AI-Powered Delay Prediction for Portfolio Management

Gaspar Rocha

AI-Powered Delay Prediction for Portfolio Management

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Gaspar Rocha
born in Vila Nova de Gaia, Portugal

TUDelft

Software Engineering Research Group
Department of Software Technology
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
`www.ewi.tudelft.nl`

DevOn
Brassersplein 1, 2612 CT Delft
`www.devon.nl`

AI-Powered Delay Prediction for Portfolio Management

Author: Gaspar Rocha
Student id: 5704340

Abstract

Delayed software projects are one of the biggest threats to the integrity of many project portfolios. If portfolio managers were able to foresee delays, they could better manage risks, make adjustments to the planning and reduce delay propagation. In their 2023 paper ”Dynamic Prediction of Delays in Software Projects using Delay Patterns and Bayesian Modeling”, Kula et al. propose an AI solution for the problem of ineffective delay prediction of software projects. Even though Kula et al. achieved positive results, they are bound to ING’s data, and thus may not be representative of software projects in other companies or industries. This thesis builds on Kula et al.’s work by applying the same methodology to a new dataset - Coca-Cola Hellenic’s Project Portfolio. By doing so, it assesses the robustness and generalisability of Kula et al.’s delay prediction model. The results clearly indicate that the model was unsuccessful at Coca-Cola Hellenic, as it proved no better than random guessing. Differences in dataset size and quality were identified as the primary cause for the lack of performance. Furthermore, contextual factors were likely a major contribution to the difference in results, namely differences in industry, organisational structure and agile maturity. These findings are valuable to anyone attempting to replicate this solution, or to organisations aiming to adopt AI-powered analytics. Future research directions are suggested, such as a requirement framework for AI solutions and further replication of Kula et al.’s work in different contexts.

Thesis Committee:

Chair:	Prof. Dr. A. van Deursen, Faculty EEMCS, TU Delft
University supervisor:	Prof. Dr. Ir. R. van Solingen, Faculty EEMCS, TU Delft
Company supervisor:	Dr. S. McGirr, DevOn
Committee Member:	Prof. Dr. U. Gadiraju, Faculty EEMCS, TU Delft

Preface

This thesis is the end of a journey. More than an academic achievement, it is a token of my life. It is also an opportunity for me to look back and reflect on the path I have taken and the people I have shared it with.

I want to start by thanking my academic supervisors, Professors Arie van Deursen and Rini van Solingen. Professor Arie, thank you for taking me into your research group. Your contributions of direction and scientific rigour have greatly influenced this work and my perspective on the meaning of true research. Professor Rini, this thesis started with you. Before there was even an idea, there was an understanding of interests, goals, and ambitions. Since that first talk we had in the Aula cafeteria, you have been my closest partner in this work. You have stood by my side week after week, checking in on my progress, but also checking in on me. Thank you.

I also want to thank Dr. Shaun McGirr, my company supervisor. You really took me by the hand and welcomed me into the (dirty) world of data. Day in and day out you were there, making sure I had what it took to get the job done. Thank you for all the moments where you carried some of the load and took some weight off my shoulders. You have given me some of the realest advice I have ever heard, and I cannot think of a better supervisor to have for this project.

I want to acknowledge my final committee member, Professor Ujwal Gadiraju. Thank you for accepting my invitation to be a part of this committee. You have gone out of your way to be here, and I hope the significance of this work serves as a worthy return for your time and effort.

I now want to thank the people of DevOn and the Waada group. I am grateful for the time I spent there. I want to thank Martin van Langen of Xeleron for all the support, be it in finding a company willing to share their data or helping me navigate the business world. I also want to thank Lisa Spruit and Floris Beljaars for welcoming me into the company and always showing interest in my work. Finally, I want to thank the closest friends I made there, Amir, Aron, Bob, Emani, Hala and Sophie.

I want to acknowledge the Coca-Cola Hellenic Bottling Company, and in particular Paul Beelen, head of their PMO. Paul, this project would not have been possible without you. Thank you for the trust you placed in me, I hope I was able to live up to it.

On a more personal note, I want to thank the closest friends I made in my time in the

Netherlands. Alessandro, Arnaud, João, and Kaushal, I am so happy that I met you. I can think of many great conversations I had with each of you, but especially when we were all together. I wish that we had been able to spend more time together, it feels like the Master's has gone by in a flash, and I hope that we will continue this friendship, wherever life takes us.

I also want to thank the good people of Aan't Verlaat. Amrita, Aniket, Bhouumesh, Chiara, Dan, Emily, Gustavo, Joseph, Stella, and Vasko. I still think it is crazy that we got along so well in a house of 11 people. I can truly say I had the best housemates in the world. Over two and a half years, you made that house feel like home.

To all the other friends I made through the years, the people from CS, the guys in the football group, my psychologist, and half of the people from the Applied Physics Master's, thank you.

As I look back beyond my time in the Netherlands, I want to thank the people in my life from Portugal who have always been there for me.

To my closest friends. Miguel, David, Marco e Nando, os grandes Pomadinha, obrigado por ainda pensarem em mim. Voltar a Portugal é sempre estranho, sinto-me em casa, mas, ao mesmo tempo, sinto-me um estrangeiro, turista. Fico muito feliz sempre que vos vejo e me fazem sentir como se nunca tivesse estado fora. A distância é longa, mas quero muito manter esta nossa ligação.

To Hugo. Obrigado por toda a companhia que me deste. Embora tenhamos estado afastados, senti, nestes últimos anos, alguns dos nossos momentos de maior proximidade. Espero continuar a ser teu amigo para sempre.

To my family. Nem sei bem o que escrever. Julgo que sempre tive noção da vossa importância no meu dia a dia, mas nunca a senti tão verdadeiramente até que me mudei para aqui. Ainda hoje me custa, e penso que vai custar para sempre, pensar em todos os almoços que perdi, todas as tardes em casa da Avó, todas as biscas italianas, as Páscoas, os Carnavais, os São Martinhos. São a minha maior dor. Hoje sei o que é Saudade. Quero agradecer-vos por tudo o que me deram, pelas pessoas que são e, acima de tudo, pelo amor que sinto de vós.

To my parents. Mãe, Pai, obrigado por tudo. Tudo aquilo que sou é vosso. Obrigado por me proporcionarem esta experiência. Obrigado pelo carinho e preocupação que tiveram nos meus momentos mais difíceis. Obrigado pelo que sofreram, e sei que sofreram, para que hoje eu pudesse estar onde estou, e ser quem eu sou. Espero conseguir fazer-vos sentir orgulhosos. Obrigado.

To Nara. Qualquer palavra é-te uma injustiça, portanto vou ser breve. Obrigado pela companhia, pela entrega, pela compaixão e sobretudo, pelo amor.

I saved these last words for myself. These were the hardest three years of my life. I hurt myself a lot, and at times I didn't think I was going to make it. I want to thank myself for reaching out for help, for not giving up. I want to apologise to myself for all the self-hatred I carried. I can now say that I am grateful to be here, and grateful for everything I have learned and experienced. I will keep on trying to be kinder to myself, and, hopefully, keep moving forward. Thank you.

Gaspar Rocha
Delft, the Netherlands
May 13, 2025

Contents

List of Figures

Chapter 1

Introduction

Project Portfolio Management (PPM) can be described as the set of practices and principles organisations use to strategically and holistically manage a collection of projects. It is the mechanism by which organisational goals are transformed into actionable projects. It originated from the revolutionary financial theory of Harry Markowitz [1], and became its own stand-alone field in the late 20th century, greatly due to the work of Cooper et al. [2], [3], [4], [5].

In practise, PPM consists of two activities: (re)prioritisation of projects and (re)allocation of resources. Although they may seem simple, these two actions gain exponential complexity as the number of projects and their interdependency increases [6]. The (re)prioritisation process requires a deep understanding of business goals, organisational strategy, risk management and potential returns. (Re)allocation, on the other hand, focuses primarily on resource optimization and scheduling.

Managing a Project Portfolio is a difficult task. While the goal is clear - maximizing the portfolio's value - fundamental problems arise from this simple premise. First off, there is no standard way to measure value [7], so prioritisation is bound to be subjective. Second, estimating the time and resource requirements of a project is also extremely complex and prone to error [8].

Risk management, particularly predicting project delays, is another major challenge in PPM [9], [10]. Software projects tend to be 30 to 40% delayed, so, when combined with ineffective risk management and delay prediction, they can threaten the integrity and validity of an entire portfolio [11]. If portfolio managers were able to foresee delays, they could better manage risks, make adjustments to the planning and reduce delay propagation.

Artificial Intelligence (AI) could be an interesting technological solution for Portfolio Management [12]. As portfolios become increasingly more complex and organisations demand simplified procedures, AI may be the key to optimize and streamline operations. One of AI's great advantages is its versatility. So, its adaptability could be put to use in different parts of portfolio management, from portfolio optimization [13] to risk management [9], or even strategic alignment [14].

In their 2023 paper [1], Kula et al. [11] propose an AI solution for the problem of ineffec-

[1] Elvan Kula, Eric Greuter, Arie van Deursen, and Georgios Gousios. Dynamic Prediction of Delays in

tive delay prediction of software projects. They developed a dynamic Bayesian model that utilizes delay patterns to accurately and continuously predict the overall delay of software projects at the Dutch bank ING. The results presented in this paper are extremely interesting because they demonstrate the predictive power of an AI model capable of learning and evolving over a project's life cycle, as well as the effectiveness of using delay patterns as an indicator of overall project delay.

Even though Kula et al.'s work achieved positive results, they are bound to ING's data, and thus may not be representative of software projects in other companies or industries. In other words, their findings are not generalisable. To be so, as Kula et al. recommend [11, p. 1021], the same method must be replicated with different settings, in order to reach more general conclusions.

This thesis builds on Kula et al.'s work by applying the same methodology to a new dataset from a different industry. By doing so, it assesses the robustness of AI-based delay prediction in Project Portfolio Management and explore its practical implications. This was made possible through a collaboration with DevOn and the Coca-Cola Hellenic Bottling Company (CCH). DevOn is a software delivery and DevOps consultancy company based in the Netherlands and India. At the start of the project, DevOn proposed this research idea to its clients and Coca-Cola Hellenic accepted.

The Coca-Cola Hellenic Bottling Company is a strategic bottling partner of The Coca-Cola Company. It is responsible for producing, packaging, distributing, and selling Coca-Cola products across three continents. CCH operates in a quite different industry from ING. The consumer goods industry is faster moving, less regulated and more demand-driven than the finance industry. This difference in industry requires distinct operational structures and ways of working. Despite this contrast, these are two companies who have made great investments in agility and in data-driven decision-making. For all these reasons, the Coca-Cola Hellenic Bottling Company is a great setting to replicate Kula et al.'s study.

This thesis is structured around two central research questions. The first:

> **RQ1** ”What are the connections between Artificial Intelligence and Project Portfolio Management?”

is addressed through a literature review that establishes the theoretical foundation of the study.

The second:

> **RQ2** ”How effective is Kula et al.'s solution when applied to Coca-Cola Hellenic's Project Portfolio?”

focuses on the empirical component, assessing the applicability and performance of Kula et al.'s solution in a new industrial context.

The rest of the thesis is comprised of six chapters. First, a literature review on Artificial Intelligence and Project Portfolio Management, where the first research question is

Software Projects using Delay Patterns and Bayesian Modeling. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1012–1023, New York, NY, USA, 11 2023. ACM.

answered. Then, a detailed description of Kula et al.'s paper, ”Dynamic Prediction of Delays in Software Projects using Delay Patterns and Bayesian Modeling”. The fourth chapter, Approach, defines the methodology and rationale for this experiment. The following chapter presents its results and answers the second research question. Subsequently, in the Discussion chapter, the broader implications of the results are interpreted, as well as this study's limitations and recommendations of future work. Finally, the last chapter states the conclusions of the thesis.

Chapter 2

Artificial Intelligence and Portfolio Management

This chapter answers **RQ1** through a literature review. It is organized as a narrative literature review [15] and follows some of the principles of structured literature review (SLR) methods [16]. The first section describes the process of collecting a comprehensive body of literature, while the second presents the analysis and synthesis of that literature. Finally, the last section contextualizes the results of the study with the overarching research goals of the thesis.

2.1 Collection of literature

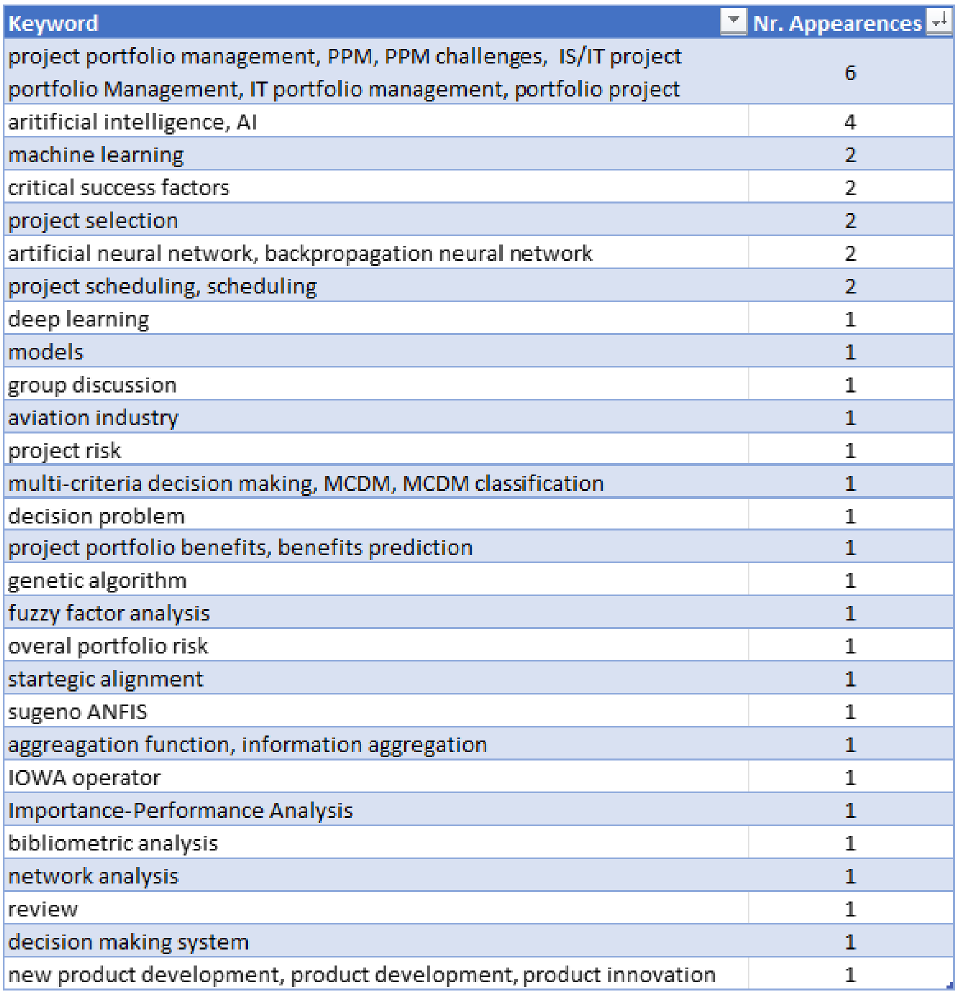
The employed methodology consists of four stages: identifying keywords and developing a search string, a first round of filtering, citation tracking, and a final, second round of filtering. The first step, combined with the citation tracking, will ensure the collection of a vast, comprehensive body of literature. Filtering will then guarantee that the final corpus contains only relevant articles, allowing a more focused analysis of the literature. Together, these processes assure that most relevant articles are collected and analysed. To standardize the research process and to ensure a high level of credibility, the research domain was limited to academic papers.

