Metropolia
University of Applied Sciences

Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences

Full Automated AEC Scheduling Framework using Machine Learning

---

Master Thesis

Construction and Real Estate Management (ConREM)

Joint Study Program of Metropolia UAS and HTW Berlin

from

Ashraf Mahdy

Date:

Berlin, 05.July.2024

1st Supervisor: Prof. Dr.-Ing. Markus Krämer

2nd Supervisor: Ricardo Knauer

Acknowledgement

To my family who had faith in me to travel and pursue my higher education

To my ConREM Batch who made this whole journey much more lovely and fun

To all my ConREM Professors who taught me so much over the last 2 years

To my thesis supervisors who offered me their knowledge and precious time

Thank you all!

International Master of Science in Construction and Real Estate Management

Joint Study Programme of Metropolia Helsinki and HTW Berlin

Date February 2024

Conceptual Formulation

Master Thesis for Mr.	Ashraf Medhat Mahdy
Student number	Metropolia UAS | 2212804
	HTW Berlin | 585969

Topic:

Full Automated AEC Scheduling Framework using Machine Learning

Signature of 1st Supervisor

Signature of 2nd Supervisor

Introduction

Planning is an important part of professional careers. In all projects undertaken, the first and most important milestone is developing a full plan and approach on how this project is going to be tackled in the most effective way that satisfies all project stakeholders. In the simplest of terms, planning is crucial for the success of any project. In the Architecture, Engineering, and Construction (AEC) field, a successful project plan can be defined by its ability to complete, and handover the required documentation and/or built structure according to the contract requirements. An effective project plan therefore ensures all project stakeholders are working unanimously towards a common goal.

Background, and Idea Generation

One of the negatives of project planning in the AEC field is that it's an esoteric job. In other words, the creation of an appropriate schedule at large depends on the project assigned planners' experience and their employment time in the company. Despite of which, many Construction projects involve poor performance metrics [1] due to the major complexities in construction processes, procurement, permitting, and various other issues. This position gets even more difficult for Junior Planners in large Construction companies since they are the most likely to lack enough hands-on prior experience (e.g., estimation of realistic baseline task durations, or process for schedule modification in case of abrupt change) [2]. Therefore, they require the most amount of teaching from experienced seniors, causing strain on their job duties due to the rounds of performance checks required by said Senior.

In the scenario that the Planner decides to pursue a position at another company, they take all the learnt experience with them. Leaving the company and the new hire with historical software-based schedules to decode in a period that is lengthened in case the existing senior in that department is not available for advice on the company processes, feedback, or decided against staying within the company as well. Additionally, the most famous scheduling software in Construction, Primavera P6 by Oracle, and Project by Microsoft, do not improve scheduling accuracy through previous project data from the work done during employment of the above planner. In the case of Primavera P6 for example it only acts as a Container or Database for all

schedules. The amount of existing data present is not the problem. Rather, it can be seen from the above example scenario the sheer volume of information in each project, which is only getting more complicated proportional to its size is enough to overwhelm most project planners [3] most often due to the tight scheduling deadlines they are subjected to.

One way to approach this weakness is through of technological advancements in the fields of Artificial Intelligence (AI), Machine Learning (ML), Data Science, and Analytics. Which is the focus of this master's Thesis. To tackle project complexity, AI, and more specifically ML, can provide the technological prowess to boost the project team working capability by considering increased complexity [3] from previous schedules.

Research Objective

The main aim of this Thesis is devising one or multiple ML algorithms that holistically study all previous projects' schedules to automate, provide useful insights, and feedback on baseline schedules in the process of being generated for new projects.

The feedback format is likely is in the form of a modified schedule file that can be imported to the scheduling software (for example a CSV File). This automation should lead to increased productivity and efficiency through reducing the multiple cycles of schedule reviews before approval. There are also recommendations for future research and expansion ideas to develop the product aspect of the Thesis further into a viable commercial product.

Research Questions

Below are the research questions this Thesis will attempt to resolve.

1. What is AI, and ML
2. The history of Literature on AI and ML in the Management of the AEC Industry
3. How can AI and ML benefit the AEC Industry in Scheduling?
4. How can we use AI and ML for further benefit AEC Management Processes

Research Methodology

1. The first step is to define the terms mentioned above. And justifications for choosing this approach rather than normal Statistical Analysis or Statistical Learning for prediction purposes.
2. After that, the coding language must be chosen and justified. This will likely be done through a summary table with the various Languages used in ML Coding against their capability, syntax difficulty, and average learning time to accomplish the goal.
3. The next major step is a literature review of the papers discussing the use of AI in the AEC Industry Management Processes with a focus on scheduling; researching how AI and ML can benefit Management processes. Specifically Scheduling benefits.
4. Original work
    a. Learning the chosen coding language
    b. Creation of Realistic Synthetic Datasets to feed the algorithm.
    c. Coding, testing and validation.

Research Resources

Literature Review, Grammar, and Citations Management

1. MetKat Finna (Metropolia E libarary)
2. Google Scholar
3. ScienceDirect
4. ResearchGate
5. Others research repositories
6. QuillBot
7. Grammarly
8. Microsoft 365 Editor

Coding Language of choice, Tutorials, and Package Manager

9. YouTube
10. PyTorch.Org
11. Programming Forums for information exchange

12. Personal information exchange from Family and Friends
13. Coding corrections and optimization from Large Language AI Models (LLMs) (Open AI's ChatGPT, Google Bard, and Microsoft Bing AI)

Research Agenda and Schedule

This is the current preliminary plan for the whole period.

1. Approx. 3 Months, May to August

Learn Python and Obtain the schedules and modify them as needed.

2. Approx. 2 Months, September, and October

Code the Algorithm, Test, and modify as needed.

3. Approx. 3 Months, November to January

Write and finalize the first couple of Chapters (Introduction, Research methodology, and Literature Review)

4. Approx. 5 Months, February to July

Write rest of MSc.

5. Approx. 1 Month, July to July 24th

Finalize and upload MSc. Thesis Document

References

[1] Junaid T, Shujaa Safdar Gardezi S. Study the delays and conflicts for construction projects and their mutual relationship: A review. *Ain Shams Engineering Journal*; 14. Epub ahead of print February 2023. DOI: 10.1016/j.asej.2022.101815.

[2] Schott C. Problems With Project Scheduling | Bizfluent. *Bizfluent*, https://bizfluent.com/list-6765471-problems-project-scheduling.html (2017, accessed March 16, 2023).

[3] Ansar A. The Future of Megaprojects. *foresight.works*, https://www.foresight.works/blog/why-traditional-scheduling-software-isnt-giving-the-results-you-need (2023, accessed March 16, 2023).

Abstract

The AEC industry is unfortunately one of the slowest to embrace technological advancements. And therefore, the managerial aspect of something like planning and scheduling is very reliant on heuristic processes and decisions by people in command with little use of previous historical data. Historical data accessibility is a non-issue, manually mining relevant information for future projects, on the other hand is just too time-consuming due to unique intricacies of projects, and differences in construction methodologies. This master's thesis aims to propose a synergistic framework for the automation of AEC scheduling using Machine Learning through holistically studying all accessible previous projects' schedules to provide useful insights, feedback, and in due course automate a large portion of baseline schedules in the process of being generated for new projects.

This entailed the development of optimized models for studying different aspects of scheduling. These aspects are most probable task list generation, activity relationships, and optimized durations. There is more than one approach provided for the activity relationships and task list generation models. Each of the alternative approaches are evaluated for suitability. And a final cohesive Inference File was created to use all the models for a final output.

Alternative approaches for activity relationships are utilizing a GNN and utilizing a Fine-Tuned LLM, while the final model utilizes a traditional classification layout with a Dataset format that involves more effort to develop. The first model for task generation is utilizing LLM approach for the work was already done on it from the previous model; in addition to a Sequence-to-Sequence LSTM RNN with a horizontal and vertical task list formatting. Finally, the durations model utilized Sci-Kit Learn's library for regression analysis with multiple model comparisons whereby the best model was selected according to a weighted average criterion set by the author. The datasets used to train, and validate these models were a fully synthetic built upon augmentation, sensitivity analysis, and randomness of a real-world dataset. Training and validation scores of each model indicate a promising ability to automate the scheduling portion significantly. Additionally, effort was put to ensure that the whole framework flows together.

Table of Figures

Table of Tables

List of Abbreviations

- United States - US
- Architecture, Engineering, and Construction - AEC
- Artificial Intelligence - AI
- Machine Learning - ML
- Deep Learning - DL
- Neural Networks - NN
- Genetic Algorithms - GA
- Knowledge Based System - KBS
- Case Based Reasoning - CBR
- Dynamic Process Template - DPT
- Critical Success Factors - CSFs
- Random Forest Classifier - RFC
- Decision Tree Classifier - DTC
- University of Cambridge's Construction Information Technology - CIT
- Long Short-Term Memory Recurrent Neural Networks - LSTM-RNNs
- Optimized Pathways for Scheduling Execution using AI - AI-OPSE
- Part-of-Activity Tagging - POA
- Recurrent Neural Network Model - RNN
- Bidirectional Long Short-Term Memory - BI-LSTM
- Natural Language Processing - NLP
- Part of Speech - POS
- Case-Based Project Management Assistant - CaBMA
- Root Mean Squared Error - RMSE
- Correlation Coefficient - R^2
- Mean Absolute Percentage Error - MAPE
- Strength, Weakness, Opportunities, and Threats - SWOT
- Graphical User Interface - GUI
- Line of Balance - LOB
- Random Number Generator - RNG
- Built Up Area - BUA
- Cross-Validation - CV
- K Nearest Neighbours – KNN
- Graphical Neural Networks – GNNs
- Graph Attention Networks – GAT
- Knowledge Graph Embedding – KGE
- Large Language Model - LLM
- Bidirectional Encoder Representations from Transformers - BERT
- Transfer Learning - TL

Introduction

Importance of Planning

"*By failing to plan, you are preparing to fail*". This is a very famous quote by Sir Benjamin Franklin, one of the United States' (US) Founding Fathers (*HISTORY, 2009*) that is touted by most Instructors in the first lecture of courses focused on Planning and Scheduling. Planning is considered a critical thinking skill present in all aspects of our individual lives. Planning is defined as "The establishment of goals, policies, and procedures for a social or economic unit" (*Merriam-Webster Dictionary, 2023*). Extrapolating on the above definition, whenever we record, write down tasks, and goals for the day, week, or a period in general, we are planning our personal lives. Consequently, planning is an important part of our professional lives as well. In all projects we undertake, the first and most important milestone is developing a full project plan and approach on how this project is going to be tackled in the most effective way that satisfies all project stakeholders. In the simplest of terms, planning is crucial for the success of any project.

Narrowing down the focus to the field of this master's thesis, Design and Construction, this can be defined as "a set of established processes used to make a decision on what tasks must be performed to achieve the project's set objectives within schedule and cost" (*Wolejszo, 2020*). Equally, a respective definition is the completion and handover of the required structure according to the contractual requirements. This includes the required and approved specifications, scheduled duration, budgetary constraints, and finally effective and safe human and material resource management. An effective project plan ensures all project stakeholders are working unanimously towards a common goal. Which allows for better success in managing inherent project risk. Additionally, the benefits of a high-quality planning strategy and approach are shown in keeping the project on track, and ensuring deadlines are met and objectives are achieved as close to their due date as possible. In essence, it provides a clear roadmap for the team involved in the project's execution and helps them prioritize certain tasks, allocate more or fewer resources as needed, and make more informed decisions.

The AEC Industry's overall value exceeds ten trillion USD (around 9.5×10^7 Euro) per year, around $\frac{1}{7}$ of global GDP (Karmakar and Delhi, 2021, Srivastava, 2023). The detail of this calculation is shown in Figure (1) below.

Figure 1 – AEC Industry Valuation (Barbosa et al. via McKinsey & Company, 2017)

Problems with Traditional Planning in AEC Industry

Now it has been established that AEC Planning and Scheduling is an essential process in the Management strategy of any new project. Delving deeper into the education aspect, undergraduate and post graduate programs focusing on Engineering Management include a course on the basics of planning, where the theory is taught and practiced through a combination of manually solving problems, and software-based scheduling plus smaller term projects or Thesis based implementation. However, leveraging the Author's personal anecdote as a BSc. holder in Construction Engineering with a specialization in Construction Management, one leading downside of these courses is that to allow students to calculate activity durations and other teaching outcomes, the courses provide "default productivity rates" that are used for that purpose. Most of the time, they are based on real-life country data or even company specific data if the instructor has access to them from prior experience. Nonetheless, upon obtaining a real-world position after graduation there are often surprises by the nature of work being only tangentially related to academic environment.

In the end, this leads to the job being at large a position highly dependent on the experience of the project assigned planner. Since the creation of an appropriate schedule depends on the planners' experience and their employment time in the company, and to an extent the project manager given they likely have the necessary professional experience. While past project schedules are accessible, manually mining contained information to build upon for future projects is just too time-consuming and error-prone, intricacies of projects, and differences in construction methodologies, are a big portion of the inability to create a comprehensive tool or framework and, in many circumstances, just unworkable. As a result, and as mentioned in the top sentence of this section, the planning, scheduling and follow up tasks are dependent on manual effort from skilled practitioners (Amer and Golparvar-Fard, 2021).

In fact, despite the experience criteria, and the AEC industry's capability of using improved technological aspects for management, many construction projects involve poor performance metrics due to the major complexities in construction processes, procurement, permitting, and various other issues (Tariq and Gardezi, 2023). However,

the second highest-ranking identified cause for delays and conflicts internationally out of the top five causes is “Poor project planning and scheduling” of (Tariq and Gardezi, 2023) only superseded by “Owner related financial setbacks”. This is also corroborated through more than one literature review on the top causes of delays that ranked “Unrealistic project scheduling” as the top cause for project delays (Mbala et al., 2018), sharing the rank with Project Complexity and Labor Shortage. And a literature review on the top causes of Global Delays and Conflicts (D&Cs) in construction projects, where “project planning and scheduling” came in second place behind owner financial problems and ahead of material problems (Tariq and Safdar Gardezi, 2023). Consequently, it can be seen how newly hired planners often need a disproportionate amount of time to understand the company’s practices and operating methods despite their prior experience or education. Which in turn can lead to haphazard planning approaches due in part to top-management urgencies creating a crunch cycle that turns the project into a poorly planned plight. In research published by McKinsey, their data indicates more than 75% of mega projects are almost 1.5 times behind schedule (McKinsey Productivity Sciences Centre, 2015).

Limitations of Traditional Planning

Despite technological advancements in the last 2 decades, traditional project planning software is still deficient in its ability to plan in a more effective way in very big projects, as it has been demonstrated internationally, around only 10% of mega projects are completed on time and under budget (*foresight.works, 2023*) and most still rely entirely on human workflows for planning and scheduling. Inadequate project scheduling can be the cause of a slippery slope of project complexities, like disputes, significant resource spending on amicable settlements, and the principal failure to address the core scheduling issues (Kumar, 2022). Even though at the beginning stages of the project developing an effective project schedule approach is critical, the ever-changing nature of AEC projects is best supported through a method or tool that accommodates for both the organisation's internal and external dependencies. As well as the potential of analysis necessary from progression and risk events (Kumar, 2022).

There are 5 overarching reasons for the current landscape (Amer et al., 2021):

1. There is no developed framework for storing Scheduling and Planning Knowledge
2. There is an overreliance on manual work templates for each project with minimal reuse.
3. There is shortage of research on automated scheduling methods that do not require heavy manual data preparation.
4. There is very limited model and framework verification on existing projects likely due to the secrecy of publicly available project information.
5. There is fragmentation in the research focusing on automated planning approaches versus optimization.

In traditional heuristics based Planning and Scheduling activity durations are calculated based on average known daily productivity rates divided by the total quantity. This step is however only a small part of the scheduling process specially in bigger companies. The Project baseline Schedule is usually subject to the inputs of the Project Controls Sector Director, Project Manager, and Client inputs on certain portions to highlight important milestones. This arrangement gets even more demanding and difficult for Junior Planners in large Construction companies since they are the most likely to lack enough proper hands-on experience. Therefore, they require the most amount of teaching from experienced seniors, causing strain on their job duties due to the rounds of performance checks. The final figurative nail in the coffin is if the Junior Planner decides to pursue a position at another company, they take all the learnt experience with them. Leaving the company and the new hire with historical software-based schedules to decode in a period that is lengthened in case the existing senior in that department is not available for advice on the company processes, feedback, or decided against staying within the company as well. Additionally, the most famous scheduling software in Construction, Primavera P6 by Oracle, and Project by Microsoft do not improve scheduling accuracy through previous project data from the work done during employment of the above planner. In the case of Primavera P6 for example it only acts as a Container or Database for all schedules.

Further Limitations in Developing Countries

In the PhD Thesis by (Salleh, 2009) aimed at unfolding the most influential reasons behind successful project delivery and vice versa included a portion on the struggles of the AEC industry in more developing countries. The most important of which inside the focus of this master's thesis is the resistance to adoption of progress, and general failure to adopt suitable practices. The research attributed a portion of these limitations to the existence of an informal AEC sector in these countries that avoids structured governance and laws in their execution.

Background and Idea Generation

Scheduling Software, Narrowing Down the Problem

The introductory sections propose the following question, how can AEC planners advance beyond scheduling software flaws? The amount of existing data for effective planning is likely not the problem. On the contrary, from the above example scenario with junior planners the sheer volume of information in each project, only getting more complicated proportional to its size is enough to overwhelm project planners not least due to the tight scheduling deadlines they are most often subjected to (foresight.works, 2023). Another common problem that lends its weight is that people are usually prone to an optimistic bias. Meaning plans usually take longer to complete than estimated. In increasingly difficult jobs, time loss is magnified (foresight.works, 2023). Compounded by the lack of historical analysis to inform future efforts resulting in prejudice or blind spots. In most cases there simply isn't time to analyse previous data and form conclusions about which tasks likely to run late or can be compressed in duration. Data in construction business is larger, varied, complicated, and created at a faster rate than ever before. The collective use of all this data is necessary to assist in making better informed decisions. However, all this data is considered useless if it is not used in predictive analysis for newer projects (Dockery, 2022). A very useful summary of the lacking emphasis on scheduling tools and processes is written by (Kumar, 2022) shown below.

1. Inaccurate estimates of resource needs, demographic trends, and technical difficulties
2. Errors in future estimates of local environment

3. Identification of potential variations and associated risks not in the same line of thought as previously defined decision-making processes.
4. The general reliance of heuristics and absence of fresh ideas leading to short-sighted habits and thinking.

Objective and Vision, Enter Machine Learning

One of the main ways to attempt the improvement of the problem explained above is through taking advantage of recent surge and technological advancements in the fields of Artificial Intelligence (AI), Machine Learning (ML), Data Science and Analytics (foresight.works, 2023). AI, and more especially ML, can give the technological power to raise the project team's working capabilities by considering increasing complexity from past schedules. Which is the focus of this master's thesis. This idea was further corroborated during the feasibility stage through finding a research paper with a similar aim that is summarized in the following sentence, "…considering the abundance of significant previous data, machine learning appears to be a viable solution to the scalability problem connected with gathering and disseminating planning and scheduling expertise. As a result, our hypothesis is to investigate if scheduling and planning knowledge and job templates may be modelled using machine learning with little human intervention." (Amer and Golparvar-Fard, 2021). This research paper will be covered more in the Literature Review section.

The main aim of this Thesis is devising one or multiple ML algorithms that holistically study all previous projects' schedules to provide useful insights, feedback, and in due course automate a large portion of baseline schedules in the process of being generated for new projects. The feedback format is likely is in the form of a modified schedule file that can be imported to the scheduling software (for example a CSV File). This automation should lead to increased productivity and efficiency through reducing the multiple cycles of schedule reviews before approval. There are also recommendations for future research and expansion ideas to develop the product aspect of the Thesis further into a viable commercial product. Additionally, in the early period of research, and as due diligence before committing to the idea fully, a small feasibility study for the proposed solution was conducted where the author contacted several people with more experience in the field

of AI and ML to determine the feasibility of the solution in the desired time frame. From family members who studied computer engineering, neighbours in the same field, to famous programming forums such as Stack Overflow, and GitHub. A formatted version of the problem is written as follows “I am trying to build Machine Learning algorithms that study historical construction project schedules for use in future schedules to understand Activity Relations, Dependencies, and Optimize Activity Durations. My time frame is 6 months as I am trying to finalize it before the start of the second study year. How feasible is it to do so and what is the best Programming Language to utilize?”. The collective response analysis will be covered in a later section of the Thesis discussing the Programming Language of choice. It is currently sufficient to mention the feasibility of the intended solution was positively given by multiple correspondents.

