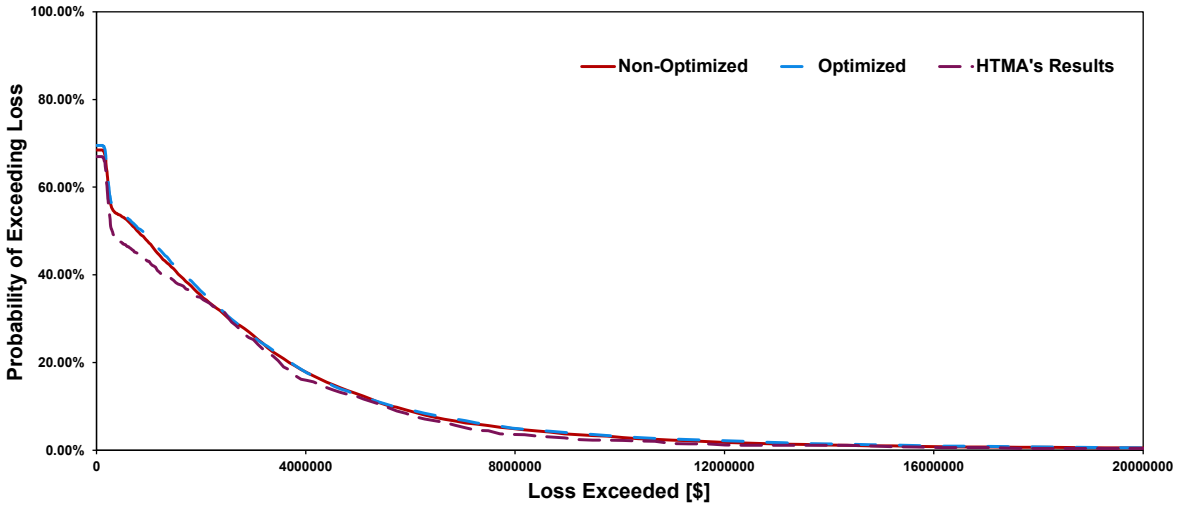


Figure 3.5: HTMA case study with approximation technique

of the array that is passed to the function; therefore, the number of N samples should not exceed 500. The current design uses Excel to demonstrate the lightweight approach and that a standard analysis tool such as Excel can support it. It is unclear why Excel crashes with more than 500 samples, and it might be a bug in Excel or the XLRisk plugin. Regardless, when implementing the analysis with a language such as R with distributed compute backends such as *future*, there is virtually no limit to the number of samples to be sampled from. Implementing the model as a dedicated application will undoubtedly remove the constraint of supporting 500 samples in the approximated technique at most.