2.1.1 Identifying keywords and developing a search string

The first step in this stage was conducting an exploratory search for articles about AI in portfolio management. The search was performed in the Google Scholar database, with the search string *(”artificial intelligence” OR ”AI”) AND (”agile portfolio management” OR ”project portfolio management” OR (”project” AND ”portfolio management”))*, and 15 academic works were selected (Appendix A.1).

These publications were then scanned and compiled to identify the main keywords of the domain. As Table 2.1 indicates, the use of keywords in this discipline is quite inconsistent, but some do stand out. So, based on the table, a first version of the search string was created: *(”artificial intelligence” OR ”AI” OR ”machine learning” OR ”ML” OR ”neural*

Table 2.1: Keyword compilation from exploratory search



Keyword	Nr. Appearences
project portfolio management, PPM, PPM challenges, IS/IT project portfolio Management, IT portfolio management, portfolio project	6
aritificial intelligence, AI	4
machine learning	2
critical success factors	2
project selection	2
artificial neural network, backpropagation neural network	2
project scheduling, scheduling	2
deep learning	1
models	1
group discussion	1
aviation industry	1
project risk	1
multi-criteria decision making, MCDM, MCDM classification	1
decision problem	1
project portfolio benefits, benefits prediction	1
genetic algorithm	1
fuzzy factor analysis	1
overal portfolio risk	1
startegic alignment	1
sugeno ANFIS	1
aggreagation function, information aggregation	1
IOWA operator	1
Importance-Performance Analysis	1
bibliometric analysis	1
network analysis	1
review	1
decision making system	1
new product development, product development, product innovation	1

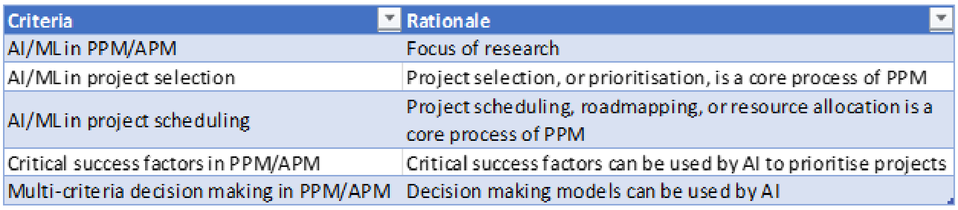
network") AND ("agile portfolio management" OR "project portfolio management" OR ("project" AND "portfolio management")).

Before starting the collection of results, a criteria had to be established to properly assess each iteration of the search string. The goal of this criteria was to convey whether or not the search string was able to collect the most relevant literature in the field. With this purpose in mind, another exploratory search was conducted in Google Scholar to collect the 30 most relevant results for AI in portfolio management. The articles already found the in first exploratory search were excluded, and then the 10 most relevant were selected as the benchmark **??**. Thus, the passing criteria for each search string iteration became its ability to gather all of the benchmark papers in its search results.

The search and collection of literature was made using Publish or Perish [1], a software

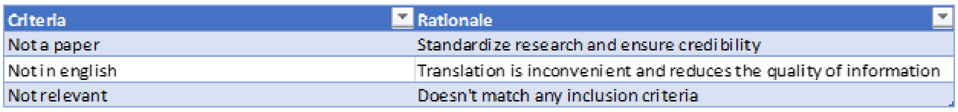
[1] Publish or Perish, by Harzing, A.W. (2007), available from `https://harzing.com/resources/publish-or-perish`

Table 2.2: Inclusion Criteria



Criteria	Rationale
AI/ML in PPM/APM	Focus of research
AI/ML in project selection	Project selection, or prioritisation, is a core process of PPM
AI/ML in project scheduling	Project scheduling, roadmapping, or resource allocation is a core process of PPM
Critical success factors in PPM/APM	Critical success factors can be used by AI to prioritise projects
Multi-criteria decision making in PPM/APM	Decision making models can be used by AI

Table 2.3: Exclusion Criteria



Criteria	Rationale
Not a paper	Standardize research and ensure credibility
Not in english	Translation is inconvenient and reduces the quality of information
Not relevant	Doesn't match any inclusion criteria

program that retrieves and analyzes academic citations. The search was performed in the Google Scholar and Scopus databases. The first iteration of the search string included all of the benchmark papers, in a total of 215 academic works.

2.1.2 First round of filtering

Filtering consisted of reviewing the titles and abstracts of each work and assessing it on the inclusion and exclusion criteria, Tables 2.2 and 2.3, respectively. It is worthy to note that many of exclusions were due to the focus on financial portfolios, an established field of literature with similar keywords and topics, but fundamentally different in essence. After the filtering, 42 papers remained.

2.1.3 Citation tracking

To ensure no major publications were missed, citation tracking was used on the 10 most cited papers in the list. Google Scholar was selected for citation tracking because it has a good mixture of precision and sensitivity [17], meaning it is likely to find the most relevant articles that cited the selected papers. For each paper, the 5 most relevant works that cited it were added to the list, a total of 50 articles.

2.1.4 Second round of filtering

The second sound of filtering was the same as the first, only applied to the works gathered through citation tracking. The final body of literature consisted of 48 papers (Appendix A.2).

2.2 Analysis and synthesis

Project Portfolio Management is a complex task. It is the process by which organisations aim to balance multiple projects, allocate resources efficiently, and achieve strategic goals.

PPM faces significant challenges, particularly in an evermore dynamic and uncertain market environment. Artificial intelligence, as a tool, has the potential of addressing these issues by enhancing analysis and decision-making, thus empowering Portfolio Managers.

One of the most important challenges of PPM is *optimisation*. Limited resources demand choices, and complex value judgements make prioritization difficult. Portfolio Managers must attempt to maximize the value of their portfolio, either by increasing their return on investment or optimising the portfolio's strategic alignment [13]. However, traditional optimisation methods often struggle with this multi-objective nature, and so result in sub-optimal solutions [13], [18].

Another major challenge of PPM is *risk management*. Projects within a portfolio tend to be interdependent, which causes the issue of project failure or delay propagating across the entire portfolio [9], [10]. Conventional risk assessment methods rely on static models, which fail to handle the dynamic and uncertain nature of PPM. Zhang et al. [9] have shown that risks related to organisational management, resource allocation, and stakeholder communication are particularly difficult to manage, especially during the early stages of the project.

Finally, *strategic alignment* is also a challenge within PPM. Organisations strive to align their portfolios with broader strategic goals, but changing priorities and conflicting criteria make it hard to maintain alignment. Once again, the pressure of dynamic market conditions emphasizes the struggle. This problem must also be monitored regularly, as studies have shown that even well-designed portfolios can drift out of alignment if not continuously adjusted [14].

Over the last 15 years, AI has been a part of innovative solutions to the aforementioned challenges. These solutions have evolved over time, with researchers building on each other's work and developing new approaches to combat the limitations of earlier methods.

In the area of *optimisation*, some of the earliest AI solutions were multi-objective evolutionary algorithms. For example, the mPOEMS algorithm proposed by Kremmel et al. [13] showed that evolutionary algorithms are able to explore complex solutions and identify optimal portfolios. Later studies expanded on this approach, combining evolutionary algorithms with neural networks to improve performance and scalability [19]. Curiously, this relationship between evolutionary algorithms and neural networks was found to be bidirectional. In 2013, a study showed that neural networks were capable of handling complex interdependencies between projects, but demanded expensive computational resources [20]. Later on, genetic algorithms enhanced the performance of neural networks, eliminating this issue [21]. More recently, fuzzy logic was proposed as a way to handle uncertainty in optimisation [22], offering a new perspective on balancing competing objectives in dynamic environments. Farid et al. [18] embraced uncertainty and developed a binary decision tree algorithm that was shown to increase the value of the portfolio by presenting multiple options for portfolio decisions including the effects of uncertainty.

As for *risk management*, the appearance of AI solutions marked a shift from older, static models. Bayesian models had a high impact in this field [9]. One study proved that fuzzy Bayesian networks could dynamically assess risks across the project life cycle and provide actionable insights for risk reduction [10]. Other studies implemented machine learning strategies to predict project risk in real-time. Notably, the Sugeno ANFIS model combined fuzzy logic with neural networks to predict overall portfolio risk while incorporating expert knowledge [23].

Finally, AI has also transformed the challenge of *strategic alignment* in PPM. A common technique to tackle this problem was using multi-criteria decision analysis to rank projects based on strategic alignment [14], [24]. Even though these methods achieved certain success, they were limited by their reliance on static criteria. More recently, researchers have been combating this issue by using AI methods that can dynamically adjust prioritisation criteria based on changing conditions [24]. Notably, Holmes et al. [7] proposed a hierarchical value framework that uses natural language processing to continuously update project valuations, guaranteeing alignment with organisational goals.

To conclude, AI has been a powerful tool to address some of the major challenges of Project Portfolio Management. One of the biggest strengths of AI is its dynamic and adaptive ability. Time and again, researches have used AI to handle rapidly changing conditions. As both the fields of AI and PPM mature, it is more and more likely for this relationship to continue to develop.

2.3 Discussion

This chapter, and particularly the previous section satisfy **RQ1**. This literature review has shown that Artificial Intelligence and Project Portfolio Management are strongly connected across multiple subjects, namely optimisation [13], [18], risk management [9], [10], and strategic alignment [14], [24]. More areas of connection may exist, but, due to the limitations of this study, they were not found. Restricting the research domain to academic papers may have excluded books or other mediums with pertinent information. The choice of keywords was another limiting factor. A more comprehensive search string would have surely collected a larger number of interesting articles. Likewise, a more exhaustive citation tracking would have had a similar effect. Therefore, researchers are encouraged to conduct literature reviews on this topic that do not suffer from the same limitations.

Kula et al.'s paper ”Dynamic Prediction of Delays in Software Projects using Delay Patterns and Bayesian Modeling” [11] is a surprising omission in this literature review. The main themes of this article fit perfectly with the risk management AI solutions, but it never came up in the collection process. A reason for this could be the lack of a direct PPM reference in Kula et al.'s paper. While risk management is mentioned as a key proponent for their solution, it is never contextualised for portfolio management. This omission might be indicative of an academic gap, distancing some risk management literature, particularly about delay prediction, from project portfolio literature. This potential disconnect justifies a deeper examination of Kula et al.'s work to understand its relevance to PPM and to determine how its approach could improve AI-based risk management. The following chapter explores Kula et al.'s paper in depth, analysing its approach, findings, and related work.

Chapter 3

Dynamic Prediction of Delays in Software Projects using Delay Patterns and Bayesian Modeling

”Dynamic Prediction of Delays in Software Projects using Delay Patterns and Bayesian Modeling” was written as part of Dr Kula’s PhD research at ING. In it, Kula et al. tackle the problem of ineffective delay prediction of software projects. This research goal was supported by empirical data that indicates that software projects, all across the industry, are delayed, on average, by 30-40% [11, p. 1012]. It was also a goal of development teams at ING to become more predictable.

3.1 Existing solutions

Some state-of-the-art solutions for this issue include global AI models. In other words, models that collect all of their data at the start of a project and thus make only one prediction. While this type of model may work well in traditional, waterfall-style development methods, they are unfit for more modern agile settings. Projects developed under this framework are more volatile, in order to respond to changing customer needs and market circumstances. The static quality of global AI models makes them unsuitable for the changing variables of an agile project. A direct response to this issue was global iterative models, i.e. global AI models applied in an iterative fashion. Although better than the original solutions, these models continue to ignore the evolution of variables throughout a project’s development. Kula et al. propose a new type of solution - a dynamic model, capable of adapting and learning from the changes in an agile project’s development [11, p. 1013].

Kula et al. went even further with their innovation, taking inspiration from the railways and air transport sectors, by adding delay patterns to their solution. Previous research had found that delays tend to form certain patterns over time. So, finding a similarity between an ongoing project’s delay and a pattern in historical data can provide an indication on the future delay development of that project.

3.1.1 Milestones

This innovative solution had a three stepped approach: collecting data, clustering to discover delay patters, and developing the dynamic model. Before collecting data, Kula et al. made a pivotal decision that shaped the rest of the study - imposing milestones. Software projects can have wildly different lengths. The absolute delay of an 8 week project will look nothing like the absolute delay of a 6 month project, and that makes it harder to identify patterns. It can be argued though, that projects will experience delay at the same relative moment in their development. So, by normalising the duration of the project, it is then possible to identify reoccurring patterns. However, these projects will have different levels of granularity, because the longer projects have more iterations, and thus more datapoints. In order to highlight existing patterns, and facilitate their discovery, it is important to create a unified timeline, or a sequence of regularly-spaced intervals. Kula et al. called these intervals Milestones. Since different projects have different development and iteration times, milestones cannot be a fixed time interval. Instead, they are based on completion rate. Kula et al. [11, p. 1014] explained: ”for example, an epic that consists of 20 iterations will achieve its 10% milestone after completing the initial two iterations”. Finally, the total number of milestones will determine the granularity of the collected data, and so, the granularity of the patterns. Since progress updates are given at the end of each iteration, and more than 80% of epics at ING are at least 10 iterations long, Kula et al. decided that their analysis would use 10 milestones [11, p. 1014].

3.2 Approach

3.2.1 Step 1 - Data Collection and Preprocessing

Data collection

For data collection, Kula et al. extracted the data from ING’s backlog management tool, ServiceNow, containing 7463 epics delivered by 418 teams over a 5 year period. To avoid noise and missing values, epics with a status other than ’Completed’, with missing planned or actual delivery dates, or with no assigned team were filtered out. To guarantee at least one update per milestone, epics with less than 10 sprints were also removed. Finally, epics that exceeded two standard deviations from the mean overall delay were excluded. So, after cleaning the data, the final ING dataset was reduced to 4040 epics from 270 teams [11, p. 1015].

Delay factors

Before diving into the predicting power of AI on delay risk management, Kula et al. investigated what factors affect on-time software delivery at ING [25]. This study employed a two-phase approach. First, Kula et al. distributed a survey to a wide array of software experts and collected a set of factors they identified as affecting on-time delivery. Then, in the confirmatory phase, they applied data triangulation to validate the survey responses. Finally, Kula et al. identified 13 predictor variables that explain more than two thirds of the delay variation of epics at ING [25]. Now, Kula et al. extracted these 13 variables at the end of each milestone for every epic at ING [11, p. 1015].