The final part of this section is about rationalization of the thesis focus. A couple of explanations are provided. The first is the general act of optimization, as explained prior, on occasion calculations are not entirely correct, this prediction serves as a second seeing eye of sorts rather than replacing the manual effort entirely at this stage, saving iteration times and management approval cycles. The optimization approach is nothing new, it is used widely in Tender and Retail Pricing optimizations. Secondly, this allows quicker overview and prototyping of project schedule that fits within the contractual time frame, as it is very occasional that properly calculated end up spilling over contractual time frame and thus requiring certain critical activities crashing. Crashing those activities based on model predictions of what can be modified is a better alternative to the widely used heuristic approach to fit the overall contractual duration as the prediction result will allow for selection of these activities according to the difference between calculation and prediction of activities that fall on the Critical Path for Optimization. An additional rationalization is mentioned by ORACLE’s VP of Data Science & Analytics (Venkatasubramanian, 2021) on the company’s website on the importance of having Future or Lead indicators for decision making. Justifying that most project indicators are considered Lag Indicators that promote reactive actions by involved parties rather than proactive actions as in for example, taking an umbrella due to the weather forecast indicating high chance of raining.

This Page was intentionally left blank as a separator between chapters.

Literature Review

The first step is setting up the correct definitions of AI, ML, and their related factors to classify the target technique that will be used to develop the solution (i.e., what kind of ML algorithms will be used). This part uses established definitions and examples from market leaders in the field of AI and ML like IBM, Nvidia, and Sci-Kit Learn.

Afterwards, a simplified overview of AI use cases in the AEC industry to highlight the lack of focus on the project management scheduling aspect. Finally, to narrow down the research to the intended area only a comprehensive Literature review of the thesis focus is carried out to find as much research focusing on the same idea or closely related ideas. Noting that the literature review will concentrate on papers with a clear process, as well as a focus on developing a viable tool for future use. Multiple research repositories will be used such as but not limited to the following:

1. Journal of Information Technology in Construction: ITCon
2. Multidisciplinary Digital Publishing Institute (MDPI)
3. HTW Berlin and Metropolia UAS Repositories
4. Google Scholar
5. ResearchGate
6. CodeAcademy
7. Academia.edu
8. ScienceDirect

Important Definitions and use cases.

Defining AI

The first important point to tackle is defining AI and ML, in addition to their types. AI is defined as the use of computers and technology to simulate the human mind's problem-solving and decision-making skills (What is Artificial Intelligence (AI)? | IBM, n.d.). This definition is extrapolated from John McCarthy's definition “It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.” In its most

basic form, AI employs computer science and large datasets to solve problems. It is the umbrella term under which ML and Deep Learning (DL) fall. AI algorithms are used in these areas to develop expert systems for predicting or classifying data. (What is Artificial Intelligence (AI)? | IBM, n.d.). However, due to its very large data volume processing capabilities it can even outshine human insights or decision-making (Dockery, 2022).

Weak (Narrow) and Strong AI

There are two main types of AI, Weak or Narrow AI which has been trained and focuses on performing certain tasks. Most famous AI examples in use today are Narrow AI. This form of AI is anything but feeble; it powers AI Assistants from Apple, Amazon, Google, and IBM. Conversely, Artificial General Intelligence (AGI) and Artificial Super Intelligence (ASI) are components of strong AI. AGI is a speculative version in which a machine possesses an intellect comparable to humans; it has a self-aware awareness capable of solving problems, learning, and planning. While Artificial Super Intelligence (ASI), would outperform the human brain's intelligence and abilities. Strong AI is currently theoretical. (What is Artificial Intelligence (AI)? | IBM, n.d.)

General Uses of AI

The use of AI can be broken down into 2 main categories. The first being Knowledge Based Systems (KBSs), and the latter being Optimization. KBSs are an artificial intelligence discipline in which computers make judgements based on prior information (Dockery, 2022). KBSs collect and study huge datasets from a variety of sources, it is broken down into knowledge base, which stores all the information, and an inference engine, which analyses the data in the framework of the knowledge base. These systems generate insights to help individuals make better decisions (McCarthy, 2007). A usage example is by Medical Professionals for providing more accurate diagnosis (Dockery, 2022). While KBSs can manage massive amounts of data. The problem is ensuring the validity of the dataset and its features being relevant, accurate, and valuable. Since the AEC industry involves multiple stakeholders, and more in a single project, the data would originate from many sources and be of varying quality. Therefore, data collection and validation are required. Not to mention the implicit difficulty of obtaining said data complicated by proprietary and legal issues (Dockery, 2022).

On the other hand, optimisation is used to analyse and predict probable outcomes to discover the best of all options. It tries to boost production and efficiency while saving time and money (Dockery, 2022). In the AEC industry it might improve labour work schedules, reduce material prices, or increase energy efficiency. The pitfall however is the requirement of a massive dataset that may include design, site data, material qualities, and construction tactics as examples. Additionally, High-performance computing is required for real-time processing and massive data volumes (Dockery, 2022).

Defining ML

Secondly, it is paramount to separate the term ML from AI, as mentioned above in the AI definition and use cases, it is considered the umbrella under which ML falls. In that respect, ML is a technique for teaching a machine to understand and study from its inputs without explicit hard coding or programming for each situation. ML in a very broad term assists a machine in reaching its AI status (Copeland, 2023) and is therefore a subfield in computer science that utilises data and algorithms to mimic how people learn, progressively improving its accuracy (What is Machine Learning? | IBM, n.d.). As a fact it is mentioned that Arthur Samuel, one of their former employees, is credited with coining the phrase "machine learning" through his research that led to the first computer beating the world champion in checkers in 1962 (What is Machine Learning? | IBM, n.d.).

Uses of ML Algorithms

The most straight forward description of ML use cases is that general or specific Algorithms are taught using statistical approaches to do two tasks, produce classifications or make future predictions based on a specific set of input data, compared to historical and critical insights in data mining. These insights then guide decisions within applications and enterprises, with the end goal of influencing key growth indicators within the company for additional profit, product ideas, and customer needs (What is Machine Learning? | IBM, n.d., UC Berkely, 2020). Because an ML algorithm updates on its own, the analytical accuracy increases with each run as it learns from the data it analyses. This iterative aspect of machine learning proves to be unique and significant since it happens without

human interaction, allowing the algorithm to discover unexpected insights without being explicitly designed to do so (UC Berkeley, 2020).

The Classification Problem can also be further broken down into 2 Parts. Multi-Class Classification: in simple terms, it is classifying a Target Input into a single Category of Available Categories called Labels. For example, Given the Model Name and Production year of a Car; the Car is Classified as “Audi”, “BMW” or “Mercedes”. And Multi-Label Classification: which is when a Target Input can be classified into multiple Categories. For example, a Film or TV-Series can be classified as “Comedy” and “Action” at the same time.

ML, Deep Learning (DL), and Neural Networks (NNs).

The final piece in this section in need of defining is the limit of this thesis in relation to the other buzzword terms in the field of AI. Namely DL and NNs. NNs are a branch of ML, and DL is a branch of NNs (What is Machine Learning? | IBM, n.d.). NNs are a structure of nodes. The first level is an input node, the intermediate levels are hidden ones, and finally an output level each level is linked to another and has weight and threshold. If the output of level exceeds threshold value of the next level, it is activated and so on. Otherwise, no data gets sent to the connected levels. DL is concerned with the number of intermediate layers in the NN as it is only considered a DL-NN if the number of intermediate levels exceeds 1, and consequently the total number of levels is more than

3 since there is 2 default levels, the input and output. The next Diagram in Figure (2) shows a graphical summary of the definitions mentioned above.

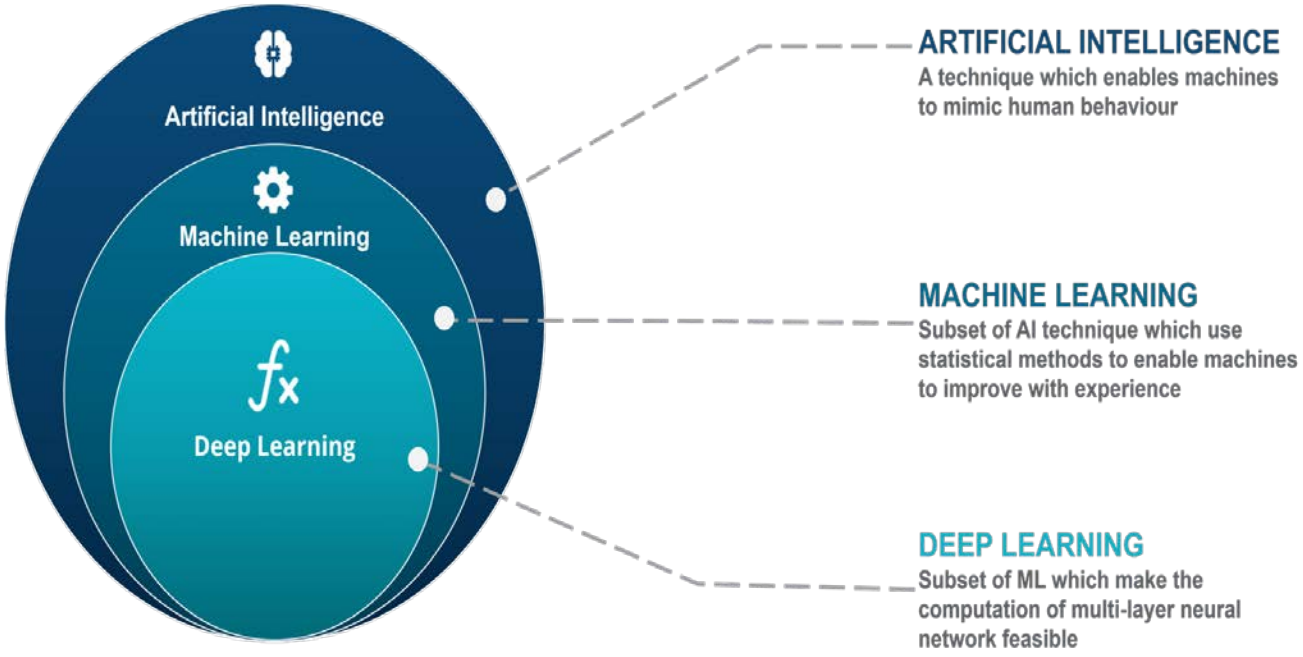


Figure 2 - Simple Hierarchy of AI and its subsets (Atul via Edureka, 2023)

Justification of ML versus Statistical Learning (SL) approach

An important point to address before moving forward is the justification of an ML approach versus a reasonably easier to implement Statistical Modelling approach. Even though ML is, strictly speaking, statistical methods, it is not accurate to consider it advanced statistics. This quote is an appropriate entry to explain the difference between both approaches “The major difference between machine learning and statistics is their purpose. Machine learning models are designed to make the most accurate predictions possible. Statistical models are designed for inference about the relationships between variables.” (Stewart, 2019). The umbrella term in mathematical analysis of data is statistics; it cannot be performed without data and a statistical data model may be used to infer the relationships in data or develop a model for prediction purposes. This section

will cover the differences in two main points first, how statistics vary from the use of machine learning, and second, how statistical models differ from machine learning.

To be more specific, there are several statistical approaches that can generate forecasts, but the accuracy of their predictions is not their strong suit. Similarly, machine learning models vary in their interpretability, ranging from very interpretable lasso regression models to opaque neural networks, although they often trade interpretability for predictive capability (Stewart, 2019, and Bzdok et al., 2018). One popular reason for the confusion is linear regression. An ML Model using a Linear Regressor can be trained to produce the same results as a statistical regression model using minimization of squared error. In this example the ML model is being trained which entails utilizing a subset of the dataset, and model performance is determined through testing the trained model on a subset of the dataset hidden during training. In this situation, the goal of machine learning is to get the most successful outcome on the test set (Stewart, 2019, and Bzdok et al., 2018). On the contrary, we construct a straight line for data in a statistical model without the need for separating the dataset. The two approaches differ in the end goal, a statistical model is only done to illustrate the connection between data points and end variable, not to predict future data. In simple terms, statistical inference. It is still possible of course to utilize the model for predictions, as a primary goal, but it will not be assessed with a test set, instead considering the relevance and reliability of the model parameters. The goal of ML Modelling on the opposite side is to create a model that can generate repeatable predictions with little or no concern if a model is interpretable. In simple terms, ML is centered around outcomes (Stewart, 2019, and Bzdok et al., 2018). The above section is summarized in the two diagrams below Figure (3)

Basis of Comparison	Machine Learning	Statistics
Definition	Machine learning is a set of steps or rules fed by the user where machine understands and train by itself	Statistics is a mathematical concept in finding the patterns from the data.
Usage	To predict future events or classify an existing material	The relationship between the data points
Types	Supervised learning and unsupervised learning	Forecasting continuous variables, Regression, classification
Input-output	Features and labels	Datapoints
Use cases	For hypothesis	Correlation between the data points, univariate, multivariable
Ease of use	Mathematics and Algorithms	Mathematics knowledge
Applications	Weather forecast, topic modeling, Predictive modeling	Descriptive statistics, finding patterns, outliers in the data
Field	Data analytics, Artificial intelligence	Artificial intelligence, data science research labs.
Stands out	Predominant algorithms and concepts like neural networks	Derivatives, probabilities
Keywords	Linear regression, Random forest, support vector machine, neural networks	Covariance, univariate, multivariate, estimators, p-values, rmse

Figure 3 – Comparison Table Machine Learning and Statistics (Pedamkar, 2023)

How Classical ML Works

A basic overview of Classical ML working methodology is as follows (UC Berkeley, 2020). A simple process diagram is also included in Figure (4) after that.

1. The Labelled Dataset is split into Training and Testing subsets.
2. The Model is trained on the Training subset using randomly assigned weights.
3. The Model is then tested on the Testing subset.
4. An evaluation function (typically denoted a loss function) examines Model performance. Since the true values of the Testing subset are known they can be compared to determine Model precision.
5. Model Optimisation Process, If the model can fit the data points in the training set better, weights are modified to narrow the gap between the true values of the Testing Set and the Model Output
6. The ML Algorithm will repeat steps 2 to 5, updating weights independently until the specified accuracy is reached or a failsafe is triggered in case of no improvement.

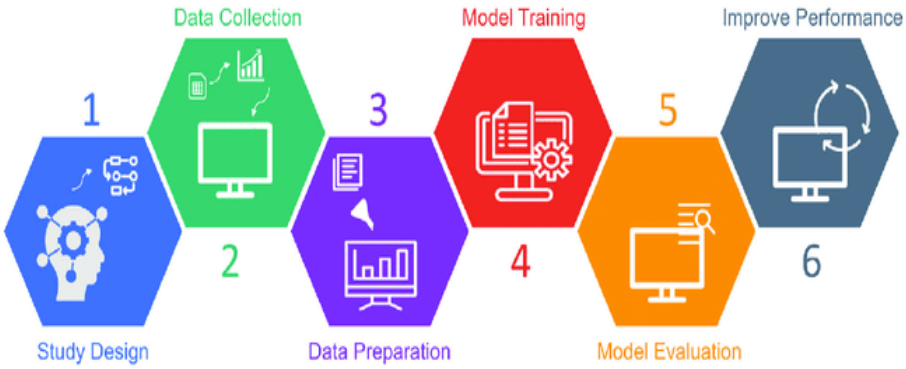


Figure 4 - Workflow of a Machine Learning Algorithm Problem (Peng et al., 2021)

Types of Classical ML

Systematically breaking down AI the final part is reached with the types of Classical ML algorithms. Classical ML is split into three main categories as follows.

1. Supervised ML
2. Unsupervised ML (or Semi-Supervised)
3. Reinforcement Learning

The use of datasets that are already manually labelled to train algorithms to properly identify and classify data or forecast and optimize future outcomes is what defines supervised ML (Salian, 2018); in simpler terms the required output compared to corresponding features or inputs are known and “labelled” accordingly in the dataset. When input data is supplied to the algorithm, the algorithm changes its coefficients until it is well fitted. This is done as a component of the cross-validations procedure to verify that the model does not overfit or underfit (Salian, 2018). Contrariwise, unsupervised ML analyses and clusters unlabelled datasets eliminating human interaction, these algorithms uncover undetected trends or data groupings. Because of its capacity to detect similarities and contrasts in data, this approach is perfect for data exploration, cross-selling tactics, consumer segmentation, and picture and pattern recognition (Salian, 2018). It additionally serves to reduce the number of unimportant features in a model with dimensionality reduction; in simpler terms unsupervised ML methods extract characteristics and detect patterns in data automatically. Semi-supervised ML sits in the halfway between the two previous approaches whereby it employs a smaller labelled data set for use to extract a larger labelled dataset from additional raw data during training. This approach improves model performance in case of insufficient labelled data for a supervised learning system; or if labelling enough data is too expensive (Salian, 2018).

To conclude this section, reinforcement ML is comparable to supervised ML, except the algorithm is not trained on sample data, it is utilizing multiple rounds of trial, error, and optimization through a reward or penalty function. In the sense that the model develops over time as it goes (Salian, 2018). To establish the optimal proposal or strategy for a specific situation, a series of successful results will be reinforced while negative outputs

will be penalized and probabilistically less to occur in future trials. A notable example to clear up the explanation more is the IBM's Watson, that won Jeopardy in 2011. Utilizing reinforcement ML to determine whether it could try an answer (or question), which square to choose on the game board, as well as how much to gamble particularly on daily doubles (Salian, 2018).

This thesis will rely exclusively on classical ML, or non-deep learning, as it is more reliant on human assistance to learn. The dataset characteristics are defined beforehand for the algorithm to grasp the distinctions between data inputs. This type of learning dataset is known as fully labelled or structured dataset. Figure (5) below ties all the above types of Classical ML together in a simple to read diagram.

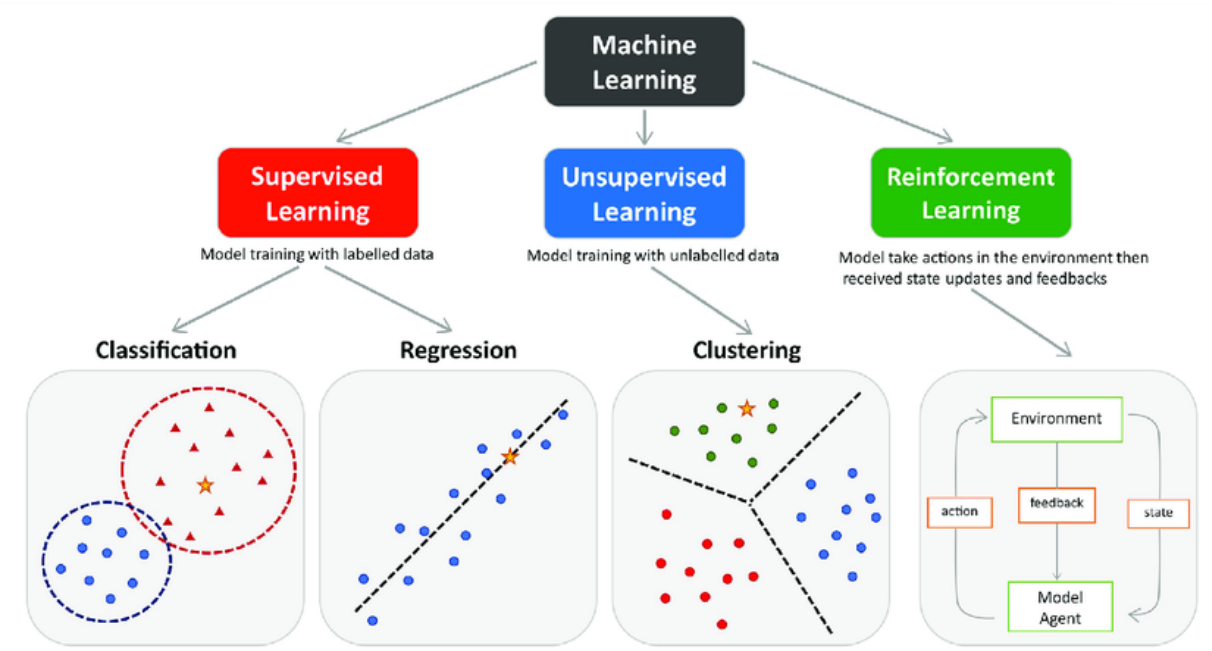
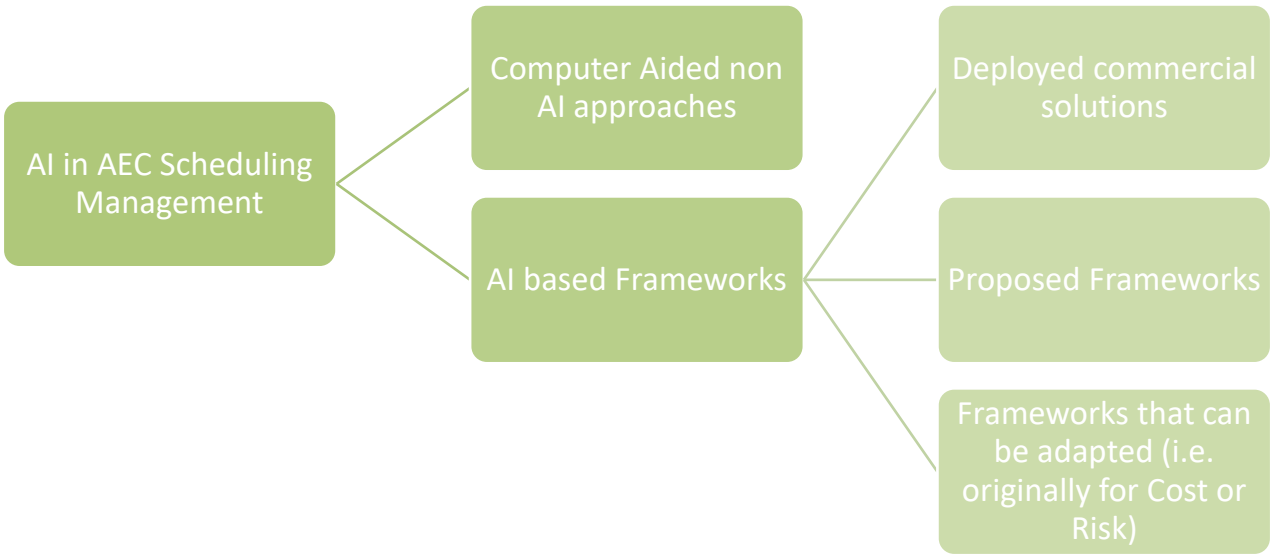


Figure 5 - The main types of ML approaches (Peng et al., 2021)

Previous Literature

The literature review is divided into two major categories, the first one is utilizing Computer Aid and capabilities to optimize construction schedules without the use of AI. And the second one is AI based frameworks. Which is further divided into three smaller ones depending on the research paper's focus. A hierarchical diagram below breaks it down.