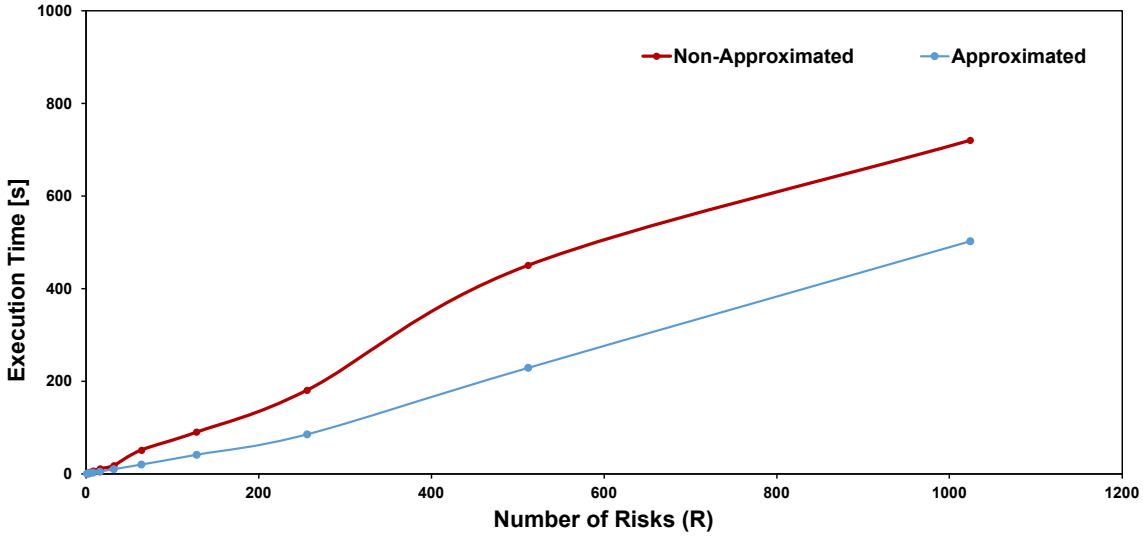


Figure 3.6: performance of increasing risks using the approximated method

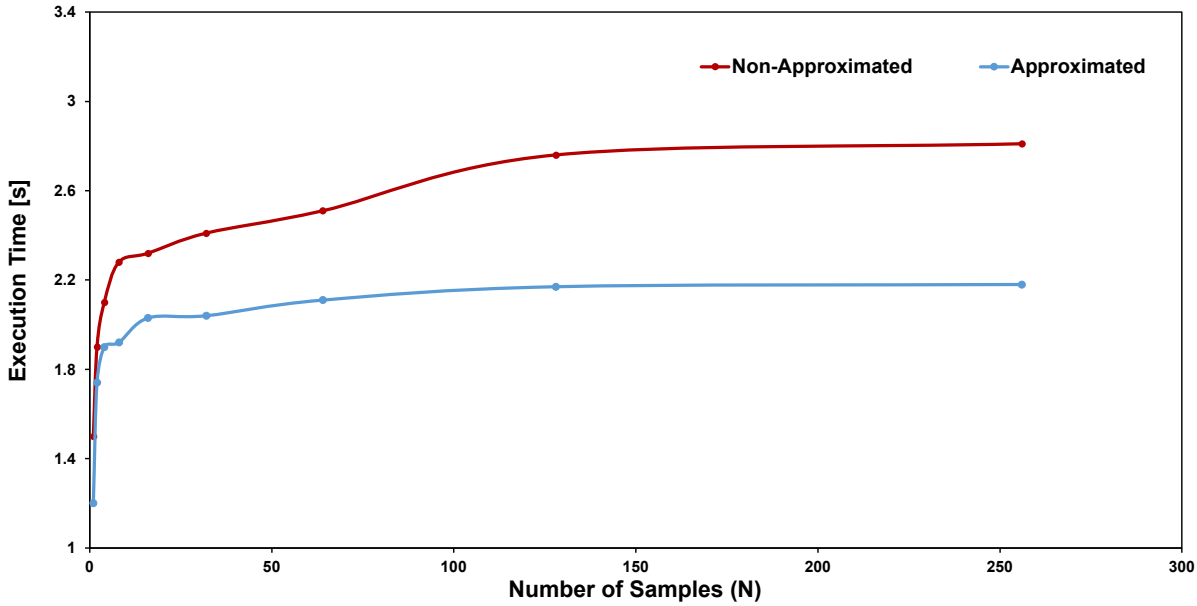


Figure 3.7: performance of increasing number of samples using the approximated method

Chapter 4

Conclusion

4.1 What Makes a Good WEP Table?

Due to their crucial involvement in decision-making, WEP tables should provide a clear and accurate representation of the likelihood of different events that reflect the judgement of experts. For this reason, it is essential to involve experts in designing WEP tables to accurately estimate the correct mapping that matches the application under consideration. Therefore, using *good* WEP tables during risk analysis is vital for better loss estimation, but what makes a *good* WEP table?

Our definition of a *good* WEP table is based on our results and findings in this study. We identified the key elements necessary for a WEP table to enhance the accuracy and reliability of risk analysis.

In Section 3.1, we introduced two experiments to measure our model's robustness. During the wrong likelihood estimation test, we found that having more entries in WEP tables will make our model less prone to likelihood estimation mistakes. Therefore, we deduce that having more likelihood mappings in WEP tables will help solve the inaccurate estimation problem during risk analysis.

Originally, intelligence analysts developed WEP tables because it solves the problem of inaccurate estimation for real numbers. Experts created WEP tables to provide a more nuanced definition of uncertainty to avoid misunderstanding and miscommunications during risk analysis. Therefore, the main idea of WEP tables is to abstract a range of probabilities into one estimative word for more informed decisions.