Measuring delays

The last step in creating the ING dataset was measuring the delay of each epic. The Balanced Relative Error (BRE) [26] was used for this purpose, a decision based on literature that suggests that BRE is superior to the commonly used Mean of Magnitude of Relative Error [27], [28], [29]. BRE was defined as follows:

$$\text{If Act - Pln } \geq 0\text{, then BRE } = \frac{\text{Act } - \text{ Pln}}{\text{Planned Duration}}$$
$$\text{If Act - Pln } < 0\text{, then BRE } = \frac{\text{Act } - \text{ Pln}}{\text{Actual Duration}}$$

where Act is the actual delivery date and Pln is the planned delivery date. Planned Duration is the difference, in days, between the planned start date and the planned delivery date, and, logically, Actual Duration is the difference between the actual start date and actual delivery date [11, p. 1015].

3.2.2 Step 2 - Detecting delay patterns

To detect patterns across multiple epics, the time-series of similar intermediate delay values must be clustered into distinctive groups. In this work, intermediate delays were measured as the number of Delayed Story Points (DSP). More concretely, a milestone i's DSP is equal to the total number of story points delayed to the next milestone $i + 1$. The DSP values are non-cumulative and normalized, in order not to affect the clustering results [11, p. 1015].

The clustering itself was performed with an hierarchical algorithm, using Dynamic Time Warping (DTW) as its distance measure. Using hierarchical clustering was necessary, because the model must cluster time-series of different lengths when labelling epics in development. DTW, on the other hand, was used because, since it is a shape-based distance measure, it is particularly useful for time-series whose time elements do not match perfectly, which is the case due to the differing iteration lengths in the epics at ING [11, p. 1015].

Finally, Kula et al. employed the elbow method to determine the optimal number of clusters for analysis. This approach resulted in four patterns that will later be described [11, p. 1015].

3.2.3 Step 3 - Developing the Bayesian model

Three Bayesian models were developed, one global, one global iterative and one dynamic. All models were developed with the same framework and following guidelines for Bayesian data analysis in software engineering research [11, p. 1016]. Kula et al. opted for a Zero-Inflated Beta distribution because the BRE values at ING are all between 0 and 1, with a high density of zeros (42%) [11, p. 1016]. The full definition of the models is described by Equations 1-7 in page 1016 [11, p. 1016]. Sampling was performed using a Hamiltonian Monte Carlo implementation and all predictors were found to have a significant effect at the 95% level [11, p. 1017]. The results show that there was no overfitting [11, p. 1017].

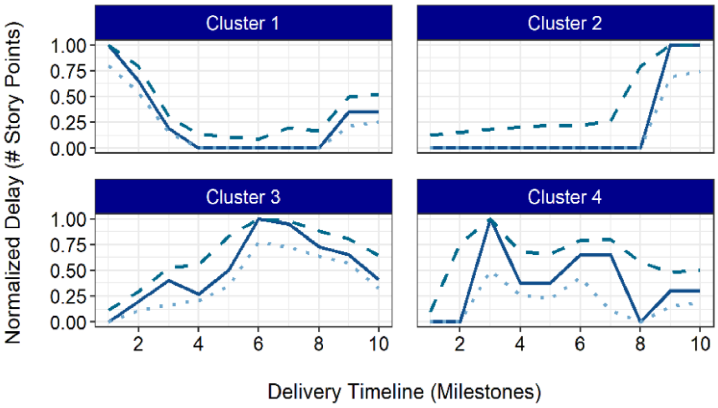


Figure 3.1: ING's delay patterns

3.3 Results

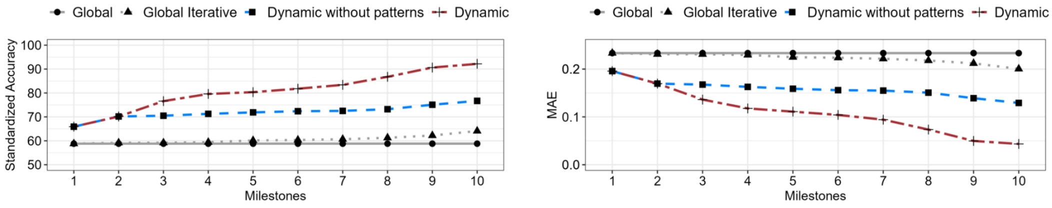
3.3.1 Delay patterns at ING

As pointed out above, using the elbow method, it was determined that 4 would be the ideal number of clusters. As such, four patterns were identified, as can be seen in Figure 3.1. The graphs include the centroids and the 25th and 75th percentile of the cluster delay distributions. The epics are clustered around the centroids with a low mean variance, which can be seen by the short distance between the 25th and 75th percentile lines. this highlights the existence of these recurring patterns [11, p. 1017]. Kula et al. performed Wilcoxon tests to determine the factors for which clusters are significantly different from the others. The factors deemed significantly different characterize the clusters, and while no causal links can be extrapolated between factors and patterns, their relationship hints at possible causes of delays. Further analysis on this subject is needed to make such conclusions [11, p. 1017]. However, one key insight that can be taken away is that patterns are indicative of overall delay. This is noticeable as the BRE values of all clusters are significantly different from each other [11, p. 1018].

3.3.2 Bayesian model results

To compare the performance between the different models, Kula et al. opted to measure the Mean Absolute Error (MAE) and the Standardized Accuracy (SA), as both have been recommended compare the performance of this type of models [11, p. 1018].

First, the performance of the dynamic Bayesian model was compared with the other Bayesian modes of development. Figure 3.2 shows the evaluation results of the global, global iterative and dynamic modes of the Bayesian models. It can be seen that, over time, the dynamic model outperforms the others. This was confirmed by a Wilcoxon test [11, p. 1019]. It can also be seen that the dynamic model with access to the delay patterns as an input feature consistently outperforms the other dynamic model after milestone 2.



(a) Standardized Accuracy over time

(b) Mean Absolute Error over time

Figure 3.2: SA and MAE results of the different Bayesian models at ING

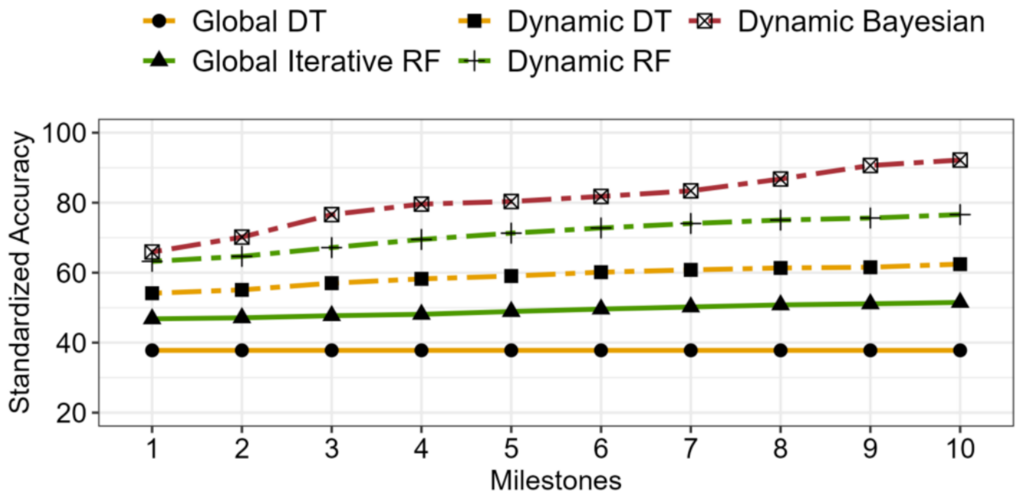


Figure 3.3: SA results of the SoTA and dynamic Bayesian models at ING

The dynamic Bayesian model was also compared with other state-of-the-art models. Choetkiertikul et al.'s Decision Tree and Random Forest models were selected [11, p. 1018]. These were trained in both global and dynamic modes. Figure 3.3 demonstrates that the dynamic Bayesian model consistently outperforms both the global and dynamic SoTA models and that, overall, the dynamic models outperform the global ones [11, p. 1020].

3.4 Main findings

The delay patterns found at ING were indicative of the overall project delay. Not only that, but their inclusion as an input feature increases the dynamic model's performance by upwards of 60% (MAE) [11, p. 1020].

Each delay pattern showed significant differences in various delay causing factors. While no causal connection can be found, the relationship between factors and patterns allows the formulation of hypotheses on the causes of delays. Further analysis and testing could lead to the discovery of actionable delay mitigation measures [11, p. 1020].

Another important finding is that dynamic models consistently outperform their global and global iterative counterparts. This was clearly observed between the different types of Bayesian models (Figure 3.2) and the SoTA baselines (Figure 3.3). This indicates that dynamic models are better suited for development settings with a high degree of change. Furthermore, the dynamic Bayesian model showed significant accuracy improvements over time, which reinforces the ability of dynamic models to predict long-term delay [11, p. 1021].

Finally, the dynamic Bayesian model outperformed the dynamic Decision Tree and Random Forest models. It also achieved the highest increase in accuracy over time. This confirms the idea that Bayesian models are more effective in quantifying and updating uncertainty over time. Since Bayesian models provide detailed information about the uncertainty of a prediction, they allow companies to be transparent in their decision making and increase their confidence in project plans [11, p. 1021].

3.4.1 Future work

As alluded to before, Kula et al. suggest the investigation of the causal relationship between delay factors and delay patterns. It is also pondered if systematic effects affect delay patterns. For example, seasonal events could have an impact on yearly delay patterns. Thus, the opportunity arises to study the seasonality and time dependency of delay patterns using pattern matching [11, p. 1021].

Previous studies have found that software projects are often delayed because of bugs or incidents. While the dynamic Bayesian model takes these into account, it will only make a new prediction every milestone. An event-driven model could be triggered by these occurrences to make a new prediction, allowing companies to react quicker and with more confidently [11, p. 1021].

Kula et al. also suggest the development of a model that considers the interaction between project deliveries. A single delayed project may cause a chain reaction, and so, future models should track the relationships between projects and study the branching effects of delays [11, p. 1021].

Lastly, and perhaps most importantly, Kula et al. acknowledge that their study may not be representative of software projects in other organisations and settings. Different industry domains, collaboration practises, or even organisation size could impact the generalisability of their findings. Replication is therefore necessary to validate these findings in different settings and to reach more general conclusions [11, p. 1021].

3.5 Related work

3.5.1 Factors Affecting On-Time Delivery in Large-Scale Agile Software Development

”Factors Affecting On-Time Delivery in Large-Scale Agile Software Development” is Dr Kula’s most cited publication and an important predecessor to their research. This study can be described as a deep dive into detecting and understanding the factors that dictate delays in software projects. Taking place at ING, this study followed a mixed-method approach, combining expert insights with data-driven strategies to identify and measure the factors affecting delays in software projects. More precisely, two surveys were conducted with 635 software experts from ING and then, to corroborate the survey answers, their repository data, containing more than 180 teams and 2200 epics, was analysed [25].

The main goal of the surveys was to understand which factors software experts perceive as affecting epic delays and the impact each of them has. Experts identified 25 factors that influence on-time delivery, and these were mapped to five dimensions - organizational, process, project, people and technical. As for the perceived impact of each factor, close to 60 % of the experts believe that all factors moderately impact epic delays. Requirements refinement, task dependencies, organizational alignment, organizational politics and geographic distribution were rated as the most impactful factors [25].

For the second part of the study, 35 proxy measures were derived from the 25 identified factors and their relationship with epic delays was analysed. The results from this section show that 13 proxy measures were able to explain 67% of the delay variation in ING's dataset. The number of sprints, the number of outgoing dependencies, historic performance, the number of years a developer has been at ING, the team's age and the team's size had the most impact on overall delay [25].

3.5.2 The work of Choetkiertikul et al.

Choetkiertikul et al. are some of the biggest contributors in software project delay prediction literature and Kula et al.'s most cited authors. In their 2015 paper, "Threshold-based prediction of schedule overrun in software projects", Choetkiertikul et al. propose one of the first data-driven approaches of identifying risks in software project development [30]. By analysing historical data, Choetkiertikul et al. identified a number of risk factors related to software projects: abnormal resolving time for an issue, abnormal repetitions in the life-cycle of an issue, a developer involved with a number of delayed issues, overloaded developers, and being similar to a large number of delayed issues [30]. They argued that these factors have thresholds which indicate whether or not they are an issue. Finally, they built a Decision Tree model that can predict if a combination of risks and values is indicative of an overall delay [30]. Kula et al. went on to use these factors and this model as a baseline for comparing with their models [11, p. 1018].

In another article from the same year, Choetkiertikul et al. tackled the role of network classification in predicting the delays of software projects [31]. At the time, most risk management relied on generic, high-level guidance or expert judgement [31]. So, naturally, studies started leveraging data-driven approaches to predict or estimate delays in software projects. However, these solutions look at software issues independently, and ignore the network of inter-dependency between these issues. The existence of these networks is the cause of the delay propagation phenomena, where the delay of one software task causes the delay of all the tasks that depend on it, triggering a nasty chain reaction. Choetkiertikul et al. propose a two-stepped solution. First, they contribute a novel technique for the construction of a task network of software development tasks. Second, they present predictive models that leverage both a task's individual factors and its position in the task network to make an accurate estimation of its delay, as well as, simultaneously, the delays of its related tasks [31].

Later on, Choetkiertikul et al. updated their approach to the growing demands of agile development [32]. The first contribution of this paper is the identification of nineteen risk factors. Choetkiertikul et al. also found that developer workload, discussion time, developer's historic performance are the risk factors with the highest importance across multiple projects [32]. To actually predict delays, they developed seven different models, each trained on the nineteen risk factors they collected from seven open-source projects. Their Random Forests model was found to be the most accurate [32], and that is, most likely, the reason for Kula et al. using it as their second baseline model [11, p. 1018].

Choetkiertikul et al. continued studying the agile method of software development and, in 2018, published a paper about predicting delays at the iteration level [33]. This innovative approach centered around the combination of both issue and iteration features as risk factors, as well as the derivation of new iteration features from the features of its issues and their dependencies. Once again, the authors built several predictive models, with Stochastic Gradient Boosting Machines being the most performant this time [33].

3.6 Relevance to PPM

It should now be clear that Kula et al.'s work is of great relevance to PPM. Furthermore, the entire field of software project delay prediction, like the work of Choetkiertikul et al., is also extremely relevant. The ability to accurately predict project delays would allow Portfolio Managers to make timely adjustments to the planning, better manage risks and, crucially, avoid delay propagation. The dynamic nature of Kula et al.'s solution would also be fit for today's rapidly changing market environment. All of this points to a disconnect in literature, where Project Portfolio Management research is unaware of the solutions available in Delay Prediction research, and vice versa.

Even though Kula et al. achieved demonstrably positive results, their solution is bound to the data they had access to at ING. As the authors of the paper recommend, it is necessary to replicate this approach in different settings to reach more general conclusions [11, p. 1021]. This replication would be of value to both PPM literature and to the organisation where the study would take place in.

Fortunately, the companies that collaborated with this thesis, DevOn and the Coca-Cola Hellenic Bottling Company (CCH), provide a suitable environment for such a replication. CCH's Portfolio Management Office has overseen the transition of more than 170 development teams to an agile workflow, all while guaranteeing the delivery of their projects. Both organisations also have domain experts willing to share the necessary insights for a successful adaptation of Kula et al.'s approach.