The most apt start for literature review is to examine research trends of AI application in the AEC industry in the last century approximately. The below figure by (Abioye et al., 2021) shows a positive trend in the number of research published on the subject that is shifted to the far right of the last 3 decades from the 1990s to the late 2010s.

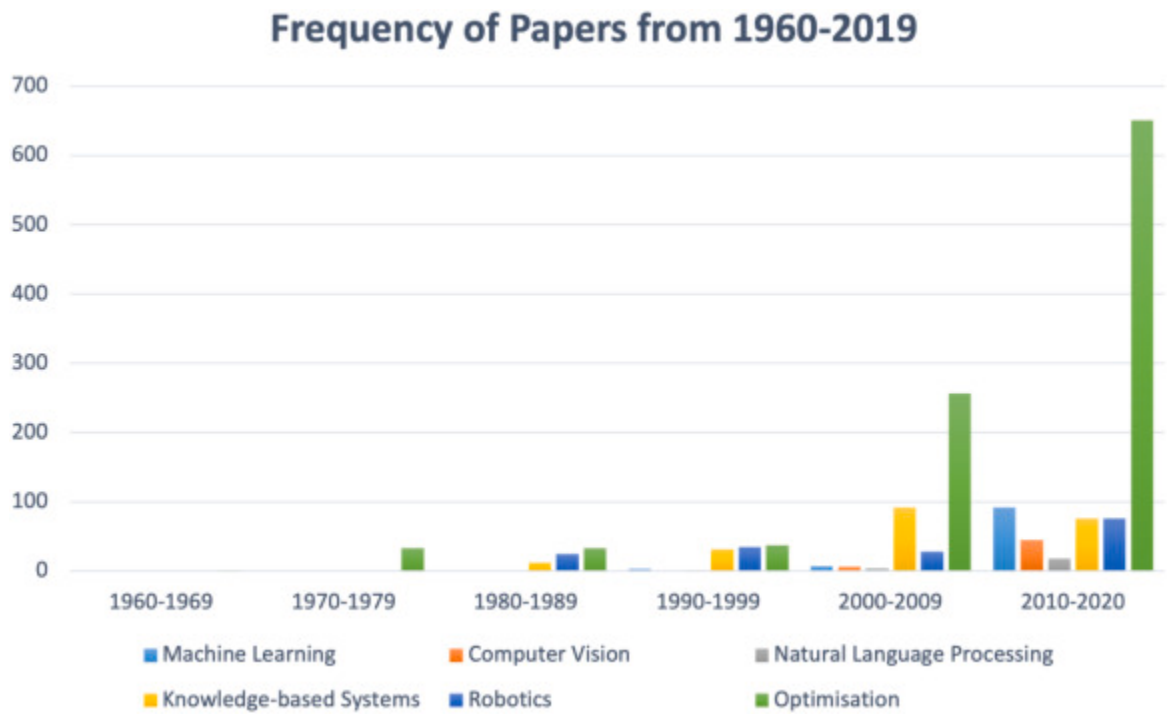


Figure 6 - AI Research focus in the AEC Industry (Abioye et al., 2021)

A comprehensive Scientific Analysis of Machine Learning Research Trends in Construction Management review published by (Van and Quoc, 2021) in the Journal of applied science and technology trends whereby they identified that a very important factor in schedule management is the level of preparation of early-stage planning has a significant influence on the project's end outcome. One of the cited studies in the research created a novel schedule-learning platform that applies ML and data science techniques to thousands of historical project schedules to improve project management as a unique and scalable method for boosting project planning dependability and trust. The model used an ANN to assist with classification models to anticipate project expenses and projected success, using early planning status as model inputs.

Computer Aided non-AI Frameworks for Automated Scheduling

It is important to initially consider methods that address Planning and Scheduling inefficiencies through Computer Aided tools that are not AI and ML dependant. One such method is a KBS Schedule Generation Framework by (Mikulakova et al., 2010) whereby the researchers propose development and assessment of schedules using BIM tools to identify the scheduling object in the schedule and the appropriate links to other objects based on industry standard BIM objects and expertise from previous projects. Additionally, their approach suggests that throughout the Execution Phase the schedule may also be re-generated flexibly with the system's assistance considering execution choices. Finally, the scheduling knowledge is saved in a logical system for future use.

Another proposed method intended for Modular Construction Projects is a simulation-based statistical predictive model for predicting the optimal arrangement of module manufacturing for maximum productivity (Bhatia et al., 2022). The suggested technique offers a complement for the experience-based approach used in standard Modular Construction Manufacturing (MCM) production planning. The researchers divided their work into three stages: data gathering, data analysis, and simulation-based planning. The Database of collected data for the input of the Statistical Predictive model included parameters as workstation process durations, module design requirements, and number of employees at workstations. Additionally, they include production line's workflow, and hours of operation. An overview is shown in the Figure below. Their model performance

indicates around 90% accuracy in their test case and is therefore regarded to be a credible prediction model.

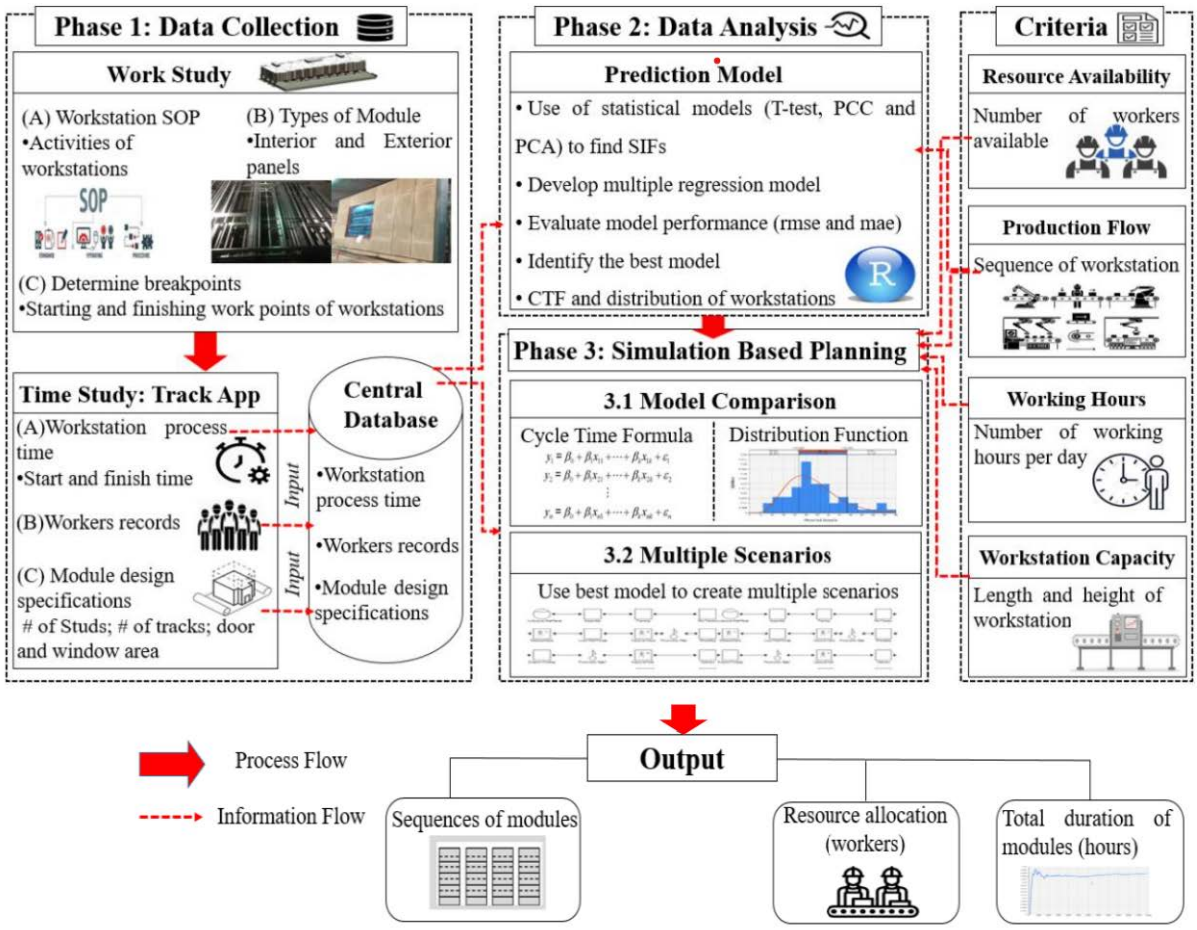


Figure 7 - Main Components of the Statistical Method (Bhatia et al., 2022)

A fantastically compelling scheduling optimization technique proposed by (Alekseytsev and Nadirov, 2022) takes into attention the fact of construction interruptions into estimating the total construction duration through determining maximum and minimum intervals of activities. The researchers clarified their approach on determining the maximum and minimum interval bounds as follows “The boundaries of the interval, as well as determining minimum and maximum construction time, are obtained by minimizing and maximizing the term of construction work performance by introducing random

interruptions into successions of critical and subcritical works". The justification for this research effort is that when scheduling a project before its actual execution it is almost impossible to predict all random interruptions that might occur due to lacking the power of hindsight. Some risk mitigation techniques may be employed to lessen the impact of said interruptions. Random Interruptions were represented as a task that requires no resources. The researchers utilized a technique called Genetic Algorithms (GAs) that falls under the category of metaheuristic algorithms; meaning through mimicking natural selection it can identify solutions for problems without a single, perfect result. This approach works sequentially as follows.

1. A population of all predictable solutions is made.
2. Each solution is evaluated for fitness to the required solution.
3. The Fittest solutions progress to the next phase while others are eliminated.
4. The progressed solutions are combined in random to create new sibling solutions.
5. Steps 2 to 4 are repeated until an optimal solution is found.

A block diagram of the proposed mechanism is provided in the figure below for a better clarification.

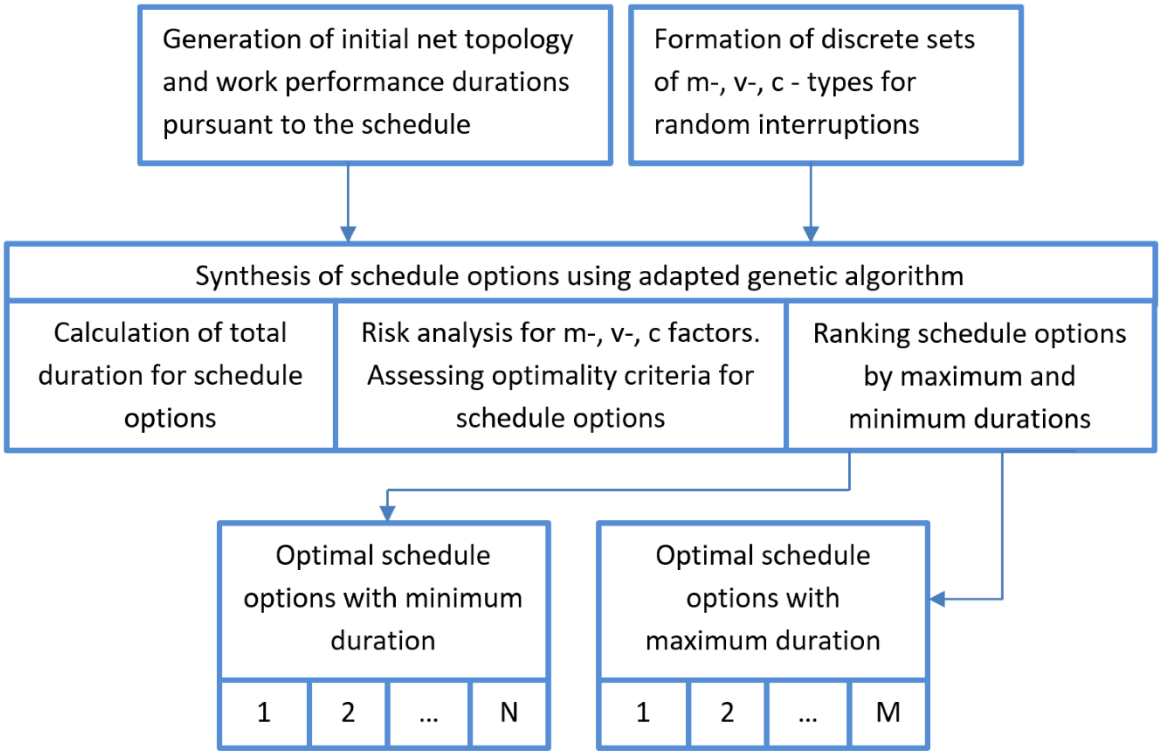


Figure 8 - GA Schedule Optimization with Interruptions (Alekseytsev and Nadirov, 2022)

AEC Industry 4.0: AI & ML in the AEC Industry

The AEC industry is notorious for being one of the latest industries to adopt technology and one of the least digitised (Dockery, 2022). As mentioned, the industry is plagued by project delays, cost inefficiencies, poor efficiency, and performance due to resistance to move to digital interactions. AEC's next big phase is AEC 4.0. It refers to taking advantage of more digital technology (Karmakar and Delhi, 2021). The 4.0 revolution is expected to be driven by data generation from multiple stages in the project design, execution, and operation. Data movement from one phase to the other, transformation of data for different use cases, and data storage or archival for easy use throughout the project lifecycle to create a collaborative environment among stakeholders (Karmakar and Delhi, 2021). Manufacturing and industrial industries have already embraced Industry 4.0.

However, the reality is that most of the AEC industry is having a very long transition period to Industry 4.0 (Karmakar and Delhi, 2021).

Tides are changing slowly towards utilizing the most out of AI in various sectors. Taking the U.S. as a primary example, in 2018 funding for construction technology start-ups more than threefold to surpass the 3 billion USD (2.85 billion Euro). Construction tech and AI start-ups surpassed fundraising records in 2018, with the industry showing thirst for innovation due to Venture Capital (VCs) raising almost 1.3 billion USD in the first half of 2018 by US-based construction businesses. in comparison to around 0.75 billion in the previous year. Figure (6) below highlights the exponential growth and disruption in the last decade that is set to only continue forward. Citing the U.S. based construction giant Bechtel developing big data and analytics centre capable of processing 5 Petabytes (5,000 Terabytes) worth of construction site information to guide its AI training algorithms. Primary amongst which is its picture recognition technology, which identifies jobsite photographs for clients and has reportedly already saved the Company more than 2 million USD. (For Construction Pros, 2019).

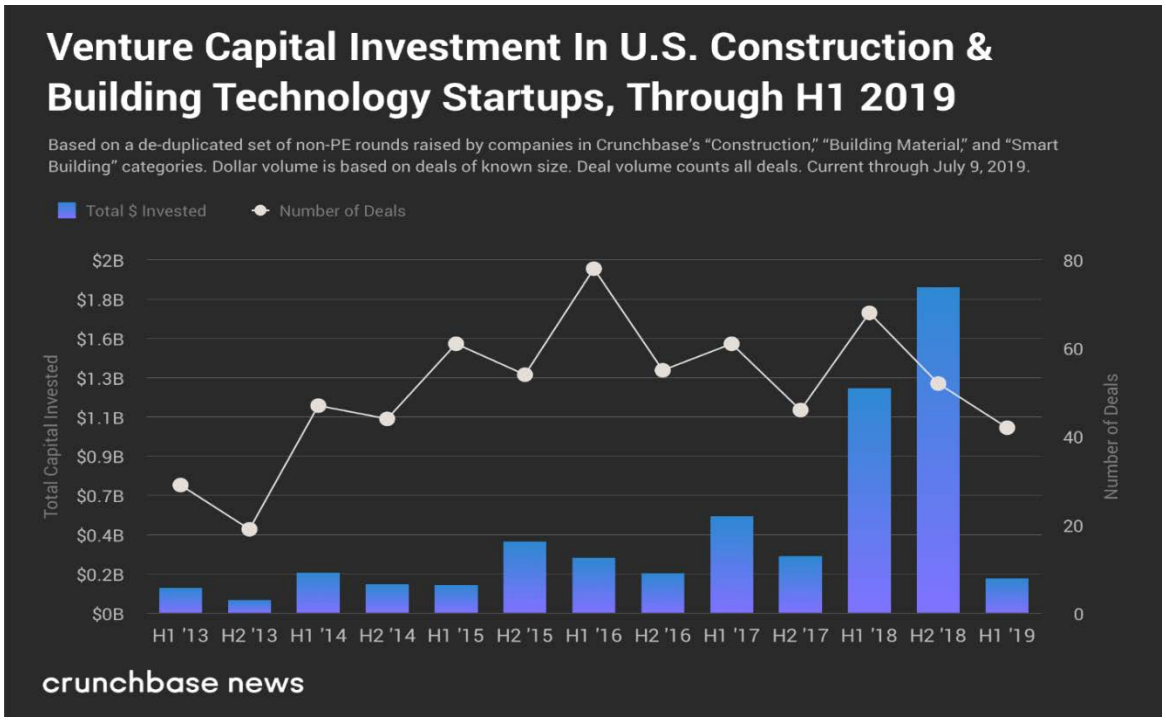


Figure 9 – VCs Investment in U.S. AEC Technology Startups, H1 2019 (Azevedo, 2019)

Majority of the time when AI and ML are mentioned in research publications or news articles about the AEC industry, the focus is about technical areas such as design improvement, faster iterations, improving the safety of the construction site, and improving the facilities management through for example predicative maintenance through making analytics systems smarter as additional data and patterns become accessible (For Construction Pros, 2018). The diagram in Figure (7) below shows the broad scope of use for these technologies along with their main subcategories.

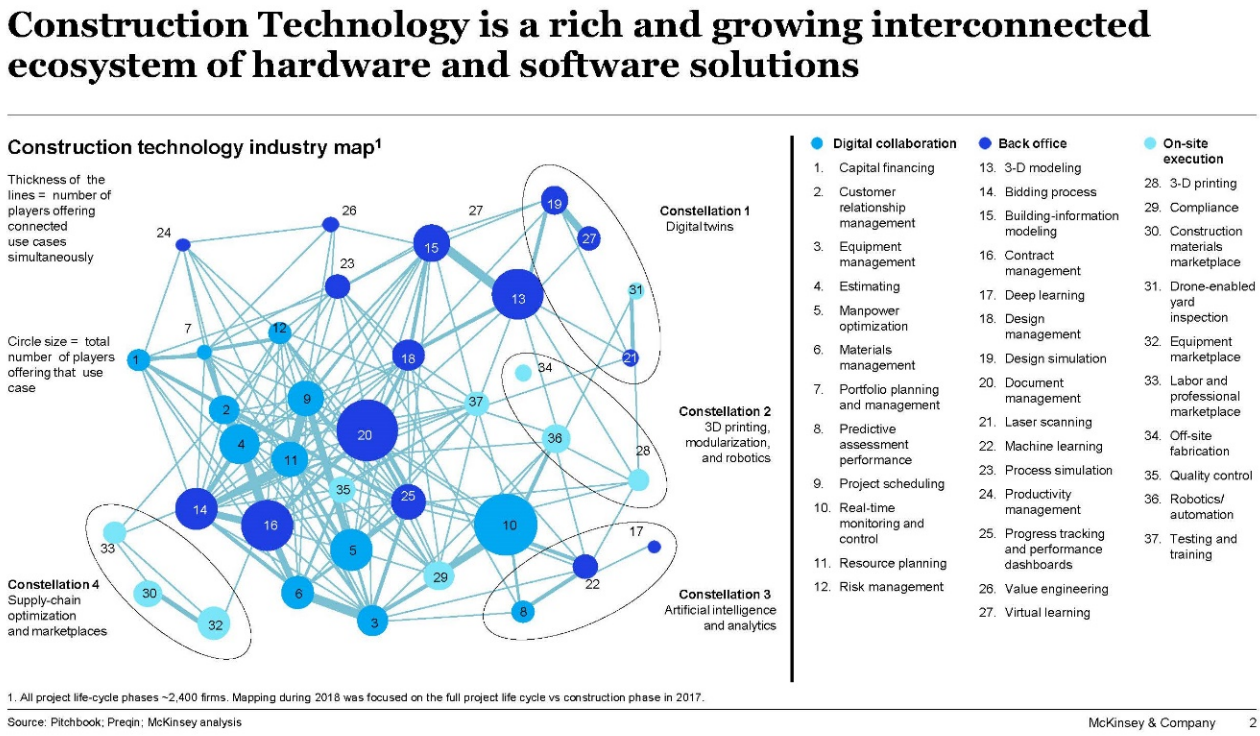


Figure 10 – AEC Industry Technology and AI use cases (McKinsey & Company, 2020)

As it will be shown in the succeeding Literature review, by and large the Design and Construction Management aspect has not been very popular for the use of AI and ML. A personal educated guess is that this area of the AEC industry is managed heavily by heuristics of experienced managers and directors with limited technological knowledge and lack of drive for straying away from known analysis methods. Some aspects of Project

Management Digitalization are included like assigning tasks, maintaining staff data where suggestions for implementing AI use can be beneficial to automate these monotonous processes, reducing mistakes and freeing up critical time resources more easily by optimising workflow and allowing staff to focus on their areas of expertise, boosting total productivity (Srivastava, 2023).