Based on our findings, we deduce that having more likelihood in a WEP table will make the analysis robust against mistakes. Considering the facts mentioned above, it's crucial to note that improperly designing a table with a lot of estimative words can result in a significant loss. This loss undermines the primary benefit of using WEP tables, which is to prevent miscommunication during risk analysis. Therefore, it is good to have more likelihood mappings, but an expert should carefully design it to ensure that the table follows best practices for creating WEP tables.

4.2 Future Work

This section provides an overview of the existing literature and research efforts that contribute to developing more quantitative analysis methods. Also, we compare our model with existing works and discuss future work to improve our model.

RISKEE [18] is a quantitive risk analysis method that is based on FAIR method [12], attack trees, and Diamond model [7]. The RISKEE method constructs a graph that contains different attack events, where each node in the graph represents an event. The method uses forward and backward propagation algorithms to evaluate the impact cost for the risks. A significant limitation of this approach is that it has a high computational cost when increasing the number of nodes in the graph, which makes it infeasible to be applied to large systems. Compared to RISKEE, our model is more scalable and has a lower computational cost, making it suitable for more applications.

Other probabilistic methods have been introduced, like MAGIC [4], which is a method that evaluates the probability of occurrence of cybersecurity risk. MAGIC is a probabilistic model that computes the likelihood of occurrence based on the organization's posture, like awareness of the employees, maturity and complexity, etc. The model is designed to consider all aspects of an organization that might affect cybersecurity incidents. After evaluating the likelihoods, MAGIC uses its outputs with the model in HTMA [13], or FAIR [12] to compute the cost for the loss and complete the quantitative analysis. Compared to MAGIC, our model is more generic and does not depend on the organization's properties, so cybersecurity experts can use our model for risk estimation and decision-making regardless of the organization's posture or background.

Despite the promising results of our study, there are several areas for future research and improvements. Firstly, our model heavily depends on the WEP tables collected and their assigned ranges. In future work, we plan to find a way to collect WEP tables based on geographical locations, as expert estimations for WEP tables highly depend on the environment that they are experiencing.

Another area of improvement is to design the model to be more customizable and add more configuration options for better usability. Furthermore, we aim to integrate our model with the method in [27] so we can have the feature to customize our model for each organization. Also, we aim to improve the performance of our model to handle more risks and reduce the overall execution time.

Lastly, we aim to implement the model with other languages, such as R, and make it compatible with different risk analysis methods, like the FAIR model [12]; this will help make our approach more diverse and suitable for all applications.

4.3 Conclusion

In this thesis, we introduced a spreadsheet-based approach for quantifying cybersecurity risks. We leveraged linguistic likelihoods and WEP tables to establish a way to convert verbal statements into quantitative values. Our cost calculation method is based on the approach in HTMA [13]. Furthermore, we used the open-source Excel add-in [26] to do MCS and estimate the loss for cybersecurity risks. Also, we compared our approach with existing methods [13, 27] by analyzing the LECs. Moreover, we conducted experiments to measure the performance and robustness of our model. We concluded that our model is robust against inaccurate WEP mappings but not robust against inaccurate likelihood estimation. Finally, we introduced another approximated technique which reduces computational cost and time to help risk analysts obtain faster preliminary results without running the full model. The proposed method is expected to provide organizations with a reliable tool for quantifying cybersecurity to help in decision-making and choosing risk mitigation actions.

References

[1] *Scenario-Based FMEA: A Life Cycle Cost Perspective*, volume Volume 5: 14th Reliability, Stress Analysis, and Failure Prevention Conference; 7th Flexible Assembly Conference of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 09 2000.

[2] AWS. What is The Monte Carlo Simulation? https://aws.amazon.com/what-is/monte-carlo-simulation/.

[3] Scott W. Barclay, Rex V. Brown, Clinton W. Kelly, Cameron R. Peterson, and Lawrence D. Phillips. Handbook for decision analysis. ERIC, 1977.

[4] Massimo Battaglioni, Giulia Rafaiani, Franco Chiaraluce, and Marco Baldi. MAGIC: A Method for Assessing Cyber Incidents Occurrence. *IEEE Access*, 10:73458–73473, 2022.