Therefore, the primary goal of this thesis became the replication of Kula et al.'s method within the context of Coca-Cola Hellenic's Project Portfolio. The rest of this document will answer the research question:

> **RQ2** ”How effective is Kula et al.'s solution when applied to Coca-Cola Hellenic's Project Portfolio?”

The following chapters describe this study's methodology, the results obtained at CCH, and the findings and implications that these entail.

Chapter 4

Approach

This chapter describes the design of the project. The overall goal was to replicate Kula et al.'s approach as closely as possible. Naturally, due to the use of a different set of data and the constraints of a Master's Thesis, some adjustments had to be made.

The chapter is divided in three sections, describing the three development stages of the project: Data Collection and Preprocessing, detecting Delay Patterns, and developing the Bayesian model.

4.1 Step 1 - Data Collection and Preprocessing

4.1.1 Backlog Data

Coca-Cola HBC employs the Scaled Agile Framework and tracks their backlog in two tools, ServiceNow and Azure DevOps. Since the focus of this study is on epics, and in order to collect all the possible information, including individual stories, all of CCH's DevOps data was extracted. This tool tracked 2164 epics, developed by 362 teams, between November 5th 2021 and February 7th 2025.

It is important to note that Kula et al. had access to 7463 delivered epics at ING [11, p. 1015]. This is a significant difference which may impact the rest of the project, particularly when training the Bayesian model.

4.1.2 Data Cleaning

Data cleaning was a long process. The first step was removing non-delivered epics from the dataset. This dropped the original figure down to 1091 delivered epics. The next step, since the goal is to predict delays, was to guarantee all the information is there to measure delays. So, epics that don't have the fields of *Target Start Date*, *Actual Start Date*, *Target Delivery Date* and *Actual Delivery Date* were removed. Fortunately, this had a small impact on the dataset, only eliminating 59 epics. The subsequent measure was making sure each epic had at least one update per milestone, in other words, at least 10 planned sprints. This was one of the filters with the biggest culling, reducing the dataset to 552 epics. Initially, data cleaning ended here, but, later on, some strange results indicated that there was something wrong with the dataset. Upon revising the dataset manually, it was found that a lot of epics had an actual duration of less than 10 sprints. In fact, more than 150 had less than a day of tracked development time. The cause of this issue was very loose progress tracking from CCH. Often times, epics were being dragged from the planning stage to completed in the same motion. Logically, all of these poorly tracked epics were creating noise in the data, and so had to be removed. The dataset was now at 355 epics. The final step, following Kula et al.'s approach, was removing epics that were more than two standard deviations from the mean overall delay. In the end, CCH's dataset was reduced from 2164 epics, to 354, which means a usability rate of 32%.

In a stark comparison, ING ended up with 4040 clean epics and a 54% usability rate (Table 4.1). These numbers are no longer even in the same order of magnitude. This will undoubtedly impact training and most likely negatively affect performance.

Company	Total Epics	Delivered Epics	Clean Epics	Usability Rate
ING	?	7463	4040	54%
CCH	2164	1091	354	32%

Table 4.1: Data cleaning at ING and CCH

4.1.3 Delay Factors

As mentioned in Chapter 3, Section 2, Kula et al. collected 13 predictor variables, identified in previous work, to predict epic delays at ING. The goal at CCH, was to use the same predictor variables, however, that turned out to be impossible. Six of these variables relate to team dynamics and developer data. Since CCH outsources most of their development, it is impossible for them to determine, for example, how many years the developers of an epic have worked for CCH. Thus, all 6 team related variables are unmeasurable at CCH. Moreover, at least in Azure DevOps, CCH doesn't systematically track cross-epic dependencies. Only a handful of epics have explicit dependencies on other epics, so the outgoing dependencies predictor variable was discarded. CCH also doesn't track stories that need to go through a security testing process, so the average story risk was added to compensate for that variable. Lastly, two more predictor variables were added that Kula et al. didn't include, namely number of revisions and total story points.

Table 4.2 contains all the predictor variables and their descriptions, including the ones that couldn't be obtained at CCH and the ones Kula et al. didn't track.

4.1.4 Measuring Schedule Deviation

The last step in creating the CCH dataset was measuring the delay of each epic. In line with Kula et al.'s approach, the *Balanced Relative Error* (BRE) [26] was used to measure the

Delay Factor	Description	ING	CCH
Number of dependencies	Number of outgoing dependencies of an epic on other epics	✓	
Changed Leads	Number of changed tribe leads during the current and previous epic	✓	
Stability Ratio	Median of the ratio of team members that did not change during the current and previous epic	✓	
Developer Age	Median of the number of years the developers working on the epic have been working at the organisation	✓	
Team Existence	Median of the number of years teams have existed in their current composition of team members	✓	
Historic Performance	Median of the ratio of on-time delivered epics over all teams working on the epic	✓	✓
Developer Workload	Median of the number of story points assigned to a developer per sprint	✓	
Unplanned Stories	Number of unplanned stories (related to bug fixes or incidents) that have been added to the epic	✓	
Team Size	Median team size in the epic	✓	
Number of Stories	Total number of planned stories related to the epic	✓	✓
Number of Sprints	Number of sprints assigned to the epic	✓	✓
Number of Incidents	Number of incidents that occurred during the development phase of the epic	✓	✓
Security Level	The ratio of user stories in the epic that need to pass a security testing process	✓	
Average Story Risk	Average story risk of all stories related to the epic		✓
Number of Revisions	Number of revisions the epic has gone through		✓
Total Story Points	Total story points of all stories related to the epic		✓

Table 4.2: Delay Factors and their descriptions

schedule deviation of all epics. BRE is defined as:

$$\text{If Act - Pln } \geq 0\text{, then BRE } = \frac{\text{Act } - \text{ Pln}}{\text{Planned Duration}}$$

$$\text{If Act - Pln } < 0\text{, then BRE } = \frac{\text{Act } - \text{ Pln}}{\text{Actual Duration}}$$

where Act is the actual delivery date and Pln is the planned delivery date. Planned Duration is the difference, in days, between the planned start date and the planned delivery date, and, logically, Actual Duration is the difference between the actual start date and actual delivery date.

A crucial decision has to be made here, and strangely it isn't mentioned in Kula et al.'s article - which start/delivery date should be used? Teams are free to change the delivery date of their epics, and reasonably so. If a considerable workload is added to an epic, it is only sensible to change the epic's delivery date. This way, all the parties involved, be it the team or the PMO, have a better idea of the actual state of the project and can plan accordingly. However, this raises the issue that if teams constantly postpone their delivery dates any time they are delayed, in the end, all epics are delivered on time, as their last delivery dates were met. Thus, a decision has to be made - which delivery date to consider? Based on the belief that a planning error that led to the late addition of workload to an epic should be reflected in that epic's final delay, it was decided that the definitive planned delivery date is the effective delivery date the moment the planning stage is over and the implementation stage starts. While this approach can be considered unforgiving, it is also the most clear, objective, and least prone to error.

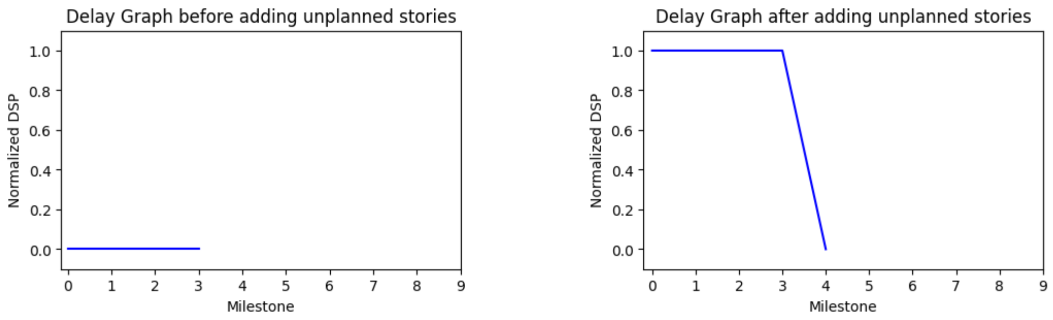
4.2 Step 2 - Detecting Delay Patterns

In order to identify patterns in the delays of epics at CCH, similar epics need to be clustered into discrete groups. Epics will be compared by how their delay evolved over time. Like at ING, this was made possible by tracking each epic's *Delayed Story Points* (DSP) at each milestone. The DSP of an epic at a certain milestone represents the amount of story points delayed to the next milestone. This value is not cumulative. This approach assumes that the story point workload is equally distributed across the entire epic duration. DSP is defined as:

$$\begin{aligned}\text{Expected Story Points} &= \frac{\text{Total Story Points}}{10} \\ \text{DSP} &= \frac{\text{Delivered Story Points}}{\text{Expected Story Points}}\end{aligned}$$

where the Expected Story Points represent the amount of story points expected to be delivered every milestone and Delivered Story Points the actual amount of story points completed at a specific milestone. Each epic's DSP values are normalized by dividing each figure by that epic's highest DSP value. This way, the absolute number of story points doesn't influence the intermediate delay values, and so, epics with very different story point values can still be clustered together.

Something strange happens when stories are added to an epic partway through its development. Since the DSP depends on the expected story points, and these depend on the total story points, if the total story points change throughout the epic, all of the DSP values have to be changed. For example, say a team is working on an epic and the expected story points per milestone is 25. For the first 3 milestones, the team has successfully delivered 25 story points per milestone, and so their DSP for the first 3 milestones is 0.0. Now, suppose



(a) Delay graph before adding unplanned stories

(b) Delay graph after adding unplanned stories

Figure 4.1: Example of impact of unplanned stories on delay graphs

that during milestone 4, due to poor planning, a considerable amount of stories are added to the epic. The total story points change, and now the expected story points per milestone is 30. Because of this change, the DSP values from the first 3 milestones have to be retroactively updated, and since they are normalized, the values are now all 1.0. Suddenly, the graphs of these intermediate delays look completely different, before and after milestone 4 (Figure 4.1). This is a realistic scenario, since most user stories at CCH are created after their respective epic started implementation (Table 4.3). This situation may result in the incorrect clustering of epics in the early stages of their development, and potentially lower the predictive accuracy at that stage.

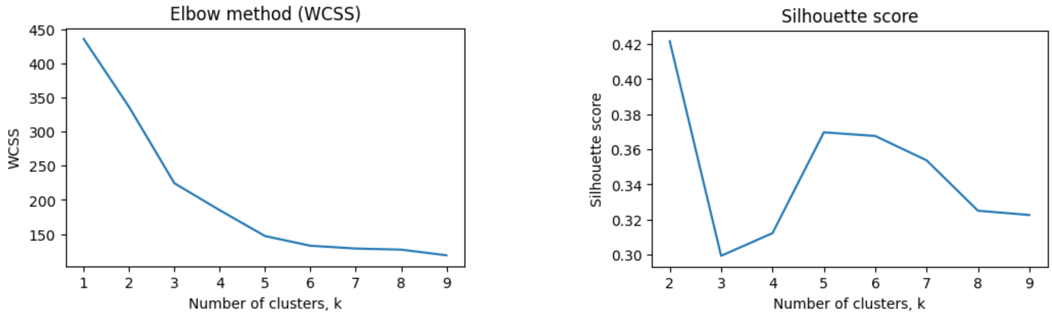
Before Start Date	Between Start and Target Dates	After Target Date
22%	50%	28%

Table 4.3: When user stories are created relative to their respective epic at CCH

The clustering algorithm was the same as Kula et al.'s, hierarchical clustering, as was the distance measure, Dynamic Time Warping (DTW). A difference arose in the selection of the optimal number of clusters, *k*. Kula et al. used the elbow method [11, p. 1015]. This method calculates the total Within Cluster Sum of Squares (WCSS) for each *k* and, by identifying the point of inflection on the curve, determines the optimal value for *k*. This method has been criticised in literature due to its subjective nature. First of all, the choice of an elbow is ambiguous and many times, depending on the data, there are no clear elbows present [34]. Moreover, the range of the parameters, namely the maximum number of clusters, will also affect the location of the perceived elbow [35]. Schubert [35] suggests alternative methods, such as the Calinski–Harabasz index or the Silhouette score. Since TSLearn [1] does not have a built-in Calinski–Harabasz index method, the Silhouette score was chosen as the method to determine the optimal number of clusters.

Figure 4.2a shows the WCSS curve for CCH's data. While there is no clear elbow, there is a reduction in slope around the *k* value of 5. Figure 4.2b shows the silhouette score and it

[1] `tslearn.readthedocs.io`



(a) Elbow method

(b) Silhouette score

Figure 4.2: Cluster number selection methods

is clear that the *k* values of 5 and 6 have the highest scores (ignoring the *k* value of 2), so 5 was selected as the optimal *k* value for CCH's data.

4.3 Step 3 - Bayesian Model Development

For the last step in the approach, it was decided to only develop the dynamic Bayesian model. In Kula et al.'s paper, it is clear that dynamic models consistently outperformed their global and global iterative counterparts in both their Bayesian models and in the state-of-the-art baselines [11, p. 1019]. The goal of this research is to verify the effectiveness of Kula et al.'s solution in a different context, not to compare the performance of dynamic models against global and global iterative models, so those were left out.

Kula et al. noted that the BRE values at ING are proportional numbers between 0 and 1, with a high density of zeros (42%). When it came to selecting a likelihood function, Kula et al. picked a function that best described the BRE distribution they were working with, namely a Zero-inflated Beta function [11, p. 1016].

It is interesting to reflect on the meaning of ING's BRE values. As could be seen before in the BRE definition, the BRE assumes a positive value when an epic is delayed and a negative value when it is delivered in advance. So, when Kula et al. state that the minimum BRE value at ING is 0, it means that ING never delivered an epic before its planned deadline. Not only that, but since 42% of the BRE values are 0, almost half of ING's epics are delivered exactly on time. There are a few possible explanations for this. First of all, ING could simply be extremely strict (and successful) in their planning and execution. Another option is ING delaying epics that are ahead of schedule. The final reason could be related to the interpretation of planned delivery dates. As alluded to before, planned delivery dates can be updated, and the choice of which one to use could severely alter BRE values. Since Kula et al. do not mention this problem, it is impossible to know what impact their decision had on ING's BRE distribution.