Commercial use of ML to Optimize Cost Estimation

The literature review will begin by covering actual commercial use cases of AI and ML in the AEC industry by industry giants. The first of which, specifically ANNs in a calculation heavy field of Project Management that is optimizing Project Cost Estimating by Pomerleau, a Canadian contractor, and Zetane, an AI development start-up. Zetane Systems collaborated with its client, to utilize AI and NN algorithms to accelerate, reduce risk, and enhance the estimating procedures that can be a big determining factor of the success, or failure of a project (Rathmann, 2022). The collaboration effort already produced some results in a prior phase whereby Zetane provided an AI capable of extracting critical information from unstructured papers presented in a Requests for proposals (RFPs) (Rathmann, 2022).

AI and ML in Commercial Planning Software InEight

The second major commercial use of AI that is specific to the Scheduling and Planning aspect is through the U.S. based AEC Project Management Software Company InEight. The company offers a Scheduling Software that highlights the use of AI and ML to “improve project predictability and performance. Leverage AI to accelerate plan development and quality” (InEight, 2023) utilizing the Historical database of projects for AI recommendations. The main approach utilized is a knowledge-based Database that gets populated overtime when archiving or developing a schedule, assign resources and costs in an unstructured or unlabelled data format (Stewart, 2021). The ML inference engine suggests edits to schedulers that are either accepted or rejected in a feedback loop to the inference engine for better future recommendations. The scheduling software offers integration tools with the most popular scheduling software in the Europe, Middle East, and Africa (EMEA) Region as well.

ORACLE's AI powered Construction Intelligence Cloud Advisor.

The company's VP of Data Science & Analytics (Venkatasubramanian, 2021) describes the solution as working from the new schedule creation to actively seek predicted difficulties based on historical and present situations to estimate the likelihood of delays. By searching for patterns in the current project it attempts to find and link the patterns to what has previously occurred. While constantly checking the project data to enhance predictions as the project continues. The below figure showcases a dashboard example of the tool output to the scheduler.

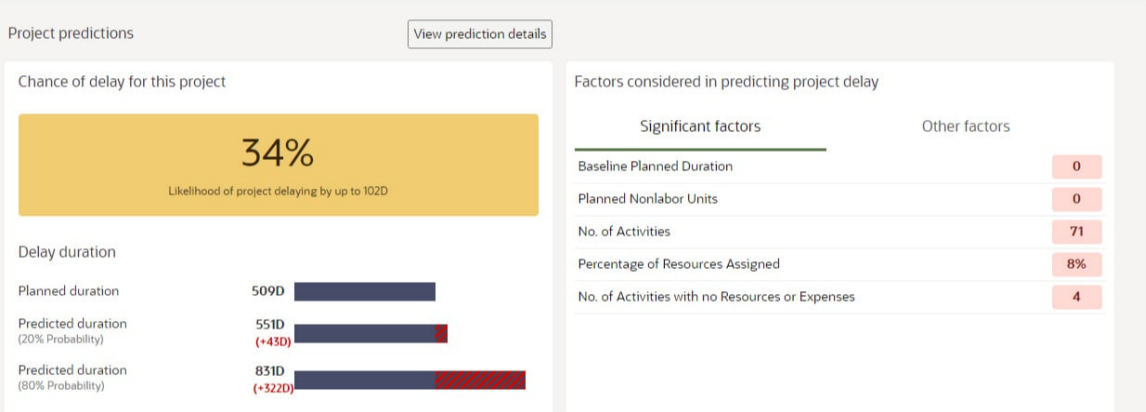


Figure 11 - ORACLE's AI Advisor Example Dashboard (Venkatasubramanian, 2021)

Overview of the Literature in Automated Scheduling until 2014

The Journal paper by (Faghihi et al., 2015) published mid 2015 in the International Journal of Advanced Manufacturing Technology aims to specifically covers the research in Automated Scheduling techniques. It spans the research of 30 years until 2014 and serves to provide valuable insight into focus areas specific to Construction Scheduling. The research covered the following main categories.

1. **Case-based reasoning (CBR)**, defined as a tool using prior knowledge of comparable situations to solve the current. It is considered a subset of KBSs.
2. **Genetic Algorithms (GA)**, defined as search heuristic inspired by the process of natural selection powerful tool for solving optimization problems, particularly in

situations where traditional methods are computationally infeasible or fail to find satisfactory solutions.

3. **Expert Systems,** defined as AI algorithms imitating human experts decision-making ability in a certain field of knowledge for problem solving.
4. **Neural Networks (ANNs),** defined as AI algorithms that mimic the working nature of the human brain. They consist of interconnected nodes, or neurons, that process information and communicate with each other using weighted connections.

A summary is provided in the below figure showing how much each category contributed to the literature.

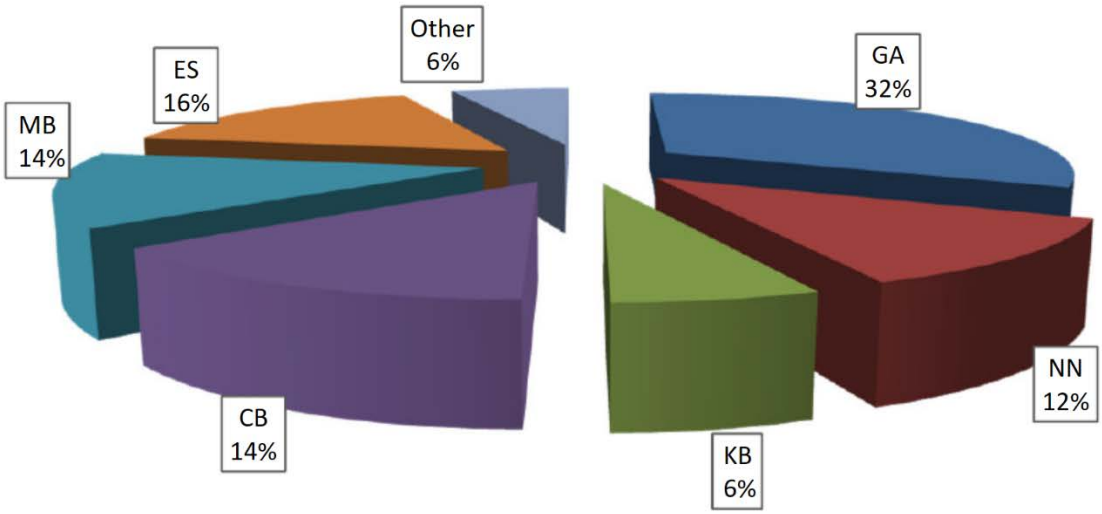


Figure 12 - Automated Scheduling Techniques in the Literature (Faghihi et al., 2015)

AI Evaluation and Prediction of Project success using Limited Datasets

While this research effort by (Bang et al., 2022) is not directly related to the purpose of this Thesis, it serves a very important tie-in due to its focus on two main aspects. The first being its use of limited datasets from the Construction Industry, one of the biggest issues for AI and ML research, discussed in a later section due to data confidentiality. And second is their research outcome highlighting most important project success factors.

Allowing other researchers to use this outcome for selection of the most relevant features to collect as they're assembling their datasets for more analysis of other aspects of AEC Projects (i.e. Risk Analysis, Cost Overruns, or this Thesis's target, Realistic Scheduling metrics).

The researchers define Project Success as several Critical Success Factors (CSFs) based on the Project Triangle of Time, Cost, and Quality. The researchers built their dataset based on the CII Nordic 10-10 database specifically for the Norwegian Construction Industry from participating companies. Afterwards, they utilized data preprocessing techniques to clean up and label the data according to a Target Prediction Value (Project Success) and Project Features upon which Project Success is related. Finally, a Classification Model was built to classify if a new unseen project will be successful or not given Project Features within the Model's Trained dataset. The researchers tested several ML Classifiers and decided on a Classifier Model called Random Forest Classifier (RFC) that utilizes a weaker underlying technique called a Decision Tree Classifier (DTC) in massive amounts all randomly built to classify different combinations of the dataset and their classification outputs are then averaged for the final classification in a binary fashion (meaning that the final Classification is A if Classification A has a larger amount of favouring DTCs than Classification B or vice-versa).

Flexible Schedule Design and Execution supported by AI.

Sponsored by the InnovateUK grant, a collaboration between University of Cambridge's Construction Information Technology (CIT) Lab, nPlan AI Scheduling Software Company, and the British Construction Giant Kier kicked off in the second half of 2019 aimed at developing Optimised Pathways for Scheduling Execution using AI (AI-OPSE). This research project is said to advance project management by creating a novel schedule-learning platform that uses machine learning and data science methods to analyse thousands of historical project schedules. The project should provide a one-of-a-kind & scalable solution for increasing project planning dependability and trust. The total duration of the research is 24 months and is divided into seven sections (Laing O'Rourke Centre for Construction Engineering and Technology, 2019). Unfortunately, the final research paper for this project was not accessible for further analysis.

AI in Construction Asset Management (CAM) Literature overview

The research by (Rampini and Cecconi, 2022) provides the best introduction to this literature review as it provides a very well thought out research methodology about the research trends of AI in Construction in general with a focus on CAM looking over more than 575 papers. In the context of this thesis the highlighted part is table 10 in section 5.4 regarding the use of AI in Project Management processes as shown in the below figure

Table 10: A list of AI-based project management methods.

Year	Application	Algorithms	Asset	Ref.
2016	Automated compliance checking	NLP	Buildings	(Zhang and El-Gohary, 2016, 2017)
2015	Assess and predict construction labour productivity	ANN	Buildings	(Heravi and Eslamdoost, 2015)
2014	Time and cost forecasting	SVR	Buildings	(Wauters and Vanhoucke, 2014)
2015	Predict project award price	ANN	Buildings	(Chou *et al.*, 2015)
2017	Predict construction labour productivity	ANN	Buildings	(El-Gohary, Aziz and Abdel-Khalek, 2017)
2012	Litigation prediction of site condition disputes	SVM	Buildings	(Mahfouz and Kandil, 2012)
2019	Predict time and cost claims in construction projects	ANN	Buildings	(Yousefi *et al.*, 2016)
2017	Bid/no bid decision making	SVM	Buildings	(Sonmez and Sözgen, 2017)
2017	Classification of construction waste material	CNN	Buildings	(Davis *et al.*, 2021)
2021	Forecast material prices	ANN	Buildings	(Mir *et al.*, 2021)

Figure 13 – Literature on AI in AEC Project Management (Rampini and Cecconi, 2022)

Dynamic Process Templates (DPTs) for activities (Amer and Golparvar-Fard, 2021).

Before exploring this research paper, the DPT term should be defined first. In the scientific literature, they're defined as templates that can be used to represent and automate complex processes. DPTs are typically based on a formal model of the process, such as a Petri net or a workflow diagram. They can be used to generate executable code, which can then be used to automate the process. DPTs offer several advantages over traditional approaches to process automation. First, DPTs are more flexible and adaptable. They can be easily modified to accommodate changes in the process or in the environment. Second, DPTs are more efficient. They can generate code that is optimized for the

specific process being automated. Third, DPTs are more reliable. They can be used to generate code that is verified to be correct (Google Bard LLM).

The research paper proposes and develops Dynamic Process Templates (DPTs) for Construction Activities based on a new vector representation in which the scheduling knowledge is represented with a generative Long Short-Term Memory Recurrent Neural Networks (LSTM-RNNs). The Model was tested and verified on a broad dataset of 32 real-world project schedules suggesting that it can learn scheduling and activity relationship modelling knowledge with excellent accuracy across several projects. The approach is utilizing company, and project specific historical scheduling data to learn and identify precedence dependencies. When given an input consisting of several activities in the future the ML model can predict which activities are most probable to succeed. Additionally, it can create a lookahead schedule at any point in the project by considering the logical limitations connected to the outstanding scope of work. Figure (14) below shows the generalized process of the research paper.

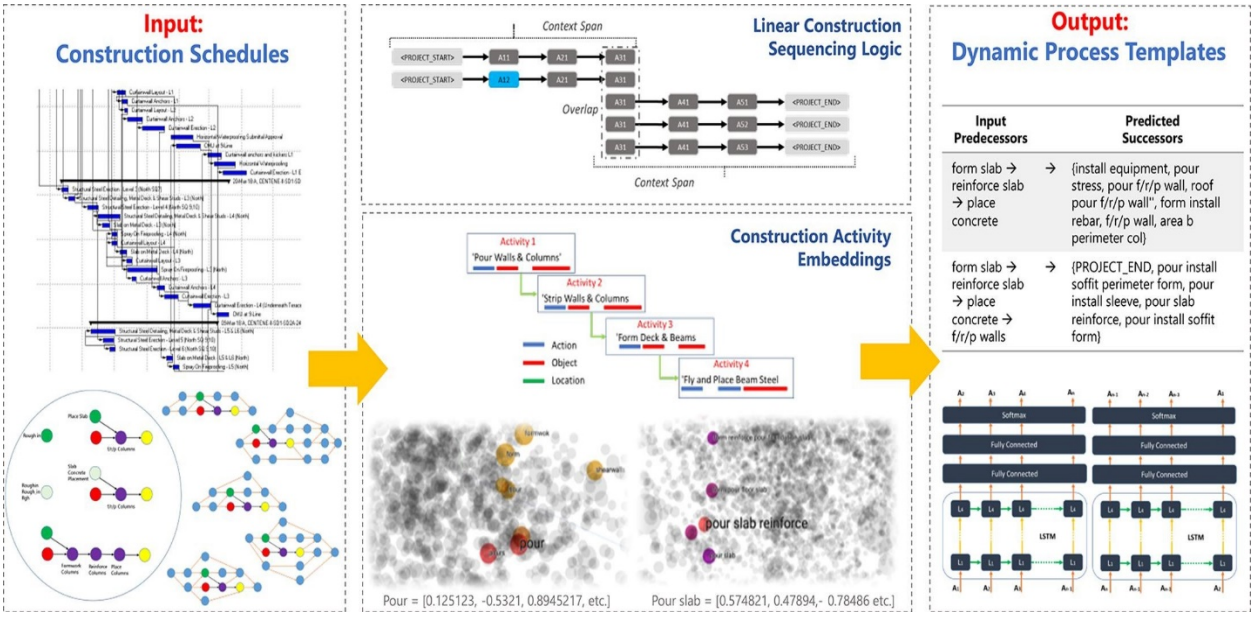


Figure 14 - Generalized overview of DPT approach (Amer and Golparvar-Fard, 2021)

Delay Mitigation in Construction Projects using ML (Sanni-Anibire et al., 2021)

This research paper by (Sanni-Anibire et al., 2021) highlights the use of ML in duration estimation of construction activities alongside other factors to develop a delay risk mitigation framework. The research objective included applying multiple ML algorithms and ensemble approaches to the dataset. Additionally, the final model generated for forecasting duration was based on an ensemble technique using ANN. According to the researcher's conclusions, AEC industry practitioners believed in framework's applicability and appropriateness in reducing construction delays. Nevertheless, its reliability is still rated poorly with about a 25% lower rating than the applicability score.

Using ML for schedule risk analysis

One of the topics in the literature that has a non-direct relationship to the duration optimization problem is schedule risk analysis since it contains the components of time and cost predictions. (Fitzsimmons et al., 2022) mentions two efforts in that area. The first research part trained a series of ANN on the total length and cost of highway projects to discover the essential project characteristics to be utilized in an estimated duration prediction model. The second model developed a more targeted approach also using ANNs to forecast earthworks durations.

ANNs for Understanding Activities in Construction Schedules

A very notable research effort from (Amer and Golparvar-Fard, 2019) that targets the deciphering of Activity meaning from the Construction on site point of view from its name in the Construction Schedule; and be able to proceed accordingly with the required action. The researchers named their method "Part-of-Activity Tagging (POA)" and are the first research effort to develop a Recurrent Neural Network Model (RNN) that uses Bidirectional Long Short-Term Memory (BI-LSTM). The developed model automatically extracts "company-specific construction sequencing knowledge" (Amer and Golparvar-Fard, 2019) from prior schedules and stores it in a dynamic editable database. The model is considered Supervised ML as it must be capable of decoding activities and identifying their elements. It was trained using more than 7000 activities and overall accuracy, named sentence accuracy by the researchers, measured at slightly above 70%. The researchers justified their problem-solving approach as like the one used in Speech and

Natural Language Processing (NLP) called Part of Speech (POS). Which landed them on a widely used NN model in that area of research with known high accuracies as mentioned above. Given the activity name, the model outputs an equivalent Construction Activity with highest likelihood. The below figure shows an example of how the dataset was labeled for the Model to understand.

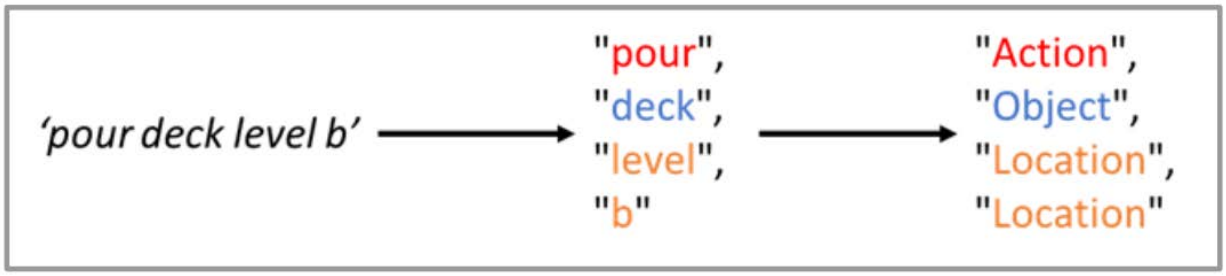


Figure 15 - Dataset Labeling Example (Amer and Golparvar-Fard, 2019)

An ML approach for accurate total duration estimation of High-Rise Buildings

In this research article the researchers attempted to use ML to predict the total construction duration of the high-rise buildings more accurately; citing the fact that total project durations are usually optimistic in nature. It is important to note that the researchers who wrote this paper are the same who wrote the research on Delay Mitigation using ML (Sanni-Anibire et al., 2021). The researchers utilized multiple model types such as Linear Regression, ANNs, SVMs, and k-NNs. The researchers used an open-source ML software developed by University of Waikato in New Zealand. Their included 35 projects completed between 1993 and 2015 obtained from Mega Project Case Study Center of China. It included some Features as follows:

1. Number of Elevators
2. Total Building Area in m^2
3. Floor Built-up Area in m^2
4. Total Number of Floors
5. Number of Floors above Ground
6. Number of Parking Floors
7. Number of Parking Spaces
8. Project Cost in Chinese Yuan
9. Total Actual Project Duration
10. Building Type
11. Structural Material
12. Commencement Season

The researchers proposed a final model using an ANN with a r^2 score of 0.69, Root Mean Squared Error (RMSE) of 301.72, and Mean Absolute Percentage Error (MAPE) of 18%.

This Page was intentionally left blank as a separator between chapters.

Research Methodology

The research methodology for this thesis is split into two main parts, the first part is involved with full analysis of the identified literature on the subject covered in the previous chapter. Afterwards, the feasibility research of the thesis target to provide grounds for the subsequent section justifying the creation of the Algorithm, coding language, and thought process. The main research Repositories for the feasibility research are as follows.