[5] Peter Bonate. A Brief Introduction to Monte Carlo Simulation. *Clinical pharmacokinetics*, 40:15–22, 02 2001.

[6] John B. Bowles. An assessment of RPN Prioritization in a Failure Modes Effects and Criticality Analysis. *Annual Reliability and Maintainability Symposium, 2003.*, pages 380–386, 2003.

[7] Sergio Caltagirone, Andrew D. Pendergast, and Chris Betz. The Diamond Model of Intrusion Analysis. 2013.

[8] Government Of Canada. Cyber threat bulletin: The cyber threat to Canada's electricity sector. https://cyber.gc.ca/en/guidance/cyber-threat-bulletin-cyber-threat-canadas-electricity-sector, 2020.

[9] Government Of Canada. Cyber threat bulletin: Cyber threat activity related to the Russian invasion of Ukraine. https://cyber.gc.ca/en/guidance/cyber-threat-bulletin-cyber-threat-activity-related-russian-invasion-ukraine, 2022.

[10] Matthew Conlen. Kernel Density Estimation Visualization tool. https://mathisonian.github.io/kde/.

[11] Center for Internet Security (CIS). Words of Estimative Probability, Analytic Confidences, and Structured Analytic Techniques. https://www.cisecurity.org/ms-isac/services/words-of-estimative-probability-analytic-confidences-and-structured-analytic-techniques, 2022.

[12] Jack Freund and Jack Jones. *Measuring and Managing Information Risk: A FAIR Approach*. Butterworth-Heinemann, USA, 2014.

[13] Douglas Hubbard and Richard Seiersen. How to Measure Anything in Cybersecurity Risk. 2016.

[14] Texas Instruments. TMS320F2800157. https://www.ti.com/product/TMS320F2800157, 2023.

[15] IPCC. Guidance Notes for Lead Authors of the IPCC Fourth Assessment Report on Addressing Uncertainties. Technical guidance, Intergovernmental Panel on Climate Change, July 2005.

[16] Sherman Kent. Words of Estimative Probability. *CIA Center For The Study Of Intelligence*, 1964.

[17] Michael Krisper. Problems with Risk Matrices Using Ordinal Scales. 2021.

[18] Michael Krisper, Jürgen Dobaj, Georg Macher, and Christoph Schmittner. RISKEE: A Risk-Tree Based Method for Assessing Risk in Cyber Security. In *Systems, Software and Services Process Improvement*, pages 45–56. Springer International Publishing, 2019.

[19] Ponemon Institute LLC and Accenture. Cost of cyber crime study. Technical report, Ponemon Institute LLC–Accenture, North Traverse City, MI, USA, 2017.

[20] Georg Macher, Harald Sporer, Reinhard Berlach, Eric Armengaud, and Christian Kreiner. SAHARA: A Security-Aware Hazard and Risk Analysis Method. *Design, Automation & Test in Europe Conference & Exhibition*, pages 621–624, 2015.

[21] Mohammad Modarres. *Risk analysis in engineering: techniques, tools, and trends*. CRC press, 2006.

[22] Johnathan Mun. *Modeling Risk: Applying Monte Carlo Simulation, Real Options Analysis, Forecasting, and Optimization Techniques*. 2006.

[23] National Intelligence Council (NIC). Analytic Standards. https://www.dni.gov/files/documents/ICD/ICD%20203%20Analytic%20Standards.pdf, 2007.

[24] Douglas Ott. Words Representing Numeric Probabilities in Medical Writing Are Ambiguous and Misinterpreted. *JSLS: Journal of the Society of Laparoscopic Robotic Surgeons*, 25:e2021.00034, 07 2021.

[25] Randolph H. Pherson. Critical thinking for strategic intelligence. 2012.

[26] Pyscripter. XLRisk. https://github.com/pyscripter/XLRisk, 2021.

[27] Paolo Santini, Giuseppe Gottardi, Marco Baldi, and Franco Chiaraluce. A Data-Driven Approach to Cyber Risk Assessment. 2019:1–8, 2019.

[28] Wikipedia. Kernel Density Estimation. https://en.wikipedia.org/wiki/Kernel_density_estimation.