CCH's BRE distribution, on the other hand, is completely different. Looking at Figure 4.3, it is clear that the distribution range is much wider, -1.58 to 3.99. To clarify, a BRE value of 4.0 means an epic took four times longer than planned to complete. Additionally,

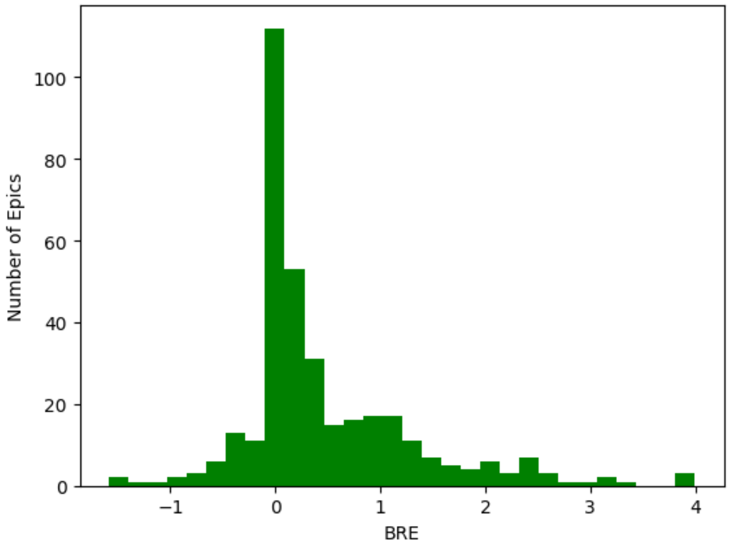
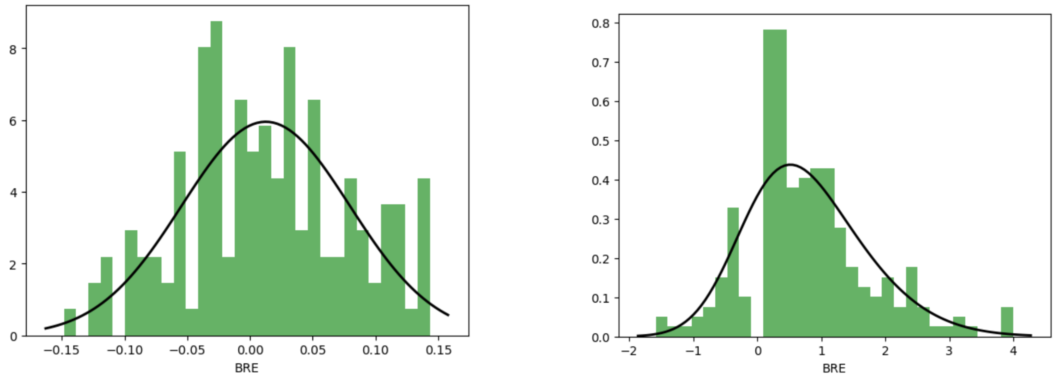


Figure 4.3: CCH's BRE distribution

even though there is a high density around 0, none of those values actually equals 0. This distribution is challenging to fit because there is no function that clearly describes it. To address this issue, multiple versions of the dynamic Bayesian model were trained, each using a different likelihood function, to identify the best-performing function.

Three models were considered: a normal model, the simplest and least computationally intensive, but also inappropriate given the shape of the distribution; a skewed normal model, slightly more complex but also closer to the real distribution; and a mixed model, a combination of a normal and skewed normal functions for different portions of the data. The idea behind this last model came from an experiment that divided the BRE distribution in two parts. By segregating the highest density area from the rest of the data, two more proportional distributions emerge: a normal distribution for BRE values between -0.15 and 0.15, and a skewed normal distribution for the remaining values (Figure 4.4). Even though the mixed model describes the data the best, it is by far the most computationally intensive, and its success depends on the Bayesian model's ability to learn to divide the dataset correctly. Given the small size of the dataset, this approach was unlikely to succeed. The results support this: the skewed normal and mixed models had $\hat{R}$ values consistently above 1.5, indicating poor chain mixing and unreliable predictions. In contrast, the normal model consistently had $\hat{R}$ values below 1.01, likely due to its simplicity. As a result, the skewed normal and mixed models were discarded, and all future references will refer to the normal model. The model's definition is provided in Equations 4.1-4.4.



(a) BRE distribution from -0.15 to 0.15

(b) BRE distribution excluding -0.15 to 0.15

Figure 4.4: Mixed BRE distributions

$$
\begin{align}
\text{BRE}_i &\sim \mathit{Normal}(\alpha + \mathbf{X}_i \cdot \beta, \sigma) \tag{4.1}\\
\alpha &\sim \mathit{Normal}(0,1) \tag{4.2}\\
\beta_1, \ldots, \beta_{10} &\sim \mathit{Normal}(0,1) \tag{4.3}\\
\sigma &\sim \text{Cauchy}(0,1) \tag{4.4}
\end{align}
$$

Chapter 5

Results

This chapter presents all the results gathered at Coca-Cola Hellenic. The first section concerns the identified delay patterns and their characteristics. The second contains the results of the Bayesian model. The final section will answer the second research question.

5.1 Delay Patterns at CCH

Using the Elbow method and the silhouette score, $k = 5$ was determined as the optimal number of clusters (Figures 4.2a and 4.2b). Thus, 5 delay patterns were identified at CCH. Figure 5.1 presents these patterns, including the centroids and the area between the 25th and 75th percentiles. The epic distribution and mean variance values indicate that these clusters are balanced and the patterns recurrent (Table 5.1).

Cluster	Epic Percentage	Mean Variance
1	5.38	0.00
2	19.54	0.02
3	14.75	0.03
4	24.92	0.06
5	35.41	0.03

Table 5.1: Epic distribution and mean variance per cluster at CCH

The patterns at Coca-Cola Hellenic have interesting shapes. First of all, they are quite polarised - most of them either start or end at 0.0 or 1.0. They also appear to mirror each other. Cluster 2's pattern shows epics that tend to be more delayed as time goes on, while the pattern from Cluster 5 shows epics being less delayed as time goes on. The same mirroring can be seen in Clusters 3 and 4. Cluster 3's centroid looks like a downward parabola, representing epics in which delays happen in the middle of development time, while Cluster 4 is sort of an upward parabola, where delays tend to happen at the start and at the end of the project. Cluster 1 is clearly the odd one out, just a straight line at 0.0. Being the cluster with the smallest percentage of epics might indicate that these have some peculiar characteristics. This will be discussed later.

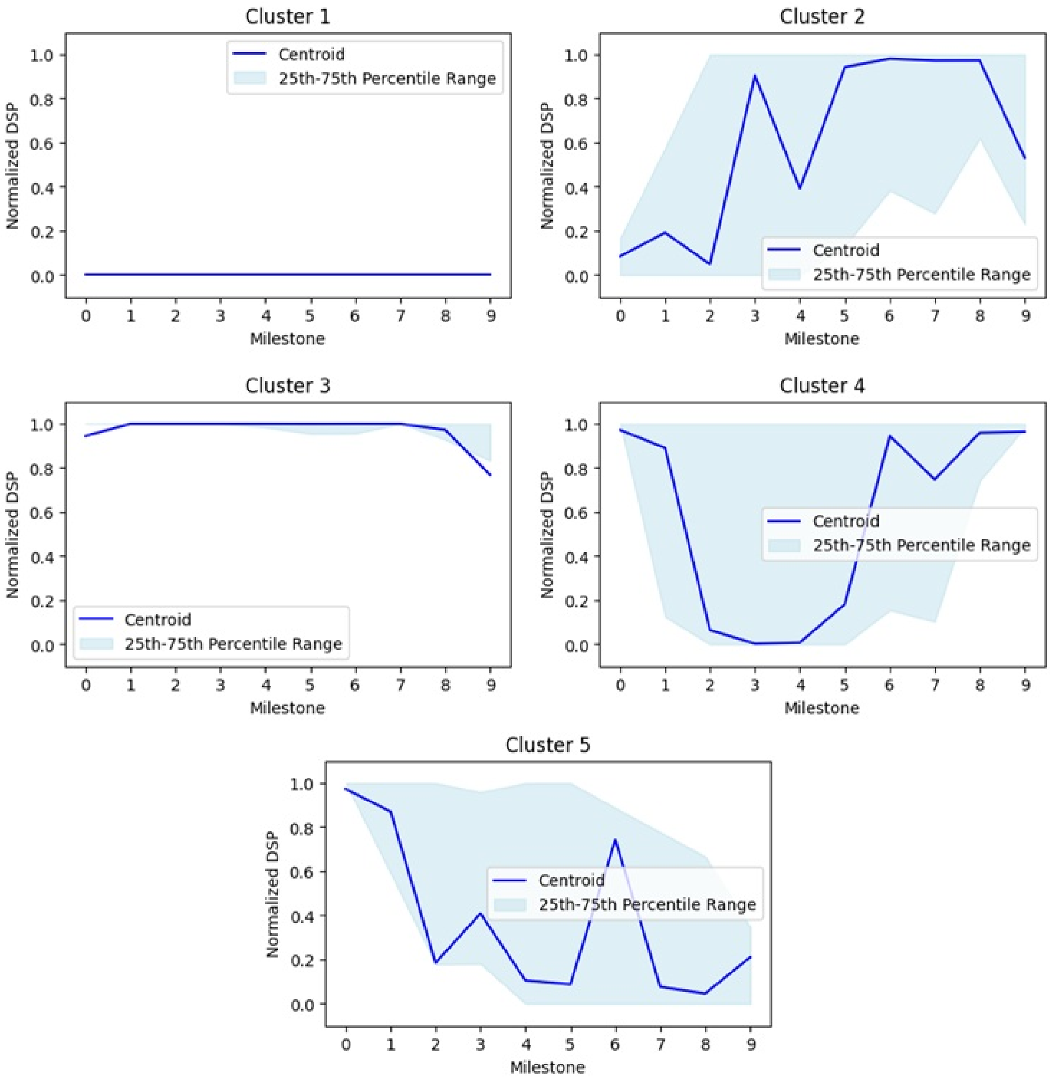


Figure 5.1: Delay patterns at CCH

The CCH patterns look very different from the ING patterns [11, p. 1017]. Logically, the centroids are quite different, and that is expected, as different ways of working lead to different types of delays. The significant difference is in the distance between the 25th and 75th percentiles. At ING, the 25th and 75th percentiles closely follow the centroid, and that highlights the existing pattern. As can clearly be seen in Figure 5.1, the 25th and 75th percentiles often border opposite extremes of the graph. There are two possible explanations for this. It could be due to the small dataset, as there are not enough epics to really carve out and highlight the underlying pattern. Or, simply, it could just be that the patterns do not exist, or at least not as clearly as at ING.

Table 5.2 contains the statistical details of the CCH clusters. For each cluster, the median values for each predictor variable are presented, and it is indicated if those values are significantly different from all other clusters. Wilcoxon tests were used for the pairwise comparisons. The significance columns from Table 5.2 clearly indicate that the epics in each cluster are very similar to each other. The only significant differences are the low values of number of stories and total story points for cluster 1, which explains its oddity, and the lower amount of planned sprints for cluster 3.

Predictor	Median					Significance				
	C1	C2	C3	C4	C5	C1	C2	C3	C4	C5
nr-incidents	0	0	0	0	0					
nr-stories	1	40	57	21.5	52	*				
nr-revs	21	17	17	16	15					
total-story-points	0	110	146	89	183	*				
average-risk	3	3	3	3	3					
nr-sprints	22	27	18	25	27			*		
hist-performance	0.17	0.25	0.15	0.18	0.2					
BRE	0.19	0.05	1.13	0.05	0.19			*		

Table 5.2: Characteristics of delay clusters (* indicates a significant difference)

The bottom row of Table 5.2 shows the statistics of overall delay, measured through the BRE, for each cluster. Cluster 3 is the only cluster with a significantly different BRE value. This indicates that delay patterns are not indicative of overall delay.

This directly opposes one of Kula et al.'s main findings. At ING, all clusters had significantly different BRE values, which indicates that patterns are indicative of overall delay. The same two explanations emerge, either the dataset is too small to highlight the existing patterns, or the patterns are not there to begin with. This difference may also have implications for the Bayesian model, as the delay pattern predictor variable will likely be much less impactful, undermining performance.

5.2 Bayesian Model Results

5.2.1 Experimental Setup and Performance Measures

In order to mimic a real-world scenario, where observed epics are used to make predictions for future epics, the epics and their milestones were ordered based on their start date. Following Kula et al.'s approach for training and evaluation, time-based 10-fold cross-validation was used. This time-based method of cross-validation guarantees that epics in the first k folds (training set) started before the epics in fold k+1 (test set). This way, subsequent training sets superset the previous ones. This insures a sequential updating of the models based on past knowledge [11, p. 1018].

As for the performance measures, *Mean Absolute Error* (MAE) and *Standard Accuracy* (SA) were used. The first can be defined as:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |Actual\ BRE_i - Estimated\ BRE_i| \qquad (5.1)$$

where N is the number of epics used for testing, *Actual BRE*$_i$ is the actual delay measured in BRE, and *Estimated BRE*$_i$ is the predicted BRE value, for an epic i. This is the simplest and most understandable performance measure. SA is based on MAE and it has a more complex interpretation. It is defined as:

$$SA = \left(1 - \frac{MAE}{MAE_{rg}}\right) \times 100 \tag{5.2}$$

where MAE is defined as the MAE of the model that is being tested and MAE_{rg} is the MAE of a large number of random guesses. SA measures how much better the model performs versus random guessing. Logically, this random guessing is very important and it needs to be defined. The same method as Kula et al. was used, the unbiased exact calculation of MAE_{rg} proposed by Langdon et al. [36]. Simply put, this definition computes the mean absolute error of selecting a prediction from the distribution of actual values. More formally:

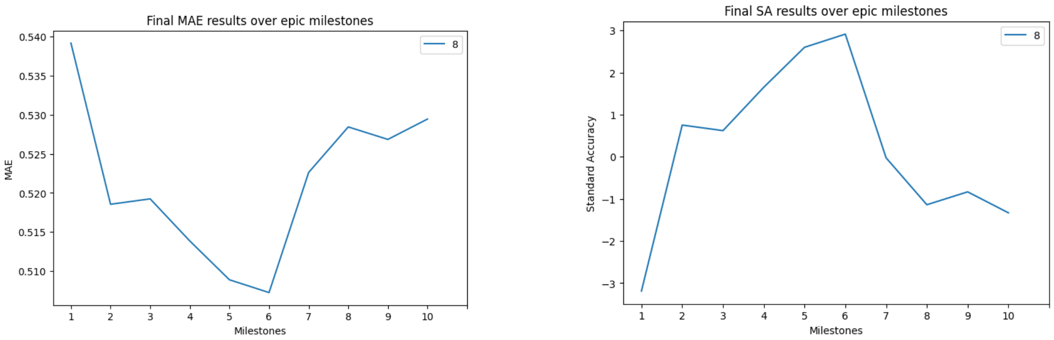
$$MAE_{rg} = \frac{2}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{i} |y_i - y_j| \tag{5.3}$$

where N is the number of samples in the distribution, and y_i and y_j are the actual BRE values for epics i and j. Finally, the lower the MAE and the higher the SA, the better the performance.

5.2.2 Results

The first result under analysis is the performance of the Bayesian model trained on 90% of the dataset. This was the model with the access to the most training data, and so, it is expected for it to be the model with the best performance. Figure 5.2 shows the MAE and SA results of the Bayesian model trained on 9 folds. Looking at the MAE results first, it is clear that there is not a lot of variation. The values range from around 0.510 to 0.540. In absolute terms, this means that the model is usually off by half a BRE unit. In other words, it is wrong by half of the epic's planned or actual duration. So, for an epic that was planned to last a year, and it actually did, the model would predict, on average, that it would take a year and a half to complete. Looking at CCH's BRE distribution, Figure 4.3, it could be argued that a 0.5 margin of error is still useful, as it would still identify epics with abnormally large BRE values (BRE $>$ 1.0). Nevertheless, a MAE of 0.5 is a poor result. This is confirmed by the SA results. Here the values range from -3% to 3%, which indicates that the model is practically as performant as a random guess.