1. Sci-Kit Learn Open-Source Library
2. Hugging-Face Open-Source Transformer Models Library

Research Findings Analysis

The details of the research on Project Success Factors by (Bang et al., 2022) includes more details on their methodology than most others collected in this Literature Review. It is mentioned that they utilized the Python Programming Language to write a Script that handles the Dataset with two very well-known and often used Mathematical Libraries called Pandas and NumPy. While the model itself was built on a massive accessible Open-Source Library for ML Adjustable Pre-Trained Models called Sci-Kit Learn (often shortened and referred to as SKLearn). This level of detail gives credence to the approach used in this Thesis as it relies on the same principles for Dataset handling and Model Development. The success criteria discovered confirm the theoretically recognized necessity of extensive early planning and analysis. A Process Chart on the used methods is shown in the below figure for illustration.

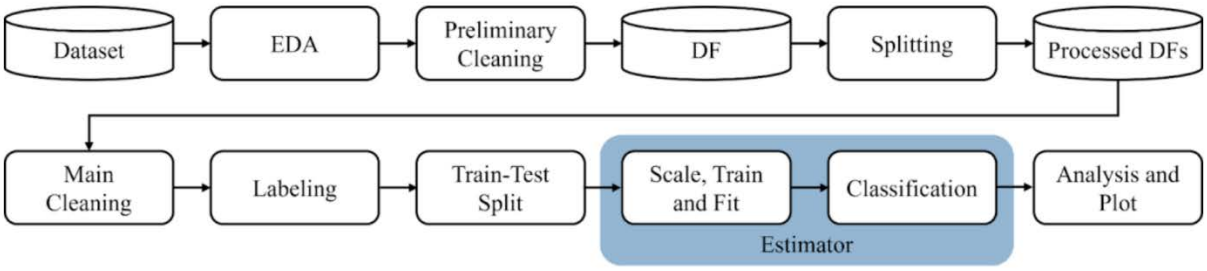


Figure 16 - Steps for Dataset and ML Model Development (Bang et al., 2022)

The most fascinating scheduling optimization technique that does not employ AI techniques is the one by (Alekseytsev and Nadirov, 2022) whereby random interruptions were introduced into critical activities to simulate unforeseen events that are a fact of actual construction projects. However, it is important to note that their research effort indicates that the target is overall construction duration optimization rather than singling out critical activities' minimum and maximum interval bounds. Therefore, this could be an improvement point to their approach. Additionally, their methodology of incorporating GAs can benefit from AI to exterminate non optimal solutions for future iterations.

It can be seen from the literature review above that the use of ML, KBS, and Optimization is mostly attributed to Project Management subfields of the AEC Industry, and specifically resource and mathematically intensive portions of these processes like Optimum Cost (Profit and Overhead) and Time (Scheduling) estimations. The literature review by (Abioye et al., 2021) on the research trends of AI in the AEC Industry shows that Optimization has always been the focus of research studies in the integration of AI. Optimization problems were defined by in the research as "a construct of the problem of making the best choice from a set of choices. Optimization is a lifelong phenomenon, originally known as a mathematical discipline concerned with finding an optimal solution to any given problem". The researchers attribute this focus to the long-standing battle with overall poor productivity levels. Another finding from study developments over the past few decades is that machine learning has surpassed knowledge-based systems (KBSs) as a sector that is important in the construction industry in the recent decade. The Author's educated guess for that is the exponential rise in computational ability of hardware, the shifting focus on AI and ML specific hardware development. And the viability of studying massive amounts of data for patterns and predictions rather than depending on human knowledge stored in a KBS.

(Van and Quoc, 2021) tabulated the most productive countries' contribution with the US leading the research front, China, and Australia as close second and respectively. Research on ML in AEC Management processes is being conducted in around one third of the world's countries.

Country	Documents	Citations	Ave. Citations	Total link strength
United States	35	846	24.17	6038
China	30	606	20.20	4225
Australia	15	357	23.80	4412
Hong Kong	12	263	21.92	3496
United Kingdom	10	213	21.30	3472
Canada	6	244	40.67	1925
South Korea	6	262	43.67	1580
Taiwan	5	204	40.80	649
Germany	4	134	33.50	2088
Viet Nam	4	66	16.50	870

Figure 17 - ML Research for AEC Management top countries (Van and Quoc, 2021)

As it can be seen from the literature overview by (Rampini and Cecconi, 2022), several researchers concluded the ability of AI to provide Time prediction for Construction activities such as (Wauters and Vanhoucke, 2014) and (Yousefi et al., 2016). In addition to tangential topics related to duration estimation like worker productivity (Heravi and Eslamdoost, 2015). This literature overview lends more credibility to the thesis focus. It can be inferred from the above literature review that there are mainly two objectives for research surrounding Planning and Scheduling inadequacies. The first one is related to the Scheduling Logic, how the activities are sequenced together within the framework of the execution, contractual, and governmental requirements. While the other objective is related to more effective calculation and estimation of activity durations through labour productivity rates. Additionally, it has been shown that there multiple technologically aided approaches with or without use of AI, Computer Vision, and ML capabilities.

One of the more important comparisons to draw from the Literature Review is the comparison with using AI in Project Estimation, as put by Zetane CEO Guillaume Herve “The labour-intensive nature of the estimating process and the risk that comes from the fact that estimators can still miss critical elements were both problems the technology

project was designed to address" (Rathmann, 2022). This quote can also be directly translated to Project Scheduling, overall planning process, and the associated risks. To provide more emphasis, In the case of Cost Estimating using traditional, even if technologically enhanced, methods is that there is no data reuse. After a bid is submitted there is no knowledge gain from it or previous estimates. Therefore, breaking the connection between the estimation provided in the bid and the actual project if won and negating any advantage from preconstruction effort on the project's execution phase. (Rathmann, 2022). This succeeding quote also reinforces the importance of utilizing AI tools "Just as construction professionals learn through experience, AI learns through analysing past data and looking for patterns" (For Construction Pros, 2019).

It is very apparent that unrealistic project scheduling is repeatedly cited as a significant factor to project delays. It is likely to happen during execution from schedule crashing due because of activity delays and may generate regular interruptions in site management owing to tool, equipment, and material supply delays. However, it is most likely to stem from an inadequate initial time schedule as well. Nevertheless, one of the main solutions to address both problems is highlighting the inefficiencies and targeted adjustments in due time. In the case of this master thesis focus it is using realistic activity durations owing to historical data trends. Rob Bryant, EVP of InEight Asia Pacific Branch was interviewed about how the advancement will provide a serious push forward in Scheduling and planning that is summarised in the following points (Heaton, 2022).

1. The greatest desire for utilizing AI and ML is in Project Scheduling.
2. By automating more manual tasks, you may save time on budget and schedule preparation.
3. Allowing project teams to create better, more realistic budgets and timetables based on previous project knowledge and outcomes. This contains outcomes from projects completed on time and, or under budget, or the opposite.
4. Utilizing data across the company rather of relying on individual management experience.

5. Analysis of learned lessons and combining it onto upcoming schedules and budget for educated future decisions and forward-thinking decision making.

Although the research paper by (Amer and Golparvar-Fard, 2021) aims to produce a similar output, which is embedding the knowledge of planning in a machine learning algorithm for future utilization and progression in Project Scheduling, the full research paper was not openly accessible to read. Nevertheless, there were enough openly accessible sections to gather a comprehensive idea about the approach which is considered more complex than the one attempted in this thesis. However, it seems that the overall goal of the algorithm is to learn a logical representation of a schedule rather than optimize durations. The embedding of Activity Sequencing in an ML model is mentioned in some additional detail in the Future Research Recommendations section.

Schedule Risk analysis using ML by (Fitzsimmons et al., 2022) highly correlates to the goal of this Thesis with a few caveats and differences. In the research part of the risk analysis is determining the optimum realistic duration during the initial planning phase to reduce deviations in the overall project duration or highly important milestones. However, the approach used takes into consideration the overall connections of the schedule even if the output is associated with each activity. The approach used in this Thesis, which is covered in a later section is the isolation of each activity and corresponding features to predict optimal realistic durations. Nevertheless, this does not mean that correlations with other activities are not considered. The dataset includes multiple features from surrounding activities that the ML algorithm uses. Thereby allowing the algorithm to learn and predict each activity specifically.

Automated Scheduling Techniques Analysis

In the Literature overview by (Faghihi et al., 2015) a noteworthy example utilizing the KBS-CBR technique that shares a similar vision to this thesis was the paper by another researcher (Benjamin et al., 1990) that suggested a prototype for construction project planning and scheduling. intended to generate schedules as well as increase the productivity of unskilled schedulers. Another notable example is a commercialization of a product called CaBMA (Case-Based Project Management Assistant) developed by (Xu &

Muoz-Avila, 2004) as a CBR solution add-in for Microsoft Project. To detect instances from current schedules and utilise previously collected cases to build a new plan and maintain the overall consistency of a new schedule.

The research paper by (Bhatia et al., 2022) is also very important to highlight due to its approach being eerily like ML in terms of the regression model development. The researchers divided their dataset into “Training” and “Testing” datasets that are iterated upon until a suitable model is found. Model Data Overfitting is also accounted for utilizing K-Fold Cross-Validation, which will be covered in a later section. The data analysis flowchart is shown below.

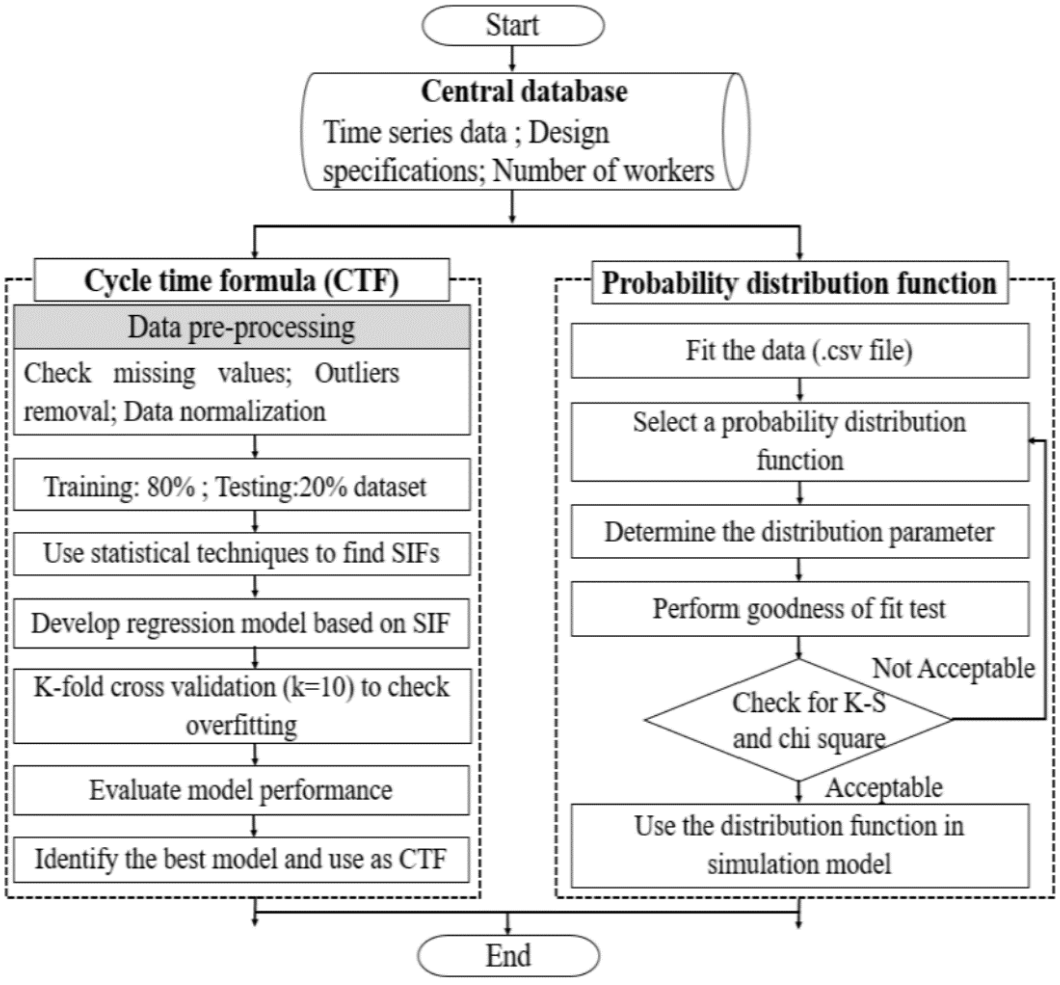


Figure 18 - Data Analysis Flowchart (Bhatia et al., 2022)

The most attracting part in their research is the paragraph about avoiding ML techniques. “To avoid overfitting, machine learning algorithms such as random forest and decision tree, as alternative methods for variable selection, are not considered because of the small size of the dataset, i.e., less than 1,000 observations. According to Makridakis et al. (2018), the use of such methods for small datasets can yield a “black box solution”, which may not be acceptable to industry practitioners”. This part is contested since a similar regression model approach is considered “Machine Learning” as explained in the Definitions section above. Nevertheless, this research paper is the closest to the target outputs of this master’s thesis say for the different industry target of design and construction execution planning compared to manufacturing.

Evaluation of the ML algorithm output is a very important task to do correctly to select the most appropriate model for a given dataset. In the research paper by (Sanni-Anibire et al., 2021) the researchers utilized Root Mean Squared Error (RMSE), Correlation Coefficient (R^2), and Mean Absolute Percentage Error (MAPE) to evaluate the regression model performance for delay cost and duration estimation problems. This research paper also corroborates the conclusions by (Bhatia et al., 2022) in that ML in Construction is not highly regarded in terms of reliability of outputs by Industry Professionals. An educated personal guess on the reasons for that would be as follows:

1. Widespread usage of ML in Construction Management is still in its infancy.
2. There is little confidence in the dataset accuracy as it mainly depends on human collection of historical data.
3. The usage of ML Algorithms for Management processes can involve a huge financial risk which requires intensive manual study of the output.
4. Heuristics of AEC Project Management techniques is widely sought after.

In closing, the following Strength, Weakness, Opportunities, and Threats (SWOT) diagram by (Abioye et al., 2021) is presented below as a summary of all AI trends covered in the literature review.

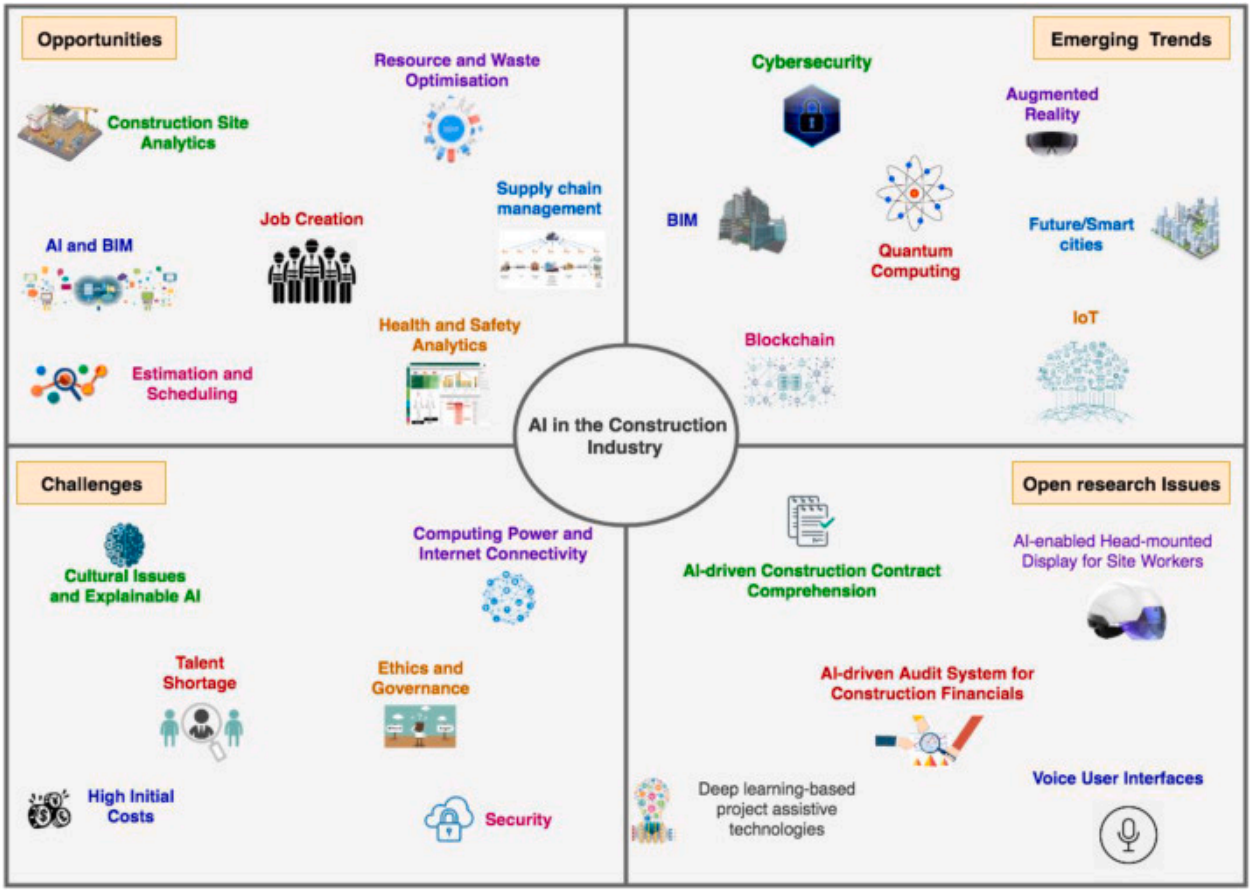


Figure 19 - SWOT Analysis of AI in the AEC Industry (Abioye et al., 2021)

A final note to touch on is deciphering the activities in the Construction Schedules by (Amer and Golparvar-Fard, 2019) as they outlined that one of the benefits of such approach is to ease the dataset building process for other research efforts in the AEC Industry relieving the strain of manually compiled knowledge datasets. This is further enhanced by (Venkatasubramanian, 2021) part on enhancement of these AI systems over time as companies start to implement standardized procedures combining different systems. Thus enabling, more consolidated data sets for a more comprehensive context for generating predictions. Finally, gaining the ability to utilise this technology is preferable in the long term since AI and ML develop and improve over time (Venkatasubramanian, 2021).

This Page was intentionally left blank as a separator between chapters.

Models and Synthetic Datasets

With the aim of correctly approaching the problem it is first important to define a methodological plan. The general outline approach for this plan was taken from the Code Academy website (Code Academy Team, n.d.) that provides multiple courses, skill and career path modules for Computer Science and AI. It includes 6 main steps for the ML Process, these steps are as follows.

1. Formulating the problem statement.
2. Collection of relevant data
3. Data pre-processing and feature engineering
4. Selection of appropriate ML Model(s)
5. Model Testing, Tuning, and Refinement
6. Finalizing Model for Prediction

Aligning the Problem Statement to ML Context

A generalized problem statement was introduced in the final section of the Introductory chapter titled “Objective and Vision”. Nevertheless, this chapter is the appropriate time to narrow it down in terms of specific ML context. The problem statement is as follows “Newly created AEC schedules can often suffer from being unrealistic due to the ever-changing nature of AEC industry during project execution. It is direly needed to improve their reliability and accuracy utilizing historical data and relevant features”.

Desired Final Outcome

The desired outcome of this Thesis is to develop, and refine the necessary ML Models, preferably with a user facing Graphical User Interface (GUI) that comprises a complete framework capable of realistic predictions for all stages of schedule generation from Activity Names, Activity Relationships and Linking, to Activity durations through studying relevant historical data to help Schedulers ensure that their new schedules are more reliable In addition to that, the development of all Datasets as Synthetic Data carved from realistic, and real-world data that is augmented based on the Author’s Real-World Experience to introduce randomness to the model and ensure Generalized Models Performance. Additionally, the model should have commercial viability in mind through

being expandable, adaptable in ability to export to a standalone application (web-based GUI or Executable) and finally, scalable to other ML models.

Separation of the Problem Components

To effectively tackle the problem statement given above it will be separated into 5 parts:

1. **Model 1,** Prediction of Most Probable Task List given preliminary Project Data
2. **Model 2,** Prediction of Activity Relationships and Link Types
3. **Model 3,** Optimization of Activity Durations
4. **Datasets,** Development of the Synthetic Datasets for each Model
5. **Customer Facing Wrapper,** Development of the GUI

Justification of Coding Language

The language of choice is justified during to the reasons outlined in this section. There are typically two main Languages for beginner type data analysts to delve into ML. Python or R. The coding language of choice is Python. The first reason being research done on a personal level through asking experts online and personal contacts as well. Since the author has basic coding knowledge from the BSc. time (Microsoft Visual Basic C++ and MATLAB) the choice was made based on the difficulty of the language, existing support, teaching material online, and availability of open-source libraries. This choice was further consolidated during the learning process while taking advantage of Code Academy's excellent beginner material on Data Analysis and ML that ended up using Python for their use of Sci-Kit Learn Open-Source Library. The total time spent learning was less than 6 months.

Model 3: Optimization of Activity Durations

Proposed Algorithm Flowcharts for Optimization of Activity Durations

The first model that will be covered is Model 3 as it contained the most amount of personalization and decision making. Firstly, an overview of the Model working through 3 proposed flowcharts for the whole system from start to finish as it is split in to major phases.

The first phase is a flowchart utilizing a simple ML model for Dataset Feature Selection and reducing complexity. This ML Model is using Lasso Regression for most relevant

features selection. It is worth noting that justification for that selection is mentioned in a later section concerning Model Training and Refining. Once the Dataset Feature Selection is complete, the second flowchart showcases the actual ML Process whereby the dataset is loaded, pre-processed as needed, and then used for Model training of different models. The most suitable model is then singled out and saved. Finally, the last flowchart is related to the Prediction of unseen variables. Typically referred to as Model Inference. The Inference dataset and trained model are imported, then the model uses the new dataset for prediction of Construction Activities' most probable Real Durations. The 3 flowcharts are shown below consequently. The second and third flowcharts include the same approaches used in Prediction of Activity Relationships and Most Probable Activities given Project Data

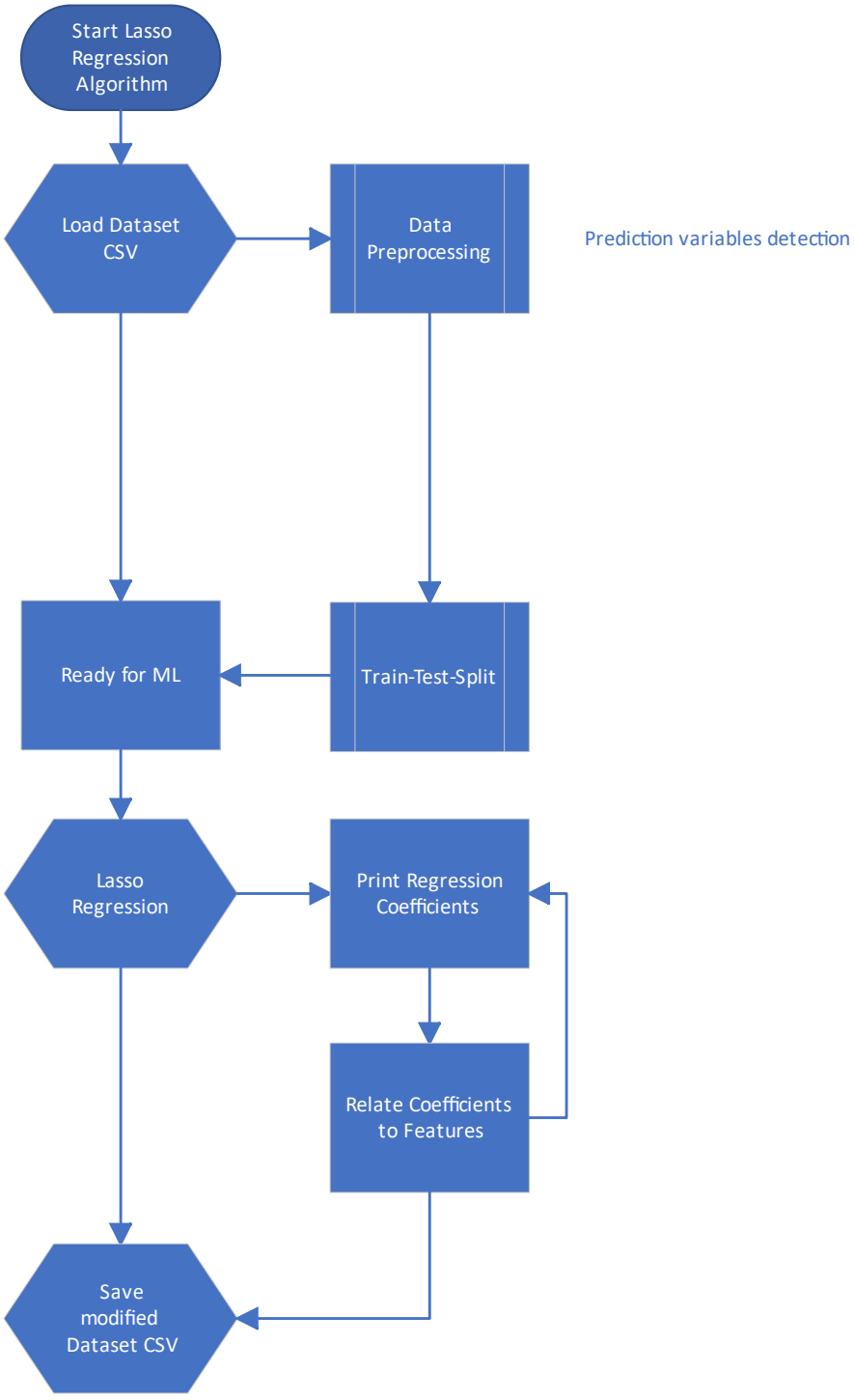


Figure 20 - Lasso Regression for Feature Reduction Flowchart

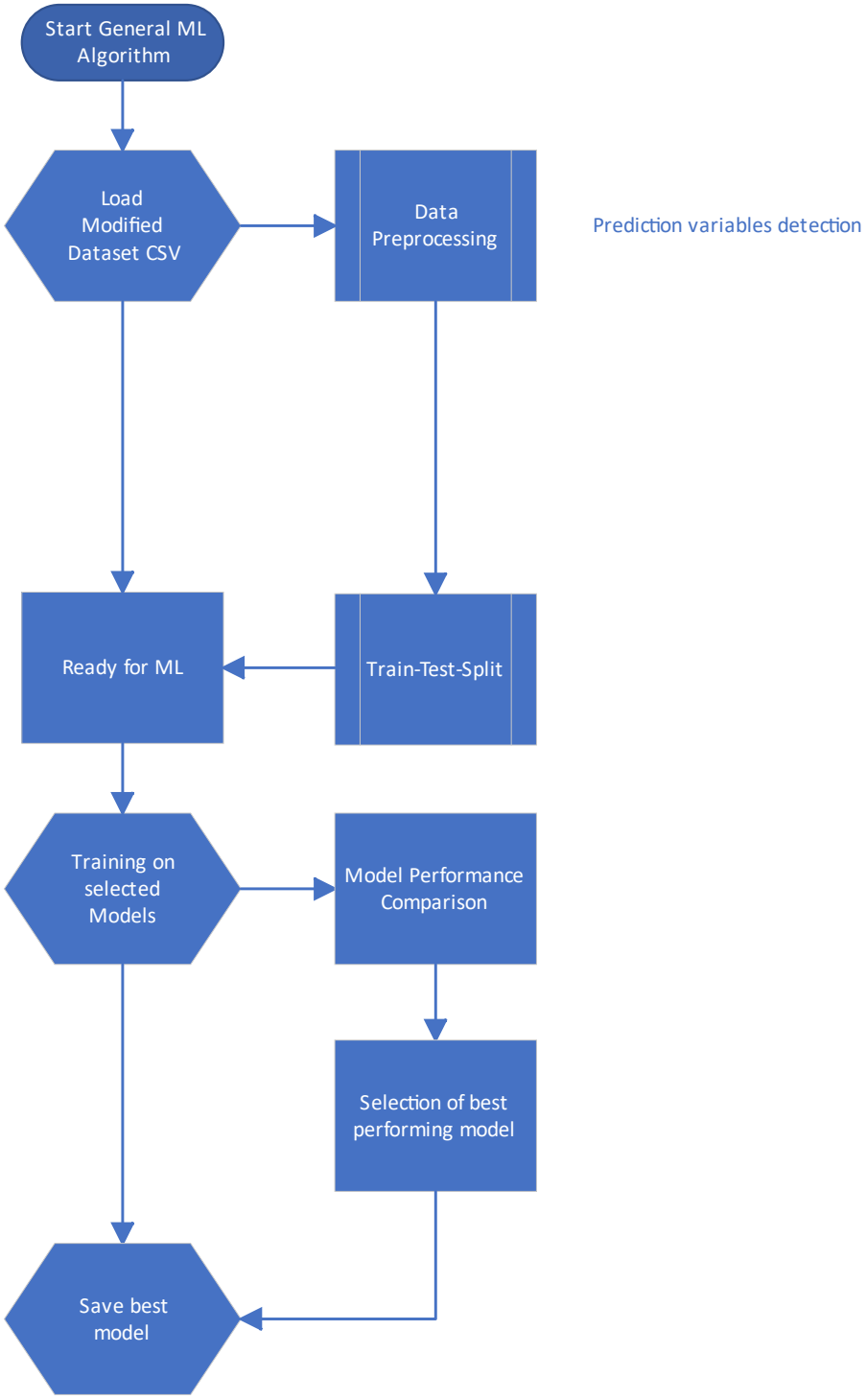


Figure 21 – General ML Process Flowchart

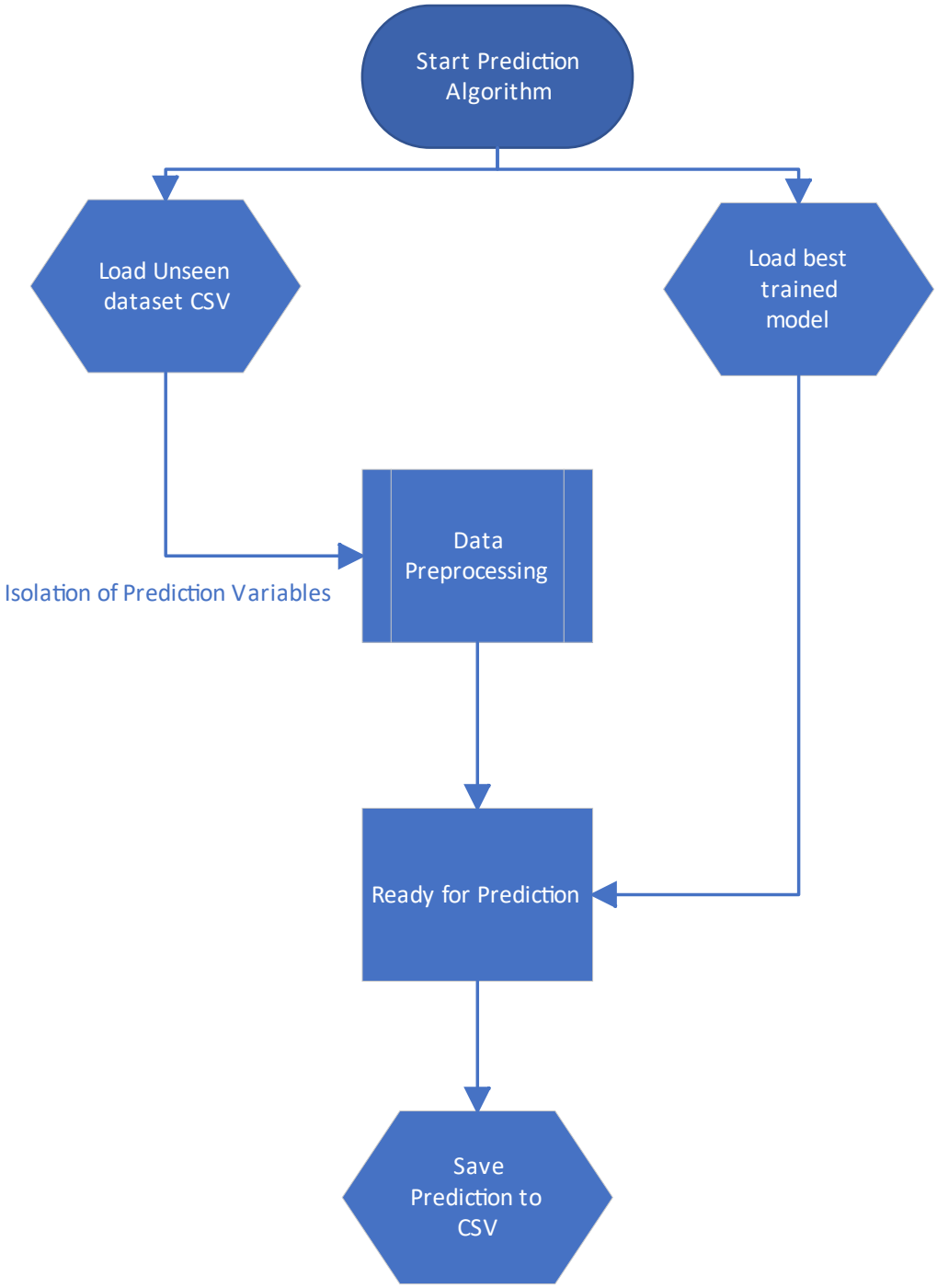


Figure 22 - Prediction of new unseen data Flowchart.

Aligning the Durations Model to Sci-Kit Learn

Before going into Python code specifics, a small refresher on how classical ML works with a specific relation to the Sci-Kit Learn library for Regression based problems. First, the definition of Regression is tackled as an algorithm or approach studying and measuring relationship across two or more variables using samples collected in a dataset (Dodge, 2008). The predicted variable is known as dependent, often represented by Y; while X are known as independent variables that explain fluctuations in Y. The result is an equation that calculates the value of the dependant variable based on the values of the independent variables.

As outlined in a previous chapter, ML hides a subset of the entire dataset for testing and validation. After fitting the Regression Equation and validating the output of the Model, the dataset is reshuffled, and another random subset is hidden. The model is re-evaluated on what is considered a new dataset and compared to the previous model. The evaluation of different models is what classifies this process as “Learning” and is known as called **Cross-Validation (CV) Scoring**. This loop is repeated a certain number of times as specified, a certain Model Score is reached, or begins to decline even. The final Regression model is the one with the best score. Additionally, a lot of Classes used employ regularization techniques to prevent overfitting the model to a specific dataset. Regularization is a penalizing term added to the objective function during training. Finally, the number of iterations over the dataset usually set by Sci-Kit learn in a parameter called “max_iter” or “n_iters” and defines the maximum number of epochs, where each epoch corresponds to one complete pass through the training dataset. It can vary from model to model between 100 or 1000, or of course user defined. It is now clearly explained the inner workings of Sci-Kit Learn ML Algorithms Using the above alignment, it is good practice to provide a process flow for better understanding using for example a Regression Class with Stochastic Gradient Descent (SGD) as the optimization technique for finding optimal coefficients in which a loss function gradient is measured over iterations on the dataset to determine model convergence.

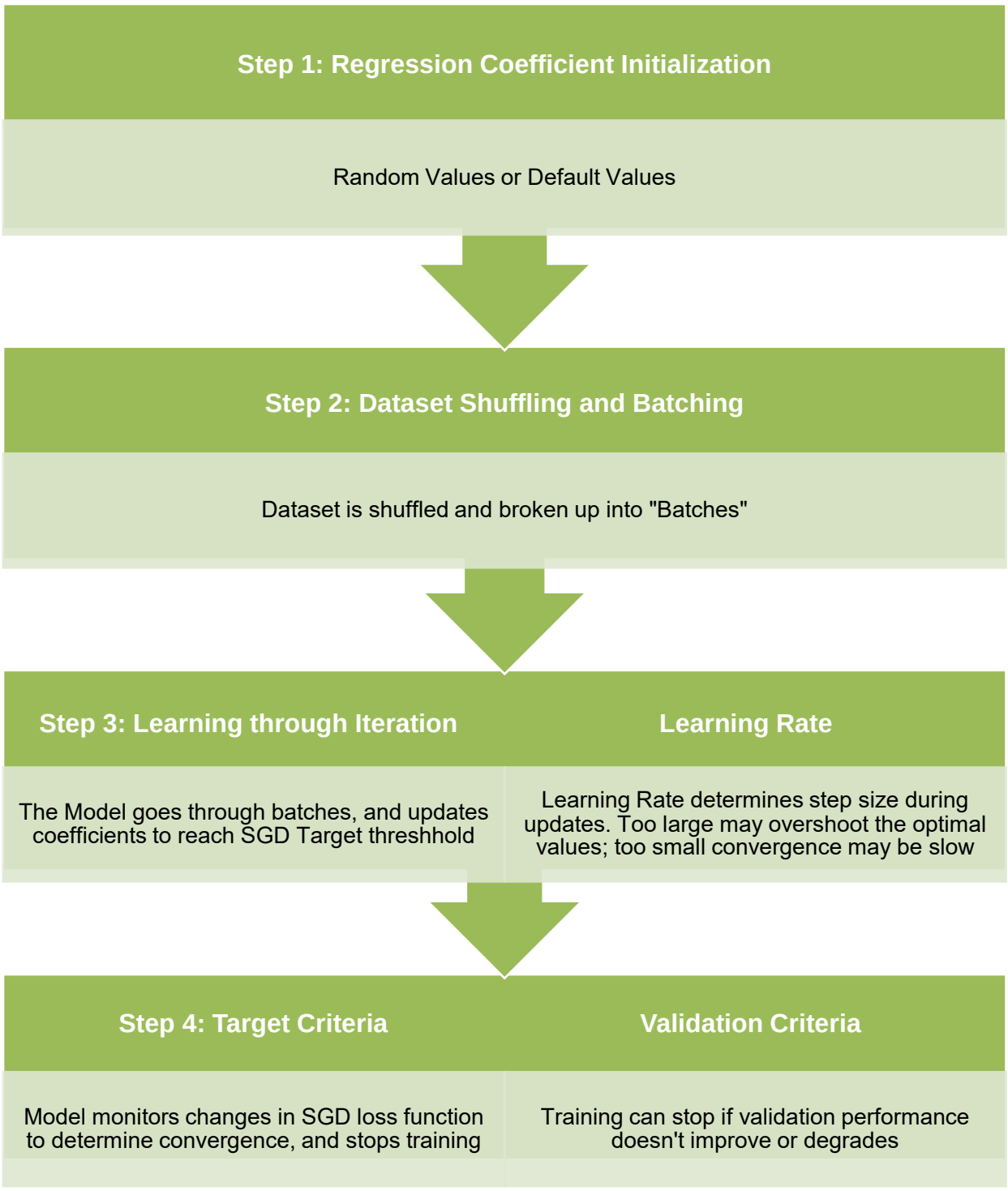


Figure 23 - Scikit Regression ML Flow

Model 3 Code Breakdown Structure

Development File 1: Lasso Regression Model for Feature Selection

Phase 1: Import and Prepare

Importing of all needed external libraries into the development file.

Phase 2: User Interface

Coding the Graphical User Interface (GUI) to load the Dataset and associated activity names applying the help of LLMs for expedited development.

Phase 3: Data Pre-processing

The user gets a print output of the inputted activities, and a .txt file for the target activities that the model is going to be trained on

Phase 4: Fitting the Model, and Feature Selection

Final Phase for this file. Using the dataset and targets for ML Model Training to identify relevant features and save them to an output file for further analysis.

Development File 2: Full ML Algorithm

Phase 1: Import and Prepare

Importing of all needed external libraries into the development file.

Phase 2: User Interface

Coding the GUI for importing the Schedules Database and Construction Activities

Phase 3: Data Pre-processing

The user gets a print output of the inputted activities, and a .txt file for the target activities that the model is going to be trained on. And the data gets pre-processed for training.

Phase 4: Fitting all Models, and Best Model selection.

Using the dataset and targets for ML Model Training to identify relevant features and save them to an output file for further analysis.

Phase 5: Fitting best performing model to Dataset.

Now that the highest performing Pipeline and corresponding model are identified through the weight criteria, they are used to fit the dataset one final time.

Phase 6: Visualization Curves and Output data

This is the final phase for this file, it is used to print out Data Visualization graphs such as Learning Curve, Prediction Error Plot Points. In addition to saving metrics like total Algorithm Run Time, Final Trained Model, and Model performance using the hidden part of the dataset to output files.

Model 3 Training, Testing and Refining

Data Collection

To verify the model's performance suitable data collection must be present and cleaned up according to the Sci-Kit criteria for the ML process. A brief explanation from the literature on how this process occurs is in order first. For example, an ML algorithm meant to forecast the possibility a contractor bidding on a project need only rely on past data from historical bid decisions without the need of the Contractor's physical presence (Akinosho et al., 2020). The data structure consists of an output column (Y), or multiples (Y_1, Y_2, …Y_n) against all collected data on the sample (project) as input columns (X) typically referred to as **features**. The number of required features and samples for acceptable model performance depends on the AI approach.

The dataset used for the development and testing of the Model is utilizing a synthetic dataset for 1000 (one thousand) projects based on the author's experience of approximate real-world quantities and pricing from the Construction Market. The full dataset is provided as an appendix to this Thesis.

It is important to highlight a few factors about the used dataset. The synthetic dataset is only built for continued progress in developing and refining the ML Model performance. This includes introducing randomized missing values to assess prediction accuracy changes and is highly beneficial for further comparison about scalability of Features versus Sample size as well. However, the main goal of this dataset is internal development. The real-world dataset is relatively modest, highly dimensional, and lacks a lot of values due to the difficulty and confidentiality of acquiring said information.