It is also interesting to analyse the evolution of the models as they are trained with more and more data. With time-based 10-fold cross-validation, one model is trained on one fold, then another model is trained on two folds, so on and so forth. This process may indicate the models dependence on data size. It is expected that as the amount of training data increases, so should the performance. Figure 5.3a shows exactly that. As can be seen by the regression line, as the number of folds increases, i.e. the amount of training data, the lower the MAE value is, in other words, the better the model performs. This is a strong indicator that if



(a) Mean Absolute Error over time

(b) Standard Accuracy over time

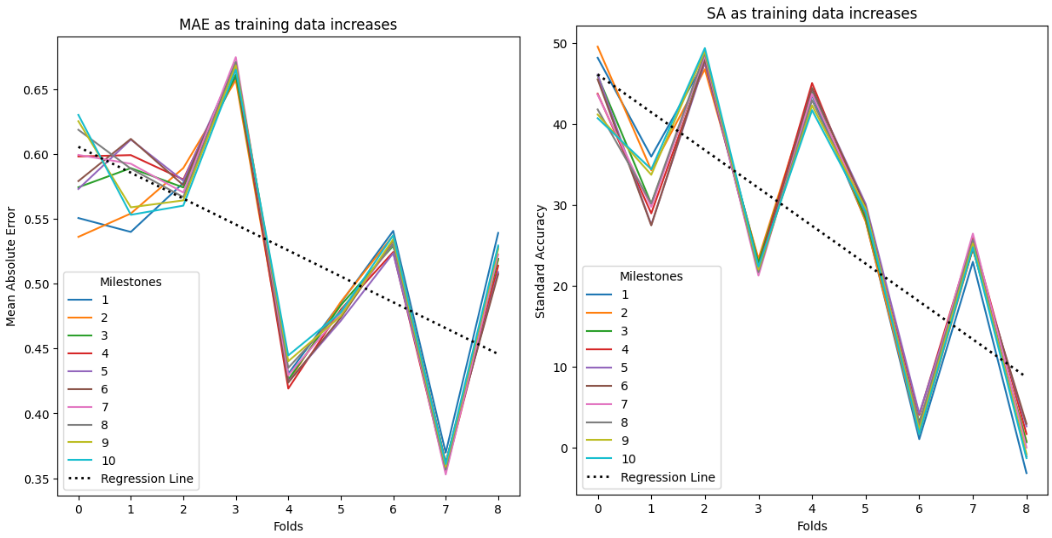
Figure 5.2: MAE and SA results of Bayesian model trained on 9 folds

CCH had more clean data, the models would perform better. This argument is undermined, however, by looking at the SA results. Figure 5.3b shows that the standard accuracy of the models does not increase with data size, it in fact decreases. How come one performance measure improves while the other deteriorates? It has to do with the random guessing used in the SA calculation. Langdon et al.'s [36] approach of computing the MAE of a large number of random guesses uses the average error of taking any particular value from the real distribution as a prediction for any other particular value (Equation 5.3). What this means for a distribution like CCH's BRE values (Figure 4.3) is that since the distribution has a high density around 0, most guesses will be taking a value around 0 to predict another value around 0. As the number of folds increase, the density around 0 becomes larger, and thus the MAE of random guesses becomes lower. That is why the SA of the models decreases over the number of training folds, even though the MAE is also decreasing - the MAE of random guesses is decreasing faster (Equation 5.2).

Finally, it is also expected for the models to perform better at later milestones. The closer an epic is to the target date, the easier it should be to predict its delay. Unfortunately, all of the models fail at this improvement. As can be seen in Figure 5.4, all of the models have flatlines, that is, their performance in milestone 1 is relatively the same as in milestone 10. This result highlights the inadequacy of the models as predictive tools. It is unclear, however, if the reason for that is a lack of clean data, or the theory itself.

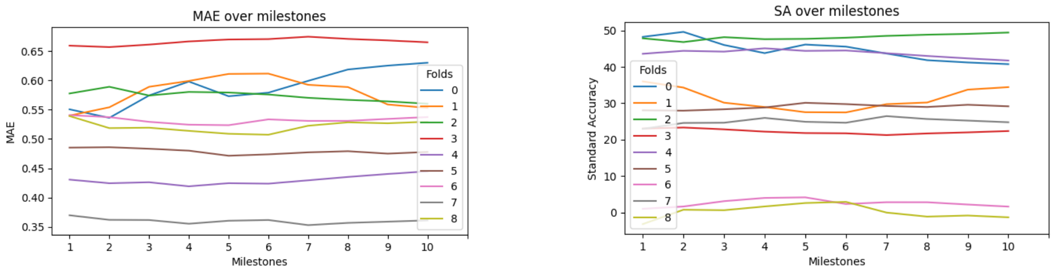
5.3 How effective is Kula et al.'s solution when applied to Coca-Cola Hellenic's Project Portfolio?

In an objective sense, the answer to this research question is simple. The Mean Absolute Error of the final model was around 0.5. This means the model is off, on average, by half a BRE unit, and that is its objective efficacy. However, from an interpretative perspective, the answer is more nuanced. If one were to focus solely on the Standard Accuracy results of the final model, they would argue that the solution is not effective at all, as it is as good as a random guess. That would be short-sighted, however, as it fails to consider the usefulness



(a) MAE results over the number of training folds (b) SA results over the number of training folds

Figure 5.3: MAE and SA results over the number of training folds



(a) Mean Absolute Error of all models over time (b) Standard Accuracy of all models over time

Figure 5.4: MAE and SA results of all Bayesian models

of a MAE of 0.5. In conditions such as CCH's (Figure 4.3), while a margin of error of 0.5 will not be useful most of the time, it is still very effective at detecting outliers, i.e. epics with abnormally large delays. In other words, despite the fine-grained predictive efficacy being low, this solution is able to identify epics that stray severely off the path, which at CCH is a not so rare occurrence.

Another interesting point to consider is that the performance of the model is improving as it is given access to more data, according to the Mean Absolute Error (Figure 5.3a). This suggests that as CCH produce more and cleaner data, this solution would become increasingly effective. As alluded to before, this argument is challenged by the fact that the Standard Accuracy of the model actually goes down as data access increases (Figure 5.3b). This goes back to the definition of Standard Accuracy, and it happens because the random guessing is also improving, even faster than the model. It is expected for the model to continue to improve as it learns on more data, but the random guessing should, at a certain point, plateau. So, given more data, it could be that the model performs better both in MAE and SA. This is, however, just an hypothesis. The concrete results on the evolution of the model as it is given access to more data are thus inconclusive.

Finally, it is important to consider the evolution of performance over the milestones. One of the simplest expectations would be that the model performs better at later milestones, rather than at the start of an epic. It should be much easier to predict the delay of an epic one sprint before its deadline than at the start of development. Figure 5.4 shows that there is no improvement in performance over the milestones. This result strongly indicates that the model did not fit the data, and thus its predictions cannot be considered reliable. Overall, this result cements that this solution was not effective at predicting delays in Coca-Cola Hellenic's project portfolio.

Chapter 6

Discussion

This chapter examines the results from the previous chapter, interprets them holistically, and explores their broader implications. Furthermore, the limitations of this study are discussed, as well as suggestions for future research.

6.1 What factors justify the differences found?

The results obtained at Coca-Cola Hellenic are drastically different from the ones at ING. While the Mean Absolute Error of the CCH model was always closely around 0.5, the MAE from Kula et al.'s original solution started at 0.19 and went as low as 0.04 [11, p. 1019]. Furthermore, the Standard Accuracy at CCH was near 0%, while at ING it started at 66% and reached 92% at the last milestone [11, p. 1019]. These disparate results reveal two different outcomes. While at ING this solution was able to achieve great results, at CCH it was unfit and proved no better than random guessing. The question then remains, what justifies these differences?

There are two types of factors that contributed to this, methodological and contextual. Starting with the first, while the intent was always to follow Kula et al.'s approach as closely as possible, the differences in the underlying data imposed some changes in methodology. Data quality had a very high impact here. First off, there is the difference in the initial dataset. Kula et al. had access to 7463 epics, while CCH's backlog only tracked 2164 epics. This contrast was exacerbated after data cleaning, where CCH was left with just 354 epics, while ING remained with more than 4000. More problems arose with the delay factors. Out of the 13 factors Kula et al. found to be explicative of delays, only 5 were available at CCH. In the end, the CCH dataset was 10 times smaller, and less than half as rich. This, of course, put a lot of strain on training the Bayesian model, and it might have been the biggest influence on predictive performance.

The methodological differences do not end there, however. The delay distributions of ING and CCH are also very different, both in shape and, especially, range. The choice of target dates may have created this disparity. As mentioned before, teams can change the target dates of epics they own, and the decision of which target date to consider for delay measurement will severely change the company's delay distribution. Perhaps, if the last target date had been used for delay measurement at CCH, the delay distribution would be a lot more similar to the ING one. Since Kula et al. do not mention this problem, it is impossible to know its full impact. Nevertheless, CCH's delay distribution is different from ING's, and so it required a different Bayesian model. The CCH model used a Normal likelihood function, unlike ING's Zero-inflated Beta function. These asymmetries in likelihood function, delay distribution, and, especially, dataset size and quality serve as an explanation to the differences in results and performance.

While these methodological factors adequately explain differences in performance, they are not the root cause of the issue. Contextual factors provide a broader perspective into these two studies and help clarify why the same solution performed differently in practice. Firstly, CCH and ING function in completely different industries. ING is a bank, a highly regulated firm where projects must follow strict compliance requirements and assessments. This controlled environment demands structure and it can lead to more predictable outcomes. On the other hand, CCH is a soft drink producer and distributor. It operates in a less strict, faster-moving market which may require rapid adjustments, making delays less predictable. These industry factors likely contributed to the differences in data quality and delay distributions, and, ultimately, the predictive performance of solution.

Another contextual factor lies in the differences between outsourcing and insourcing. ING develops its main software applications in-house. This allows them to have total control over the development process and to make changes to their data architecture more easily. On the contrary, CCH outsources most of their development work. Because of this, they are unable to track the same features as ING, reducing the richness of their data. The variability in ways of working may also weaken the correlation between delay factors and actual delays.

Finally, agile maturity may have also played a role. ING became an agile organization in 2015, while CCH only started its agile journey in 2021. This solution is meant for agile corporations and it has a lot of dependencies on agile artifacts. Logically, having a 6 year head start, ING has had more time to collect agile artifacts and, more importantly, cultivate standards of quality. It is unfeasible for an organisation to perfectly implement agile overnight, it is a slow, continuous process, which means that CCH's earlier epics often fall short of this solution's data requirements. This is reflected in the difference in dataset sizes, before and after cleaning.

6.2 How useful is this solution for Coca-Cola Hellenic?

With its current efficacy, this solution is only useful for CCH as an outlier detector. A Mean Absolute Error of 0.5 does not allow portfolio managers to make fine-grained decisions around risk management, but it does serve to identify epics with abnormally large delays. This is not, however, the intended purpose of this solution, and so there are surely other tools that would more effectively detect outliers.

Currently, the most useful option for CCH would be to lay the necessary groundwork for this solution. The simplest, and most direct way to achieve this is to start collecting the 13 delay factors Kula et al. mention and to maximize the number of epics making it through the data cleaning process. As previously mentioned, more than 150 epics in the CCH dataset had less than a day of tracked development time. Stricter progress tracking would not only significantly increase the size of the training dataset, but also shorten the gap between what is real and what is represented in the data, making any results more valuable. If, however, CCH wanted to be even more methodical, the way to achieve the best results from this solution would be to start by replicating Kula et al.'s 2022 paper, "Factors Affecting On-Time Delivery in Large-Scale Agile Software Development" [25]. This is the study where Kula et al. identified the 13 delay factors of most importance at ING. By following the same approach, CCH would guarantee to collect the most significant delay factors for them, as they might differ from ING.

Last, but not least, it is paramount to understand Coca-Cola Hellenic's needs and wants. Does CCH want this solution? How important is predictability to them? As previously mentioned, the differences in industry have a great impact on predictability. The consumer goods industry will always be more volatile, and thus less predictable than banking. For this reason, perhaps, predictability may not be such a high priority for CCH as for ING. Moreover, the desire to be more predictable came from the engineers and developers at ING. The people who wanted to know are the people who created the data. This made enacting the necessary changes much easier. Given CCH's organisational context, namely the use of outsourcing, they may face a lot more inertia. An organic, bottom-up change will always be swifter than a top-down enforcement, especially given the constraints of outsourcing.

6.3 Limitations

While this study provides insights into AI-powered delay prediction and to the generalisability of Kula et al.'s work, it is not without its limitations. These indicate the boundaries of the findings, and should be an important consideration when interpreting the results.

The first, and most basic limitation of this thesis is the misalignment of data and reality. It must be clear that all the data in this study is just a representation of reality. The verisimilitude of that representation is undefined. Throughout development, there were some indications of this disparity between data and reality, like more than 150 epics having less than a day of tracked development time. Ultimately, the model may be trained on misleading inputs, which reduces the reliability of its predictions.

Adding to the misalignment of data and reality is the concept of milestones. While their intended purpose is understandable, it is a lot easier to compare epics if their datapoints align perfectly, the use of milestones is increasing the artificiality of the data. There is, fundamentally, no need to add milestones, as epics are already broken down into sprints and these are the most natural points to collect project data. The use of milestones also discards useful data. For example, by choosing to use 10 milestones, an epic with 20 sprints is only represented by half of its datapoints. Overall, the concept of milestones imposes structure at the cost of realism.

Another limitation relates to the robustness of the results. After cleaning, the CCH dataset was reduced to 354 epics. Furthermore, these had a relatively small amount of datapoints, 7 variables each. It is very difficult to extract consistent results from this small a dataset, which explains the variability that can be seen in the Results chapter. This lack of robustness may put the validity of the results into question, and introduces a lot of uncertainty when interpreting them.

Finally, the limited generalisability of this thesis must be highlighted. The findings of this study are specific to Coca-Cola Hellenic and its organisational context. This data may not be representative of software projects in other organisations. So, further replication is needed to continue to generalise these findings.

6.4 Recommendations for Future Work

For starters, and continuing the last remark, further replication of Kula et al.'s work is necessary. One example where their solution proved unsuccessful is not enough to discredit the generalisability of their work, especially given the limitations of this study. So, it would be interesting to replicate Kula et al.'s solution again, in different settings, but with more data to analyse.

Secondly, organisations could benefit from a requirement framework for a successful implementation of this solution. This framework would detail the prerequisites a company must fulfil for the delay prediction model to be effective. Its requirements could be the collection of Kula et al.'s 13 delay factors, or any number of delay factors relevant to each organisation, as well as a minimum recommended number of clean epics. A framework like this would act as a diagnostic tool, allowing organisations such as CCH to evaluate their readiness and even highlight the areas where improvement or investment is needed before adoption.