Synthetic Database Breakdown

As mentioned above, the Synthetic Database consists of 1000 schedules. It is aimed at mimicking a traditional Line of Balance (LOB) Activities and works of typical Building Projects. The dataset template was utilized from one of the Author's Graduate Course Projects in which the Author's Group had to construct a time schedule utilizing the LOB Method for a Residential Medium Sized Building. A total of 18 activities selected for this database as follows:

1. Project Start Mobilization
2. Earth Works
3. Concrete Skeleton Works
4. Plumbing Water Supply First Fix
5. Plumbing Drainage First Fix
6. Exterior Walls Installation
7. Exterior Plastering Works
8. Waterproofing for Wet Areas
9. Electrical Wiring First Fix
10. Ceramic Tile Works
11. Doors and Windows Installation
12. Elevator Installation
13. Interior Walls Installation
14. Interior Plastering Works
15. Painting Works
16. Electrical Wiring Second Fix
17. Plumbing Second Fix
18. Heating System Second Fix

Each of the above activities has several associated features. The number of chosen features represents the most amount of data that can be known during the planning phase. These features are as follows:

1. Project Location
2. Project Type
3. Year
4. Season
5. Contract Value
6. Soil Type
7. Floor Built Up Area (BUA)
8. Planned Duration (per target activity)
9. Normalized Quantity Difference (median for each 100 samples then 10 median average statistical analysis, per target activity)

The numbers inside each project were calculated using Microsoft Excel Random Number Generator (RNG) clamped to upper and lower duration boundaries. For example, the Excavation activity was calculated using the following:

```
randbetween(5,10)
```

Indicating that the duration is randomly selected between 5 and 10 days. Additionally, some of the features like project location were calculated using more sophisticated formulas as follows:

```
CHOOSE(RANDBETWEEN(1, 4),
"Berlin","Hamburg","Frankfurt","Munich")
```

Indicating that the random selection happens between the numbers 1, 2, 3, 4 with each number representing the corresponding location. Finally, to keep the Synthetic Dataset as realistic as possible. Some features were calculated with a dependency on others. For example, the Planned Duration for the Plain Concrete Footings was calculated as follows:

```
=IF(I3="Residential",RANDBETWEEN(3,6),IF(I3="Mixed
Use",RANDBETWEEN(10,15),IF(I3="Office
Building",RANDBETWEEN(10,20),IF(I3="Commercial",RANDBETWEEN(20,3
5),))))
```

Indicating that the duration for this activity is dependent on the Project Type. Additionally, to account for missing data effect, an excel command was used to randomly select cells accounting to a total of 10% of the dataset and replacing them with “ ” indicating a missing value for the algorithm. More than one dataset with different amounts of missing data percentages were made (from 20 to 70%). This formula is broken down as follows.

```
=IF(RAND() < 0.1, "", B2)
```

Where:

- RAND() < 0.1 Indicates probability of a selected cell is randomly less than 0.1
- " " replace value with nothing.
- B2 - Target Column (for example Cell B Row 2)

The amount of missing data will be compared in a later section to analyse its effect on model performance.

Feature Selection with Lasso Regression

A Simplified ML model was developed utilizing only Lasso Regression for that purpose. The information for this model was gained from Sci-Kit Learn Website page titled "1.13.4.1. L1-based feature selection" (scikit-learn, n.d.); and further corroborated in (Tajziyehchi, 2021) master's Thesis whereby the author applied LASSO regression to their dataset decreasing the feature count by more than 90% (from 281 to 21). The reasoning behind this is it's a regularised regression approach that reduces coefficients to zero, allowing it to choose just the most important characteristics (Google Bard LLM). The coefficients penalization term is controlled by the variable alpha "Constant that multiplies the L1 term, controlling regularization strength. alpha must be a non-negative float i.e. in [0, inf)" (scikit-learn, n.d.) whereby a value of "**alpha = 0**" provides no penalty for Coefficients making it effectively ordinary linear regression; and a larger value forces a sparser matrix of coefficient values that shrinks more coefficients towards 0 allowing for only the most influential features to be exposed. A range of alpha values was used to see the effect on Feature Set coefficients as follows.

```
Alpha = [0.01, 0.1, 1, 10]
```

Comparison of Horizontal or Vertical Expansion

This section aims to compare the model performance versus Horizontal (more features) and Vertical (more samples/projects) Expansion of the dataset. The horizontal expansion of the dataset. Horizontal Expansion is already handled using the Lasso Regression Step for extraction of most important features to be used for the main algorithm.

For this comparison it is worth noting two important things

1. The alpha value was fixed at **alpha = 0.01** as to allow all coefficients to participate in the fitting process for the most optimal fit. A comparison of the alpha values is considered in a following section.
2. Vertical Expansion comparison fixes the model at **80% Training and 20% Testing Split**. The comparison data is outlined in the below table.

Samples	Fit Score	Test Score	% Difference	Iterations Performed
25	0.9999	-0.7470	174.7	7,478
50	0.9998	0.7952	20.4	10,517
100	0.9996	0.73397	26.6	50,581
200	0.9986	0.6899	30.9	33,720
500	0.9897	0.7658	22.6	109,655
1000	0.9580	0.8363	12.7	140,019

Table 1 - LASSO Regression Fit versus Test Scores in Relation to Sample Sizes

As can be seen from the above table, the bigger the sample size, the more iterations are needed to fit the data perfectly. Plotting the curve of Sample size versus Iterations seems to indicate a Logarithmic Relationship for the Iterations needed to Fit the Model as follows.

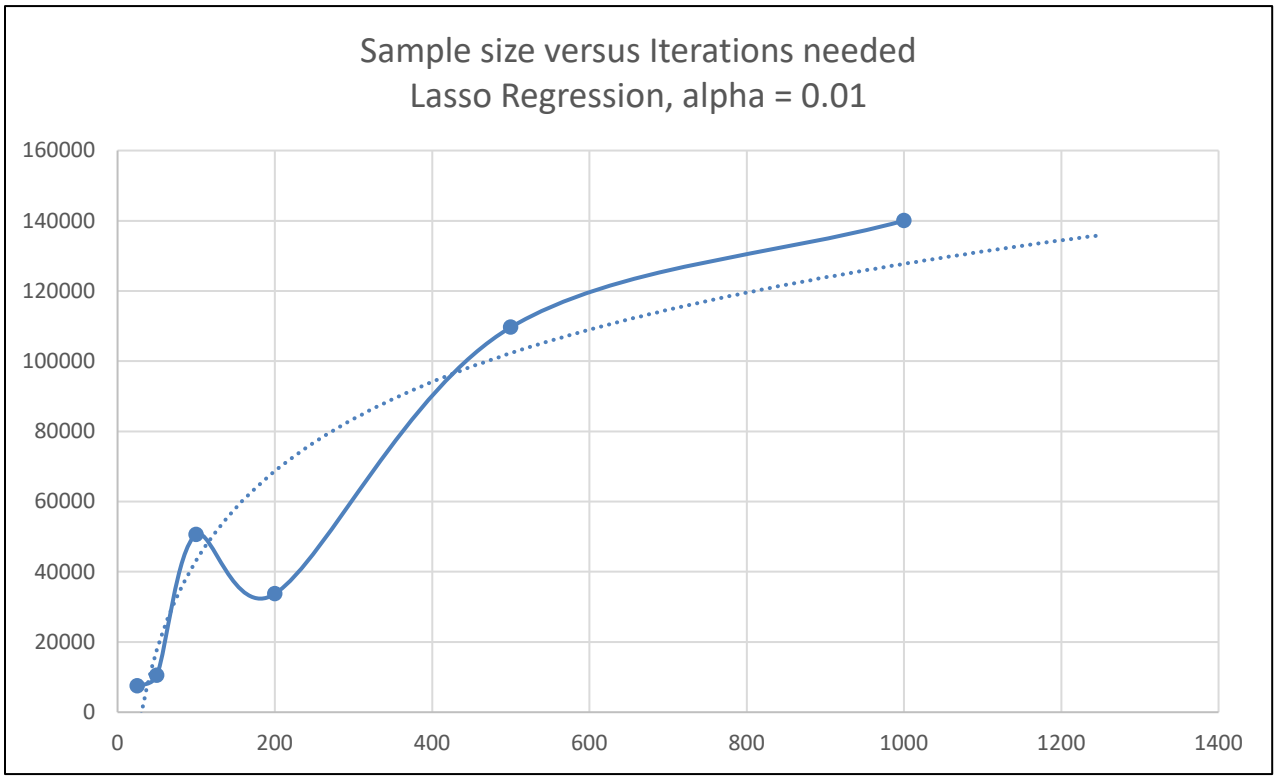


Figure 24 - Sample size versus Iterations needed for Lasso Regression at alpha = 0.01

Figure 25 – Graph view of Table 2

In addition to the Number of Iterations the model performance against the sample size was also measured as a preliminary assessment for the full model performance. The green line in the graph above represents the secondary vertical axis of the % difference between the model Fit Score on the Training Set, and the Model Test Score on the Testing Set. From around 50 samples or larger the percentage difference seems to favour the larger sample set for better test performance indicating a good, generalized model performance.

Feature Selection, and Model performance with different Alpha Values

This comparison focuses on using the full synthetic dataset of 1000 samples and changing the value of alpha to assess model performance and the most important identified features. The data is presented in the tables below as well as graphically where possible for better visualization. As a kind reminder, a bigger value of alpha forces the coefficients closer to zero allowing for a sparser matrix of coefficients.

Alpha	Fit Score	Test Score	% Difference	Iterations Performed
0.01	0.9580	0.8363	12.7	140,019
0.1	0.8918	0.8903	0.2	1,125
1	0.8463	0.8588	-1.5	790
10	0.6514	0.6692	-2.7	393

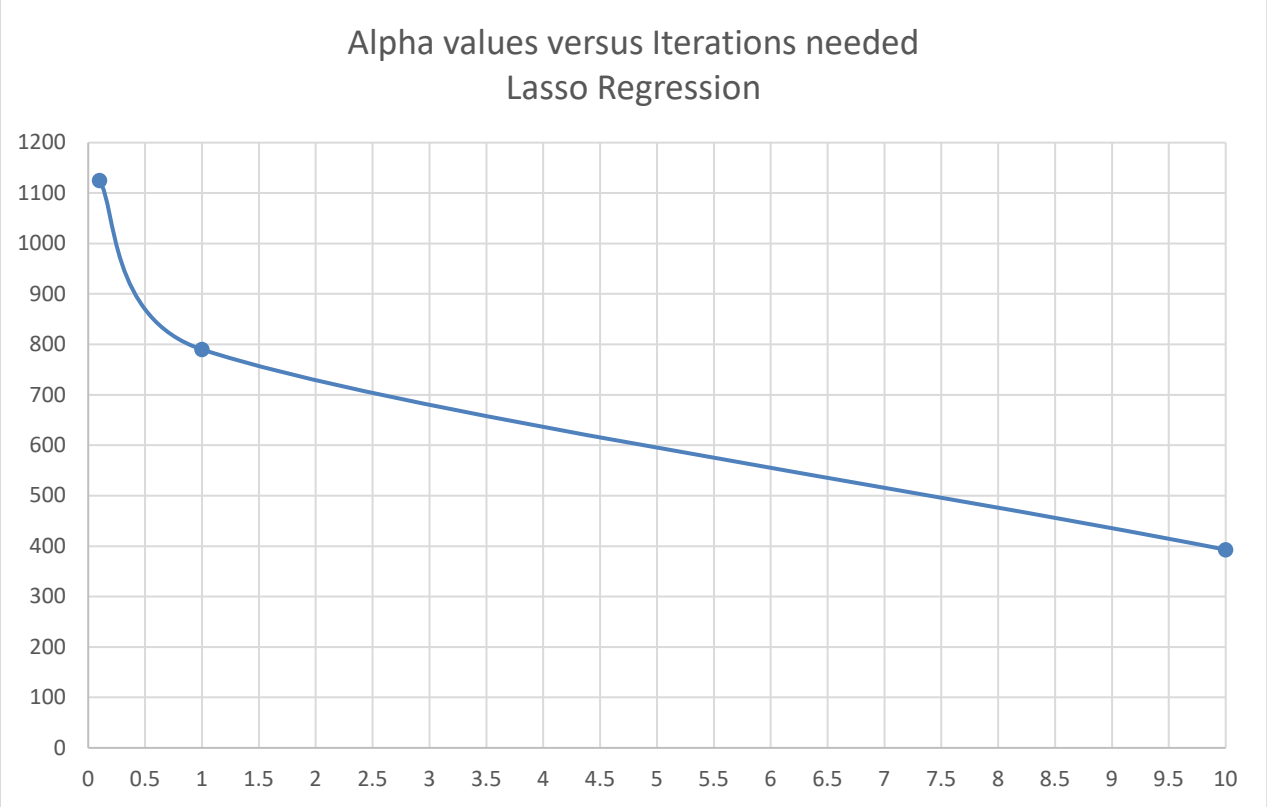


Figure 26 - Alpha values versus Iterations needed.

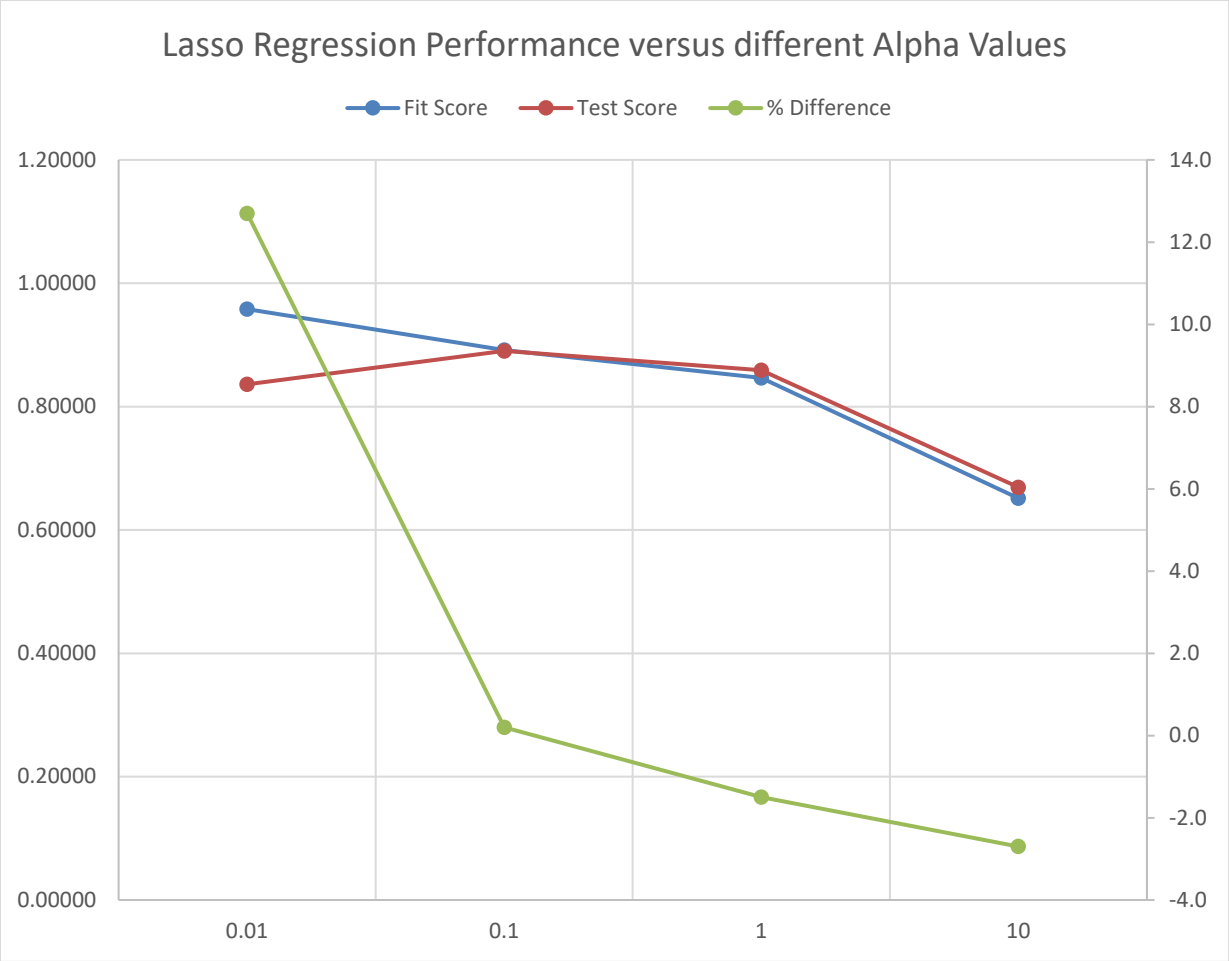


Figure 27 - Lasso Regression Performance versus different Alpha Values

Selection of most optimal Alpha Value and Corresponding Dataset Features

As it can be seen from the 2 above figures, increasing the alpha values allows for a way faster training of the Lasso Model for Feature selection with more **140-fold less iterations** going from **alpha = 0.01** to **alpha = 0.1**. However, increasing the alpha value more does begin to separate the Test Score from the Training score, indicating a value of 0.1 is close to the optimal value for this dataset.

Therefore, going forward with model evaluation, the feature selection was based on alpha = 0.1

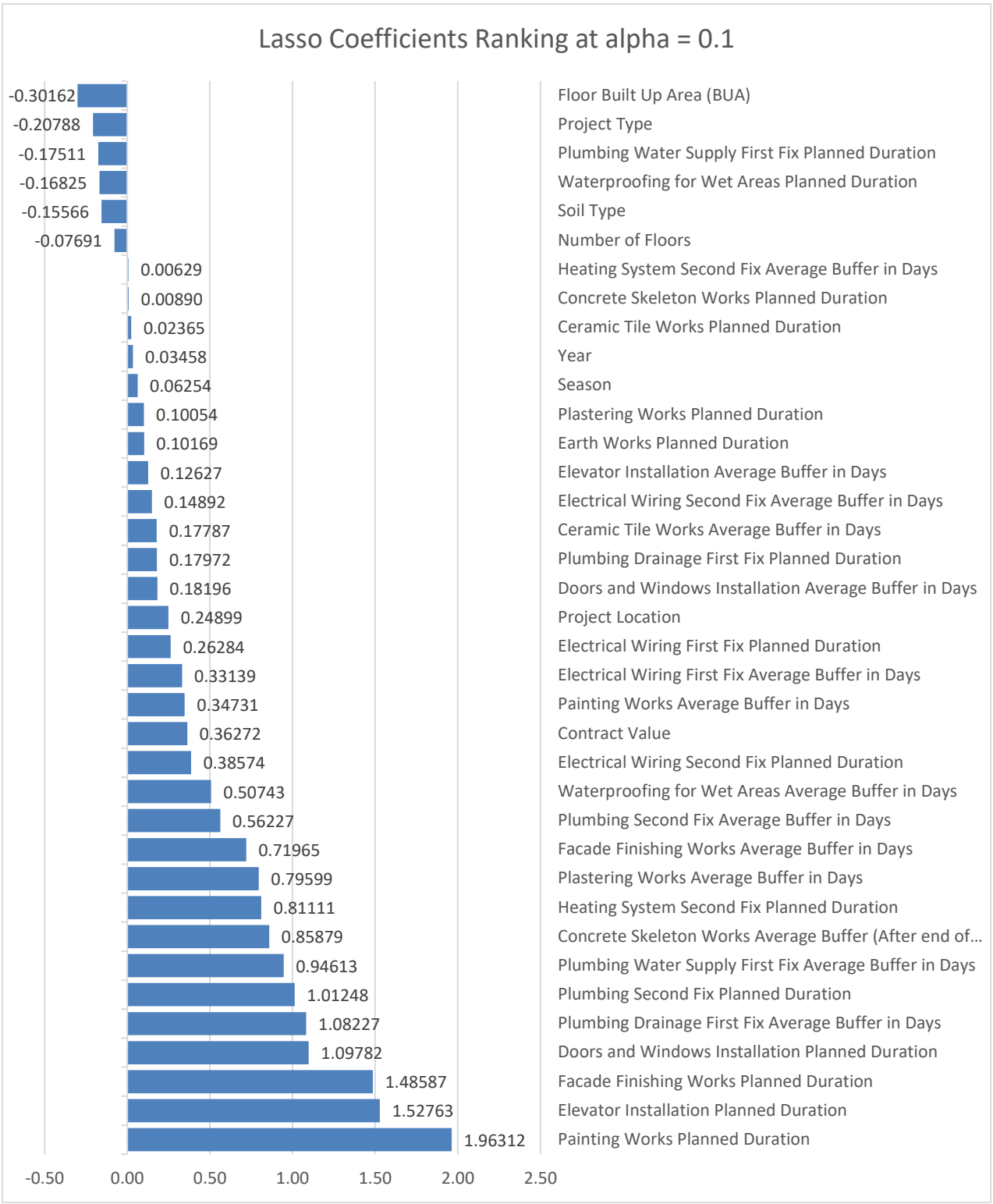


Figure 28 – Lasso Coefficients Ranking at alpha = 0.1

According to the above chart the following top 3 positive and negative features were selected.