Lastly, it would be interesting to investigate the effects of two changes in the methodology of this solution. The first is the choice of target date. As previously mentioned, teams can change the target dates of epics they own, and the decision of which target date to consider for delay measurement will severely change the organisation's delay distribution. A more systematic exploration of how this choice influences the results, as well as how useful the alternatives are, would add substantial value to this solution. The other methodological change would be discarding milestones. As argued before, milestones add structure at the cost of realism. It would be interesting to investigate the difference in results, as well as the perceived impact on the people using this solution.

Chapter 7

Conclusions

This study set out to investigate the effectiveness of Kula et al.'s AI-powered delay prediction model when applied to Coca-Cola Hellenic's Project Portfolio. The results clearly show that the solution, which was successful at ING, had very limited, if any, success at CCH. While ING was able to achieve a strong predictive performance, with a Mean Absolute Error as low as 0.04 and a Standard Accuracy of upwards of 90%, CCH's model performed poorly, with a Mean Absolute Error hovering around 0.5 and a Standard Accuracy near 0%, indicating that the solution is no better than random guessing. This disparity has revealed the limitations of directly transferring an AI solution to a different organisation, namely the dependency on quality data.

Differences in dataset size and quality were identified as the primary justification for lack of performance. While ING's model was trained on over 4,000 projects, CCH was left with just 354 epics after cleaning, and fewer than half of the originally identified 13 delay factors were available. In the end, instead of a rich dataset like ING's, CCH's data was sparse and incomplete. Methodological differences further amplified this problem, including disparities in the shape and range of delay distributions, the choice of likelihood functions, and the handling of target dates.

On the other hand, contextual factors are likely a major contribution to the difference in results. CCH and ING operate in fundamentally distinct industries, with differences organisational structures, processes, and regulatory constraints. These contextual factors influenced not only the availability and quality of the data, but also the predictability of delays themselves.

These findings are valuable to anyone attempting to replicate this solution, or to organisations aiming to adopt AI-powered analytics. They highlight the importance of assessing both data readiness and contextual fit before implementation. As a whole, this study underscores the need of thorough evaluation and preparation when applying AI solutions.

Looking forward, a number for avenues of future work are proposed. For CCH to unlock the full potential of this solution, necessary groundwork has to be laid. The two main steps here are the collection of more delay factors and the improvement of data tracking. CCH, and other organisations in similar positions, could benefit from a requirement framework for AI solutions to use as a diagnostic tool, evaluating their readiness and highlighting areas of improvement. Finally, more replication efforts are needed to continue to generalise the work of Kula et al., and the exploration of the impact of two methodological changes, namely the handling of target dates and the removal of milestones, could reveal significant insights.

To conclude, this thesis offers a broader lesson - AI solutions like Kula et al.'s are not easily transferable. Their success depends heavily on data quality, organisational context, and application environment. Understanding and addressing these dependencies is crucial for any organisation aiming to adopt AI-powered analytics.

Bibliography

[1] Harry Markowitz. Portfolio Selection. Technical Report 1, 1952.

[2] Robert G. Cooper, Scott J. Edgett, and Elko J. Kleinschmidt. Portfolio management in new product development: Lessons from the leadersr - I, 1997.

[3] Robert G Cooper, Scott J Edgett, and Elko J Kleinschmidt. New Product Portfolio Management: Practices and Performance. Technical report, 1999.

[4] Robert G. Cooper, Scott J. Edgett, and Elko J. Kleinschmidt. New problems, new solutions: making portfolio management more effective. *Research Technology Management*, 43(2):18–33, 2000.

[5] Robert Cooper, Scott Edgett, and Elko Kleinschmidt. Portfolio management for new product development: Results of an industry practices study. *R and D Management*, 31(4):361–380, 2001.

[6] Juliane Teller, Barbara Natalie Unger, Alexander Kock, and Hans Georg Gemünden. Formalization of project portfolio management: The moderating role of project portfolio complexity. *International Journal of Project Management*, 30(5):596–607, 7 2012.

[7] R. Chadwick Holmes and Hemant Kumar. Defining a Flexible Value Framework for Digital Products and Services Using Systems Engineering and AI Approaches. 11 2023.

[8] Rajeshwar Vayyavur. *An Intelligent Project Portfolio Management System Approach to Enhance Project Maturity, Performance and Success Rate* . PhD thesis, California Intercontinental University, 12 2016.

[9] Bingbing Zhang, Libiao Bai, Kaimin Zhang, Shuyun Kang, and Xinyu Zhou. Dynamic assessment of project portfolio risks from the life cycle perspective. *Computers & Industrial Engineering*, 176:108922, 2 2023.

[10] Libiao Bai, Kaimin Zhang, Huijing Shi, Min An, and Xiao Han. Project Portfolio Resource Risk Assessment considering Project Interdependency by the Fuzzy Bayesian Network. *Complexity*, 2020:1–21, 11 2020.

[11] Elvan Kula, Eric Greuter, Arie van Deursen, and Georgios Gousios. Dynamic Prediction of Delays in Software Projects using Delay Patterns and Bayesian Modeling. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1012–1023, New York, NY, USA, 11 2023. ACM.

[12] Monther Tarawneh, Huda AbdAlwahed, and Faisal AlZyoud. Innovating Project Management: AI Applications for Success Prediction and Resource Optimization. pages 382–391, 2024.

[13] Thomas Kremmel, Jiří Kubalík, and Stefan Biffl. Software project portfolio optimization with advanced multiobjective evolutionary algorithms. *Applied Soft Computing*, 11(1):1416–1426, 1 2011.

[14] Khalifa Mohammed Al-Sobai, Shaligram Pokharel, and Galal M. Abdella. A Framework for Prioritization and Selection of Strategic Projects. *IEEE Transactions on Engineering Management*, 71:2310–2323, 2024.

[15] Roy F Baumeister and Mark R Leary. Writing Narrative Literature Reviews. Technical Report 3, 1997.

[16] Barbara Kitchenham. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical report, 2007.

[17] Suzanne K. Linder, Geetanjali R. Kamath, Gregory F. Pratt, Smita S. Saraykar, and Robert J. Volk. Citation searches are more sensitive than keyword searches to identify studies using specific measurement instruments. *Journal of Clinical Epidemiology*, 68(4):412–417, 4 2015.

[18] Mahboubeh Farid, Mikael Palmblad, Hampus Hallman, and Johannes Vänngård. A binary decision tree approach for pharmaceutical project portfolio management. *Decision Analytics Journal*, 7:100228, 6 2023.

[19] Farshad Faezy Razi and Seyed Hooman Shariat. A hybrid grey based artificial neural network and C&amp;R tree for project portfolio selection. *Benchmarking: An International Journal*, 24(3):651–665, 4 2017.

[20] Dimitrios C. Tselios, Ilias K. Savvas, and M Tahar Kechadi. Multiple project portfolio scheduling using recurrent neural networks. *International Journal of Simulation and Process Modelling*, 8(4):227, 2013.

[21] Yuanyuan Tian, Libiao Bai, Lan Wei, Kanyin Zheng, and Xinyu Zhou. Modeling for project portfolio benefit prediction via a GA-BP neural network. *Technological Forecasting and Social Change*, 183:121939, 10 2022.

[22] Fatma Demircan Keskin. A two-stage fuzzy approach for Industry 4.0 project portfolio selection within criteria and project interdependencies context. *Journal of Multi-Criteria Decision Analysis*, 27(1-2):65–83, 1 2020.

[23] Anass Zaidouni, Mohammed Abdou Janati Idrissi, and Adil Bellabdaoui. A Sugeno ANFIS Model Based on Fuzzy Factor Analysis for IS/IT Project Portfolio Risk Prediction. *Journal of Information and Communication Technology*, 23(2):139–176, 4 2024.

[24] M. Kandakoglu, G. Walther, and S. Ben Amor. The use of multi-criteria decision-making methods in project portfolio selection: a literature review and future research directions. *Annals of Operations Research*, 332(1-3):807–830, 1 2024.

[25] Elvan Kula, Eric Greuter, Arie van Deursen, and Georgios Gousios. Factors Affecting On-Time Delivery in Large-Scale Agile Software Development. *IEEE Transactions on Software Engineering*, 48(9):3573–3592, 9 2022.

[26] Y. Miyazaki, M. Terakado, K. Ozaki, and H. Nozaki. Robust regression for developing software estimation models. *Journal of Systems and Software*, 27(1):3–16, 10 1994.

[27] T. Foss, E. Stensrud, B. Kitchenham, and I. Myrtveit. A simulation study of the model evaluation criterion mmre. *IEEE Transactions on Software Engineering*, 29(11):985–995, 11 2003.

[28] B.A. Kitchenham, L.M. Pickard, S.G. MacDonell, and M.J. Shepperd. What accuracy statistics really measure. *IEE Proceedings - Software*, 148(3):81, 2001.

[29] Dan Port and Marcel Korte. Comparative studies of the model evaluation criterions mmre and pred in software cost estimation research. In *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*, pages 51–60, New York, NY, USA, 10 2008. ACM.

[30] Morakot Choetkiertikul, Hoa Khanh Dam, and Aditya Ghose. Threshold-based prediction of schedule overrun in software projects. In *Proceedings of the ASWEC 2015 24th Australasian Software Engineering Conference*, pages 81–85, New York, NY, USA, 9 2015. ACM.

[31] Morakot Choetkiertikul, Hoa Khanh Dam, Truyen Tran, and Aditya Ghose. Predicting Delays in Software Projects Using Networked Classification (T). In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 353–364. IEEE, 11 2015.

[32] Morakot Choetkiertikul, Hoa Khanh Dam, Truyen Tran, and Aditya Ghose. Predicting the delay of issues with due dates in software projects. *Empirical Software Engineering*, 22(3):1223–1263, 6 2017.

[33] Morakot Choetkiertikul, Hoa Khanh Dam, Truyen Tran, Aditya Ghose, and John Grundy. Predicting Delivery Capability in Iterative Software Development. *IEEE Transactions on Software Engineering*, 44(6):551–573, 6 2018.

[34] DAVID J. KETCHEN Jr. and CHRISTOPHER L. SHOOK. THE APPLICATION OF CLUSTER ANALYSIS IN STRATEGIC MANAGEMENT RESEARCH: AN ANALYSIS AND CRITIQUE. *Strategic Management Journal*, 17(6):441–458, 6 1996.

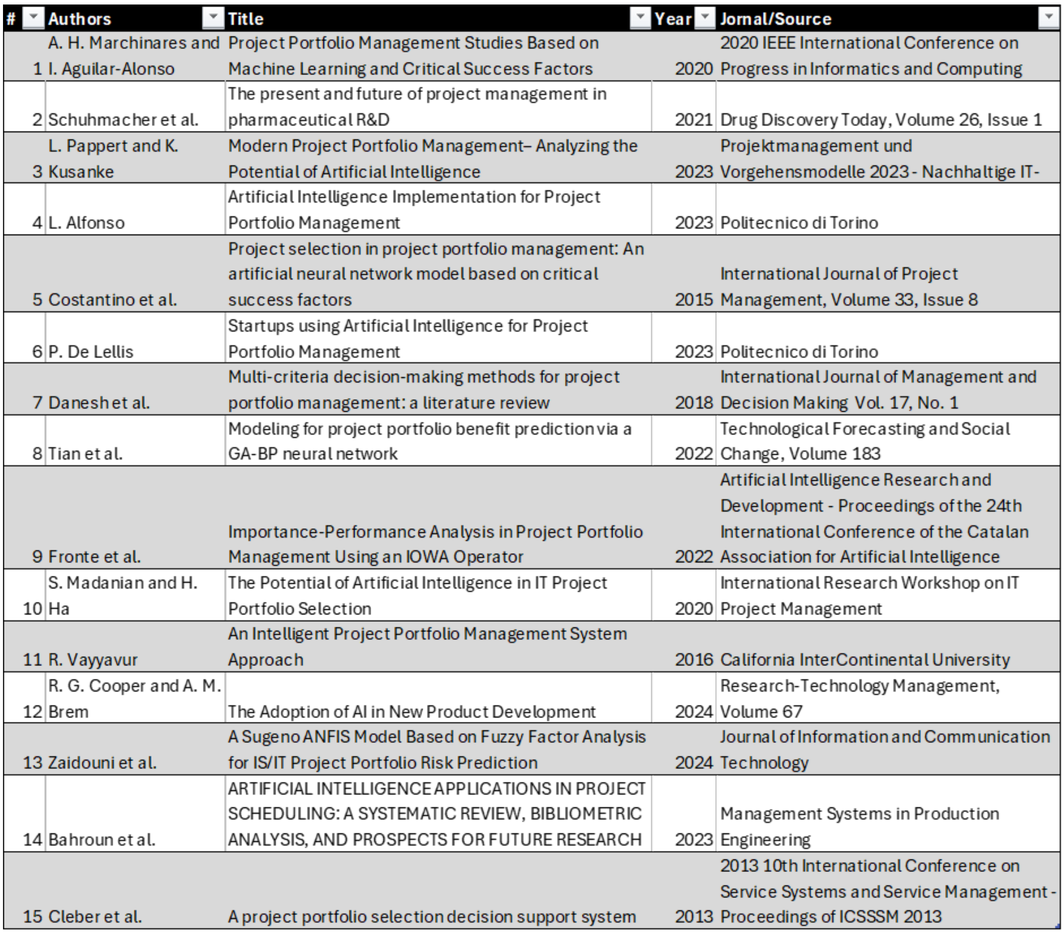
[35] Erich Schubert. Stop using the elbow criterion for k-means and how to choose the number of clusters instead. *ACM SIGKDD Explorations Newsletter*, 25(1):36–42, 6 2023.

[36] William B. Langdon, Javier Dolado, Federica Sarro, and Mark Harman. Exact Mean Absolute Error of Baseline Predictor, MARP0. *Information and Software Technology*, 73:16–18, 5 2016.

Appendix A

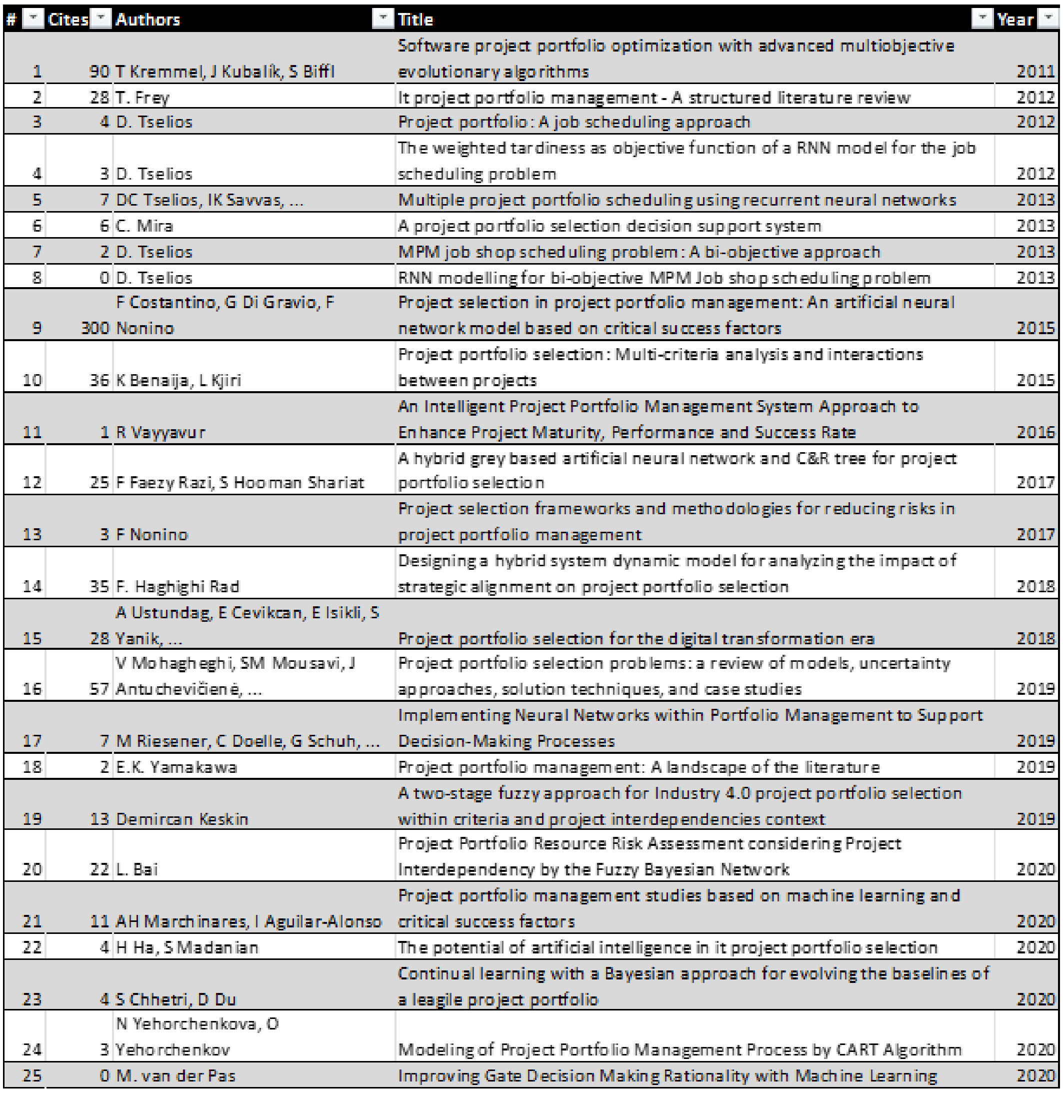
Literature Review Information

A.1 Exploratory Search Papers

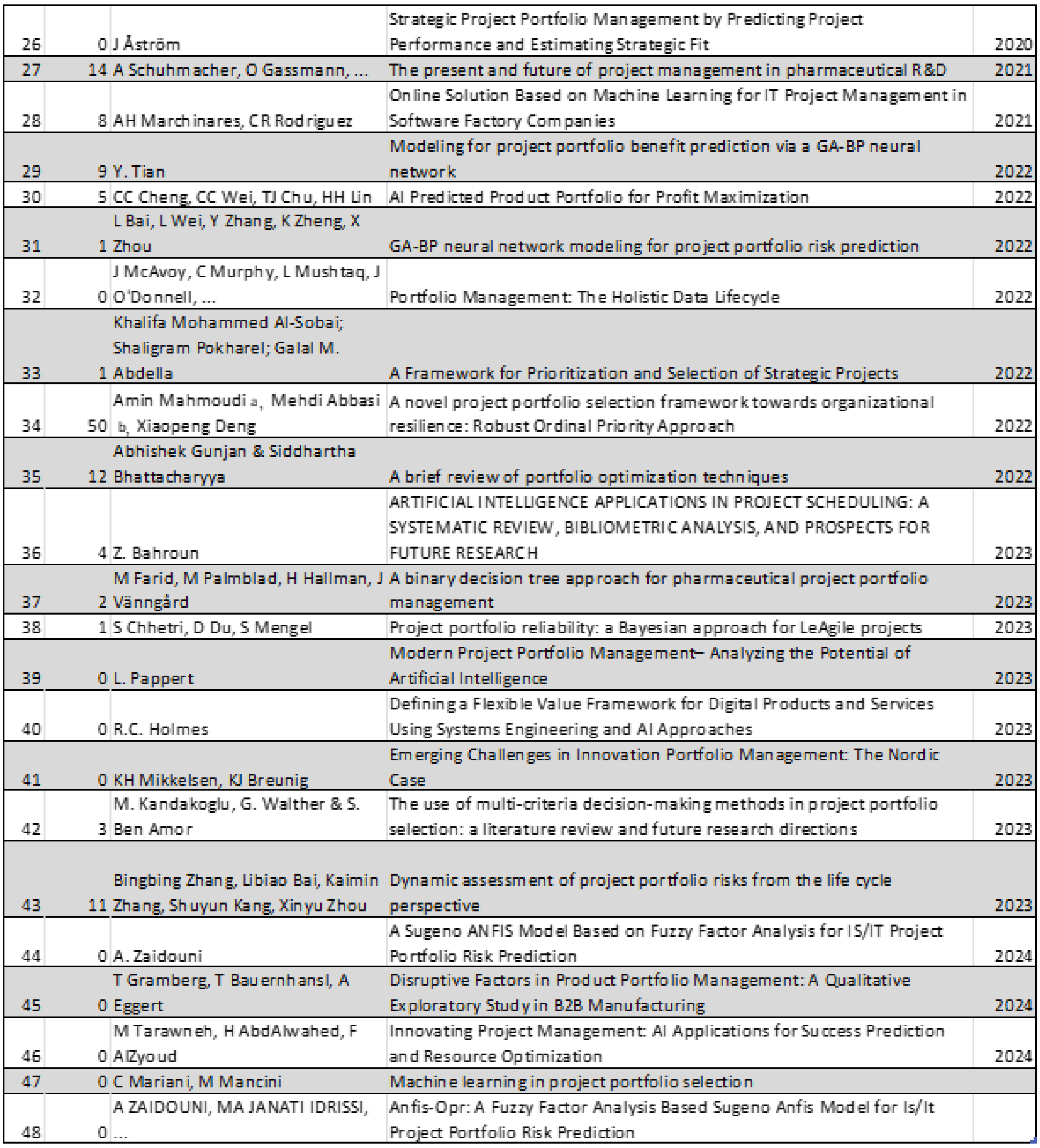


#	Authors	Title	Year	Jornal/Source
1	A. H. Marchinares and I. Aguilar-Alonso	Project Portfolio Management Studies Based on Machine Learning and Critical Success Factors	2020	2020 IEEE International Conference on Progress in Informatics and Computing
2	Schuhmacher et al.	The present and future of project management in pharmaceutical R&D	2021	Drug Discovery Today, Volume 26, Issue 1
3	L. Pappert and K. Kusanke	Modern Project Portfolio Management– Analyzing the Potential of Artificial Intelligence	2023	Projektmanagement und Vorgehensmodelle 2023 - Nachhaltige IT-
4	L. Alfonso	Artificial Intelligence Implementation for Project Portfolio Management	2023	Politecnico di Torino
5	Costantino et al.	Project selection in project portfolio management: An artificial neural network model based on critical success factors	2015	International Journal of Project Management, Volume 33, Issue 8
6	P. De Lellis	Startups using Artificial Intelligence for Project Portfolio Management	2023	Politecnico di Torino
7	Danesh et al.	Multi-criteria decision-making methods for project portfolio management: a literature review	2018	International Journal of Management and Decision Making Vol. 17, No. 1
8	Tian et al.	Modeling for project portfolio benefit prediction via a GA-BP neural network	2022	Technological Forecasting and Social Change, Volume 183
9	Fronte et al.	Importance-Performance Analysis in Project Portfolio Management Using an IOWA Operator	2022	Artificial Intelligence Research and Development - Proceedings of the 24th International Conference of the Catalan Association for Artificial Intelligence
10	S. Madanian and H. Ha	The Potential of Artificial Intelligence in IT Project Portfolio Selection	2020	International Research Workshop on IT Project Management
11	R. Vayyavur	An Intelligent Project Portfolio Management System Approach	2016	California InterContinental University
12	R. G. Cooper and A. M. Brem	The Adoption of AI in New Product Development	2024	Research-Technology Management, Volume 67
13	Zaidouni et al.	A Sugeno ANFIS Model Based on Fuzzy Factor Analysis for IS/IT Project Portfolio Risk Prediction	2024	Journal of Information and Communication Technology
14	Bahroun et al.	ARTIFICIAL INTELLIGENCE APPLICATIONS IN PROJECT SCHEDULING: A SYSTEMATIC REVIEW, BIBLIOMETRIC ANALYSIS, AND PROSPECTS FOR FUTURE RESEARCH	2023	Management Systems in Production Engineering
15	Cleber et al.	A project portfolio selection decision support system	2013	2013 10th International Conference on Service Systems and Service Management - Proceedings of ICSSSM 2013

A.2 Final Literature



#	Cites	Authors	Title	Year
1	90	T Kremmel, J Kubalík, S Biffl	Software project portfolio optimization with advanced multiobjective evolutionary algorithms	2011
2	28	T. Frey	It project portfolio management - A structured literature review	2012
3	4	D. Tselios	Project portfolio: A job scheduling approach	2012
4	3	D. Tselios	The weighted tardiness as objective function of a RNN model for the job scheduling problem	2012
5	7	DC Tselios, IK Savvas, ...	Multiple project portfolio scheduling using recurrent neural networks	2013
6	6	C. Mira	A project portfolio selection decision support system	2013
7	2	D. Tselios	MPM job shop scheduling problem: A bi-objective approach	2013
8	0	D. Tselios	RNN modelling for bi-objective MPM Job shop scheduling problem	2013
9	300	F Costantino, G Di Gravio, F Nonino	Project selection in project portfolio management: An artificial neural network model based on critical success factors	2015
10	36	K Benaija, L Kjiri	Project portfolio selection: Multi-criteria analysis and interactions between projects	2015
11	1	R Vayyavur	An Intelligent Project Portfolio Management System Approach to Enhance Project Maturity, Performance and Success Rate	2016
12	25	F Faezy Razi, S Hooman Shariat	A hybrid grey based artificial neural network and C&R tree for project portfolio selection	2017
13	3	F Nonino	Project selection frameworks and methodologies for reducing risks in project portfolio management	2017
14	35	F. Haghighi Rad	Designing a hybrid system dynamic model for analyzing the impact of strategic alignment on project portfolio selection	2018
15	28	A Ustundag, E Cevikcan, E Isikli, S Yanik, ...	Project portfolio selection for the digital transformation era	2018
16	57	V Mohagheghi, SM Mousavi, J Antuchevičienė, ...	Project portfolio selection problems: a review of models, uncertainty approaches, solution techniques, and case studies	2019
17	7	M Riesener, C Doelle, G Schuh, ...	Implementing Neural Networks within Portfolio Management to Support Decision-Making Processes	2019
18	2	E.K. Yamakawa	Project portfolio management: A landscape of the literature	2019
19	13	Demircan Keskin	A two-stage fuzzy approach for Industry 4.0 project portfolio selection within criteria and project interdependencies context	2019
20	22	L. Bai	Project Portfolio Resource Risk Assessment considering Project Interdependency by the Fuzzy Bayesian Network	2020
21	11	AH Marchinares, I Aguilar-Alonso	Project portfolio management studies based on machine learning and critical success factors	2020
22	4	H Ha, S Madanian	The potential of artificial intelligence in it project portfolio selection	2020
23	4	S Chhetri, D Du	Continual learning with a Bayesian approach for evolving the baselines of a leagile project portfolio	2020
24	3	N Yehorchenkova, O Yehorchenkov	Modeling of Project Portfolio Management Process by CART Algorithm	2020
25	0	M. van der Pas	Improving Gate Decision Making Rationality with Machine Learning	2020



26	0	J Åström	Strategic Project Portfolio Management by Predicting Project Performance and Estimating Strategic Fit	2020
27	14	A Schuhmacher, O Gassmann, ...	The present and future of project management in pharmaceutical R&D	2021
28	8	AH Marchinares, CR Rodriguez	Online Solution Based on Machine Learning for IT Project Management in Software Factory Companies	2021
29	9	Y. Tian	Modeling for project portfolio benefit prediction via a GA-BP neural network	2022
30	5	CC Cheng, CC Wei, TJ Chu, HH Lin	AI Predicted Product Portfolio for Profit Maximization	2022
31	1	L Bai, L Wei, Y Zhang, K Zheng, X Zhou	GA-BP neural network modeling for project portfolio risk prediction	2022
32	0	J McAvoy, C Murphy, L Mushtaq, J O'Donnell, ...	Portfolio Management: The Holistic Data Lifecycle	2022
33	1	Khalifa Mohammed Al-Sobai; Shaligram Pokharel; Galal M. Abdella	A Framework for Prioritization and Selection of Strategic Projects	2022
34	50	Amin Mahmoudi [a], Mehdi Abbasi [b], Xiaopeng Deng	A novel project portfolio selection framework towards organizational resilience: Robust Ordinal Priority Approach	2022
35	12	Abhishek Gunjan & Siddhartha Bhattacharyya	A brief review of portfolio optimization techniques	2022
36	4	Z. Bahroun	ARTIFICIAL INTELLIGENCE APPLICATIONS IN PROJECT SCHEDULING: A SYSTEMATIC REVIEW, BIBLIOMETRIC ANALYSIS, AND PROSPECTS FOR FUTURE RESEARCH	2023
37	2	M Farid, M Palmblad, H Hallman, J Vänngård	A binary decision tree approach for pharmaceutical project portfolio management	2023
38	1	S Chhetri, D Du, S Mengel	Project portfolio reliability: a Bayesian approach for LeAgile projects	2023
39	0	L. Pappert	Modern Project Portfolio Management– Analyzing the Potential of Artificial Intelligence	2023
40	0	R.C. Holmes	Defining a Flexible Value Framework for Digital Products and Services Using Systems Engineering and AI Approaches	2023
41	0	KH Mikkelsen, KJ Breunig	Emerging Challenges in Innovation Portfolio Management: The Nordic Case	2023
42	3	M. Kandakoglu, G. Walther & S. Ben Amor	The use of multi-criteria decision-making methods in project portfolio selection: a literature review and future research directions	2023
43	11	Bingbing Zhang, Libiao Bai, Kaimin Zhang, Shuyun Kang, Xinyu Zhou	Dynamic assessment of project portfolio risks from the life cycle perspective	2023
44	0	A. Zaidouni	A Sugeno ANFIS Model Based on Fuzzy Factor Analysis for IS/IT Project Portfolio Risk Prediction	2024
45	0	T Gramberg, T Bauernhansl, A Eggert	Disruptive Factors in Product Portfolio Management: A Qualitative Exploratory Study in B2B Manufacturing	2024
46	0	M Tarawneh, H AbdAlwahed, F AlZyoud	Innovating Project Management: AI Applications for Success Prediction and Resource Optimization	2024
47	0	C Mariani, M Mancini	Machine learning in project portfolio selection	
48	0	A ZAIDOUNI, MA JANATI IDRISSI, ...	Anfis-Opr: A Fuzzy Factor Analysis Based Sugeno Anfis Model for Is/It Project Portfolio Risk Prediction	