Feature Name	Coefficient
Painting Works Planned Duration	1.96
Elevator Installation Planned Duration	1.53
Facade Finishing Works Planned Duration	1.49
Plumbing Water Supply First Fix Planned Duration	-0.18
Project Type	-0.21
Floor Built Up Area (BUA)	-0.30

Table 2 - Top 6 Lasso Features and Corresponding Coefficients for alpha = 0.1

Preliminary Analysis of Top Features

Preliminary analysis of the top selected features seems to be logical and realistic at first glance. The Project Type and per floor BUA negatively affect Activity Realistic Optimal durations meaning they would need more time. While the planned durations of the Activities at the tail end of the schedule seem to be a positive correlation which is logical in the sense that they are most likely Critical Path activities that would require the most effort to ensure their actual durations match or precede their planned counterparts

Durations Model Performance

Model Selection Justification

Now that the top 6 features have been identified it is time to move further to actual model performance. It is important to address some of the decisions made in the second development file since it contains the main ML Algorithm. First, justification for the models chosen from Sci-Kit Learn Library were based on the below figure from their website.

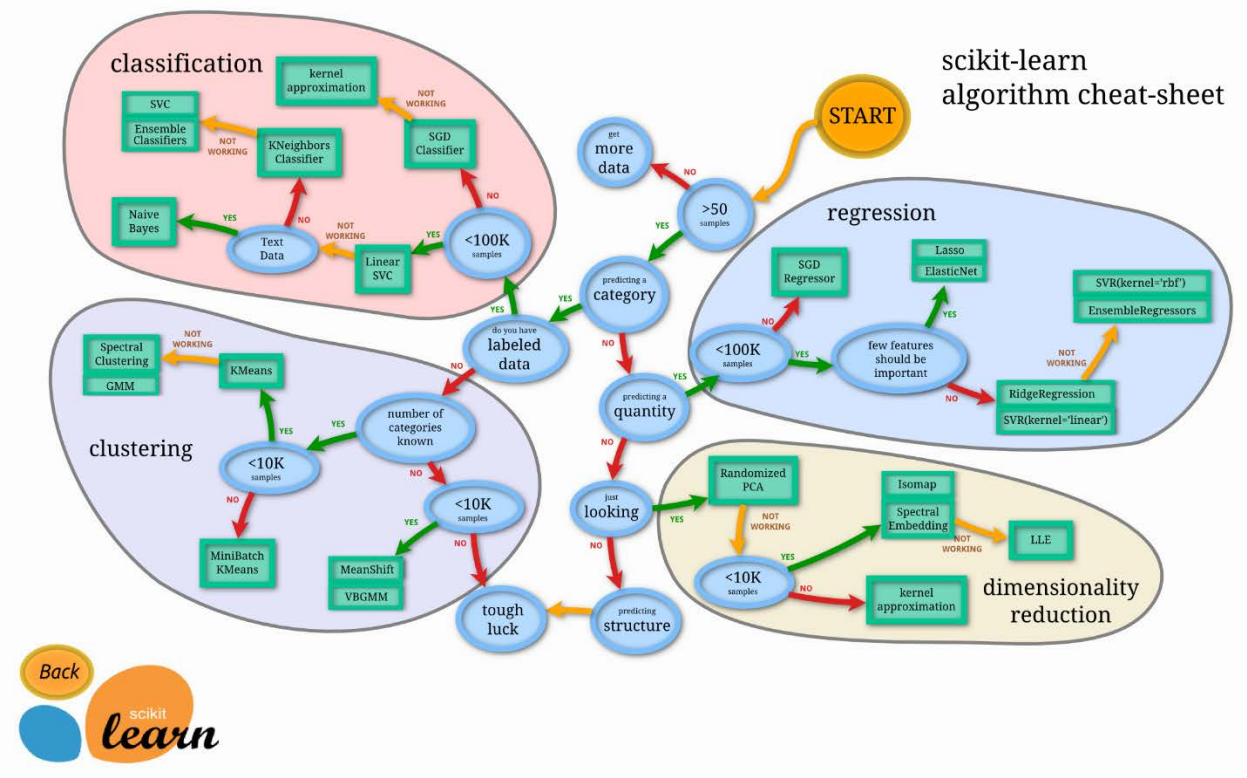


Figure 29 - Sci-Kit Learn Algorithm Cheat Sheet (scikit-learn, n.d.)

Following the above figure, the following route is taken:

Furthermore, other models, like a Dummy Regressor Model was added for a simplicity check of the model performance, as suggested by Sci-Kit Learn. And traditional Linear Regression, was added as in for example, Dimensionality simplicity check of the collected data if the Linear Regression Base Learner is always the best performing.

Application of Boosting Ensemble Methods

In addition to the selected Models (referred to herein after as Base Learners), Sci-Kit learn offers a selection of more robust Models called Ensemble Techniques. One of the most famous examples of those techniques is Random Forests Classifiers. Whereby many

decision trees are deployed on the model, each using different parameters and portions of the dataset. Afterwards, the outcome of the Forest is determined using the most used classification for example (Khushaktov, 2023). Which allows for a more robust algorithm resistant to overfitting (understanding a singular dataset too well to be used for general purpose). A simple diagram is shown below for better illustration.

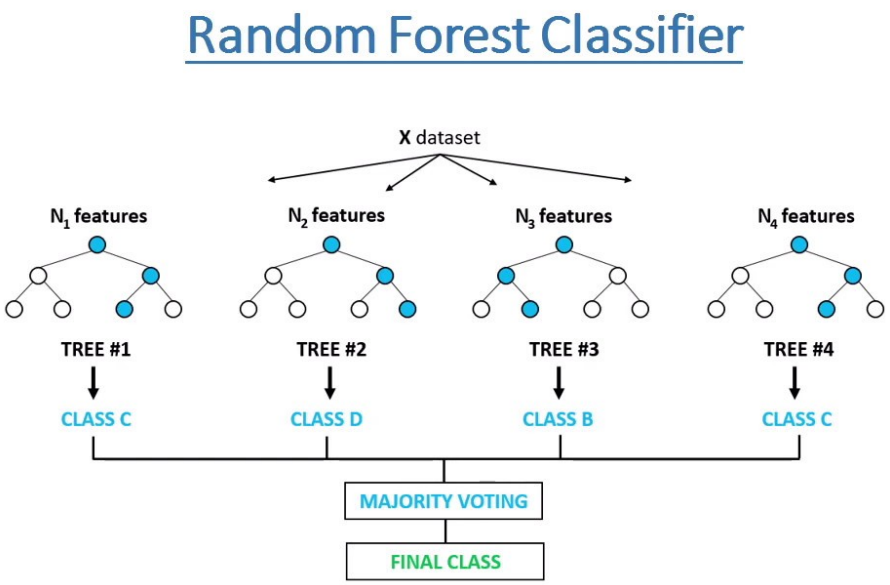


Figure 30 - Random Forest Classifier Diagram (Khushaktov, 2023)

Some of the provided techniques by Scikit Learn were utilized as follows:

1. **Stacked Generalization:** Combining Estimators through a Base and a Final Estimator to avoid biases of singular estimators. According to Scikit Learn the advantages of this approach is to "…combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability / robustness over a single estimator" (scikit-learn, n.d.). A Stacking Matrix is provided further down to exhibit how it was used.
2. **Bagging Meta-Estimator:** Similar in effect to the Random Forests approach as it creates multiple estimators on random subsets of the training dataset; then

combine their outcomes to generate a final prediction. It is a relatively straightforward way to enhance a single model without changing the underlying base algorithm (scikit-learn, n.d.). **It is very important to note that the Default Base Estimator in this Ensemble Library, a Decision Tree Regressor, was left as is. Since this Estimator is used as the final step of the Stacked Generalization Algorithm from the weak Base Learner**.

3. **Ada Boost:** In essence, to train a series of base learners on frequently updated versions of the data. Predictions are then pooled using a vote by weighted majority (or sum) to get the final projection (scikit-learn, n.d.). Each iteration, data is modified by assigning weights to each of the training samples. Initially, all those weights are set to $w = \frac{1}{N}$ where N is the number of samples. The initial process trains the base learner on raw data; then sample weights are individually adjusted for each iteration, and training is repeated on reweighted data. Until at a certain step, the weights of training examples that were mistakenly predicted by boosted model produced in the previous step are increased, while the weights of those that were correctly predicted are dropped. With more iterations, examples that are harder to forecast gain more weight. Therefore, the result is each base learner is forced to focus on the examples ignored by the prior ones in the series (scikit-learn, n.d.). **It is very important to note that the Default Base Estimator in this Ensemble Library, a Decision Tree Regressor, was left as is like the above Bagging Meta Estimator. Since this Boosting Estimator is used as the final step of the Stacked Generalization Algorithm from the weak Base Learner**

4. **Extreme Gradient Boosting Regression (XGBoost):** Optimized and highly efficient Open-source Library (XGBoost documentation, n.d.). It uses the **Gradient Boosting framework** (**Gradient Descent:** Typical optimisation procedure for minimising a cost function. Goal is to discover the optimal combination of parameters that minimises the difference between prediction & actuals. By randomly initialising the model's weights or coefficients. Then, calculating the gradient of the cost function with respect to each parameter, it continuously adjusts the values of the weights in the path of the steepest descent of the cost function.

Gradient Boosting: The ensemble technique combines several base learner's models to build a better one. By fitting the new model iteratively to residual errors of prior model. Final prediction is total of all models' predictions. The focus of gradient boosting is on the errors caused by prior models (Shi, 2023)).

Almost all the available Ensemble Methods applicable to this Regression Problem were imported into the model to ensure expandability of the Algorithm beyond a singular dataset scenario. The table below showcases the Regressor Matrix used for a total of **16 models** fitted to the dataset and compared.

Final Matrix of Chosen Regressors

Base Regressor Boost Regressor	Linear	Lasso	ElasticNet	DummyRegressor
AdaBoostRegressor	LinearRegression with AdaBoostRegressor	Lasso with AdaBoostRegressor	ElasticNet with AdaBoostRegressor	DummyRegressor with AdaBoostRegressor
BaggingRegressor	LinearRegression with BaggingRegressor	Lasso with BaggingRegressor	ElasticNet with BaggingRegressor	DummyRegressor with BaggingRegressor
GradientBoosting Regressor	LinearRegression with GradientBoosting Regressor	Lasso with GradientBoosting Regressor	ElasticNet with GradientBoosting Regressor	DummyRegressor with GradientBoosting Regressor
XGBRegressor	LinearRegression with XGBRegressor	Lasso with XGBRegressor	ElasticNet with XGBRegressor	DummyRegressor with XGBRegressor

Table 3 - Combination Matrix of chosen Regressors.

Finally, the **Stacked Generalization Regressor** was wrapped in a **Multioutput Regressor** Class to enable the prediction of multiple variables at the same time. For this purpose, a Class called **Regressor Chain** was used as it predicts considering the entirety of the available features as well as the predictions of earlier models in the chain (scikit-learn, n.d.). However, one of the difficulties of such complications is that Hyperparameter Tuning of base learners is not a possibility, although the stacking approach is originally

meant as an offset to this step. The final best model is selected according to the following Weighted average criteria Highest Score.

$$\mathbf{0.5*Test\ Score + 0.2*r^2 - 0.3*Median\ Abs.\ Error}$$

Selection of this weighted average formula was based on the Author's choice of trying to give the most weighted average to the Model Test Score on unseen data for generalization and the smallest Median Absolute Error in days. The model was tested with 3 datasets.

1. Full features dataset
2. Top 6 features only
3. Full dataset without the top 6 features

Dataset	Top Regressor	Test Score	Score	r^2 Score	Med Abs Error in Days	Run Time in minutes
Full Feats	ElasticNet Stacked with AdaBoostRegressor	0.90	-0.74	0.90	4.5	10
Top 6 Feats	LinearRegressor Stacked with AdaBoostRegressor	0.90	-0.725	0.90	4.52	6.25
Full – Top 6 Feats	ElasticNet Stacked with AdaBoostRegressor	0.89	-0.74	0.89	4.56	11

Table 4 - Duration Model Performance with Top 6 features versus full 25 Features

Analysis of Model Performance

When looking at the data from the above table isolating the top 6 features only to train the model does result in a better value in terms of run-times and model performance. However, that value is not statistically significant in comparison with either the full dataset, or even removing the top 6 features entirely, indicating that the underlying patterns can be learnt from the other interconnected features at the expense of more computational power to converge.

Nevertheless, the importance of feature selectin is clearer in the below figures comparing the Learning Curves and Prediction Errors for the 3 Tested datasets.

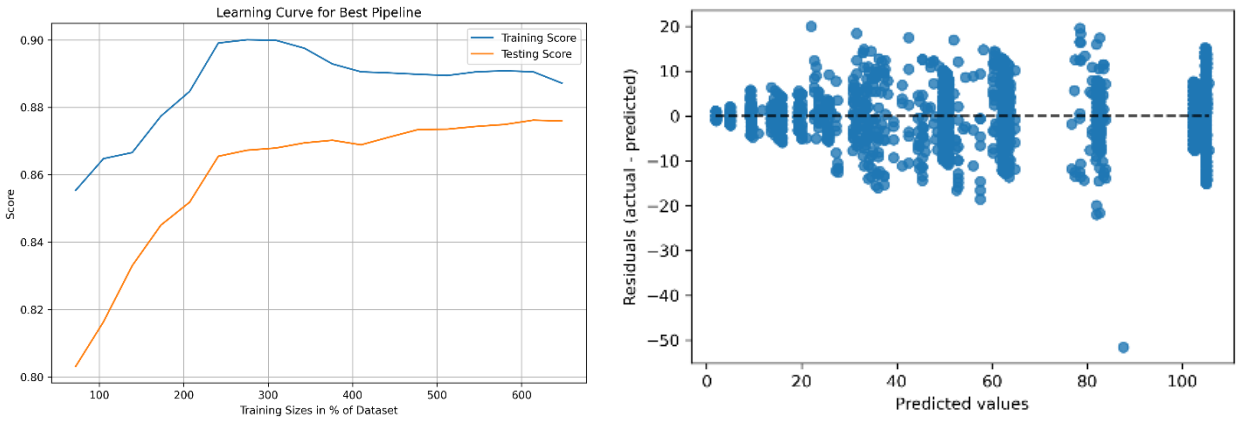


Figure 31 - Learning Curve and Prediction Error for Full Dataset

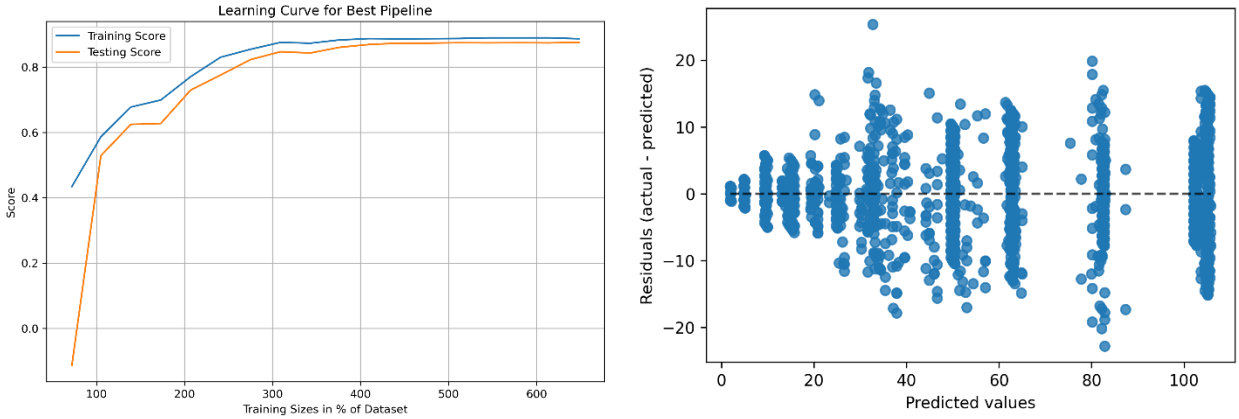


Figure 32 - Learning Curve and Prediction Error for Top 6 Features Dataset

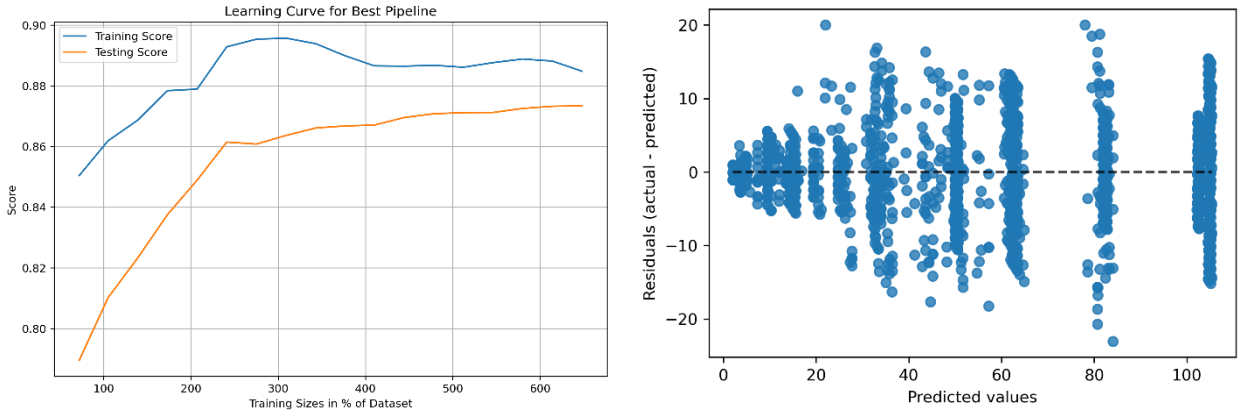


Figure 33 - Learning Curve and Prediction Error without Top 6 Features

The above figures represent the model's ability to learn and generalize to unseen data given multiple dataset sizes. The figure on the left represents the models' score and the figure on the right is the model predictions against residuals from true labels.

While the model's metrics are almost identical given any type of dataset. It can clearly be observed that isolating the Top 6 features alone allow the model to learn and generalize way mor effectively as shown by the middle figure Learning Curve of the Training and Testing Scores being very closely related for given dataset sizes. Additionally, the Prediction Residuals error plot is much smaller in the middle figure shown by Residuals being closer to 0 (ideal prediction).

The Learning Curve and Prediction Error Plots are a stronger indication that isolating the top 6 features was a successful extraction of the most relevant features in the Durations Model

Comparison of Different Train-Test-Splits using 6 Top Features Dataset

The next Comparison is aimed at seeing the difference between different Training and Testing Splits for the Duration Model. For this comparison the Top 6 features dataset with number of samples 1000 Projects and different Training portions are tested according to the below table

Train %	Top Regressor	Test Score	Score	r^2 Score	Med Abs Error in Days	Run Time in minutes
40%	Lasso Stacked with AdaBoostRegressor	0.888	-0.771	0.888	4.64	3.75
60%	Lasso Stacked with AdaBoostRegressor	0.889	-0.764	0.889	4.62	5.25
80%	LinearRegressor Stacked with AdaBoostRegressor	0.90	-0.725	0.90	4.52	6.25

Table 5 - Comparison of Different Training % on Duration Model Performance

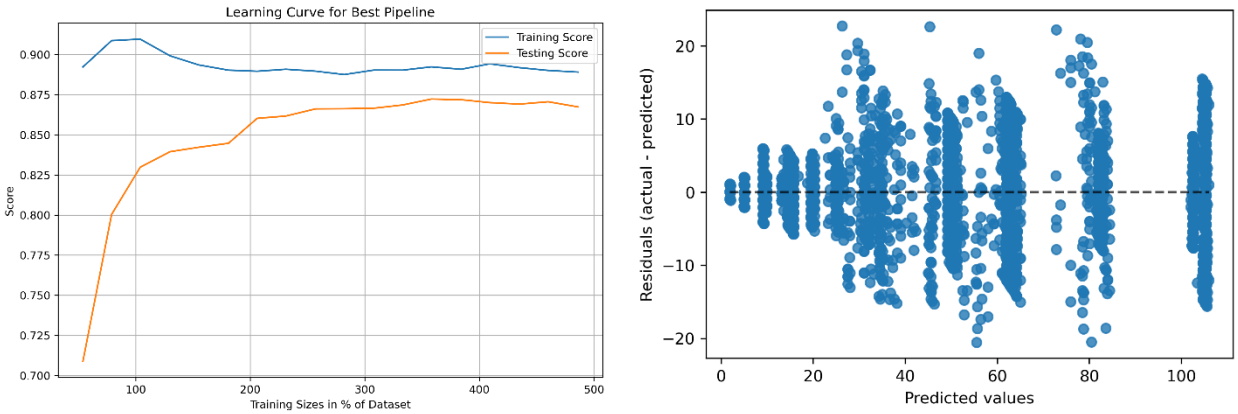


Figure 34 - Learning Curve and Prediction Error Plot of 60% Training Split

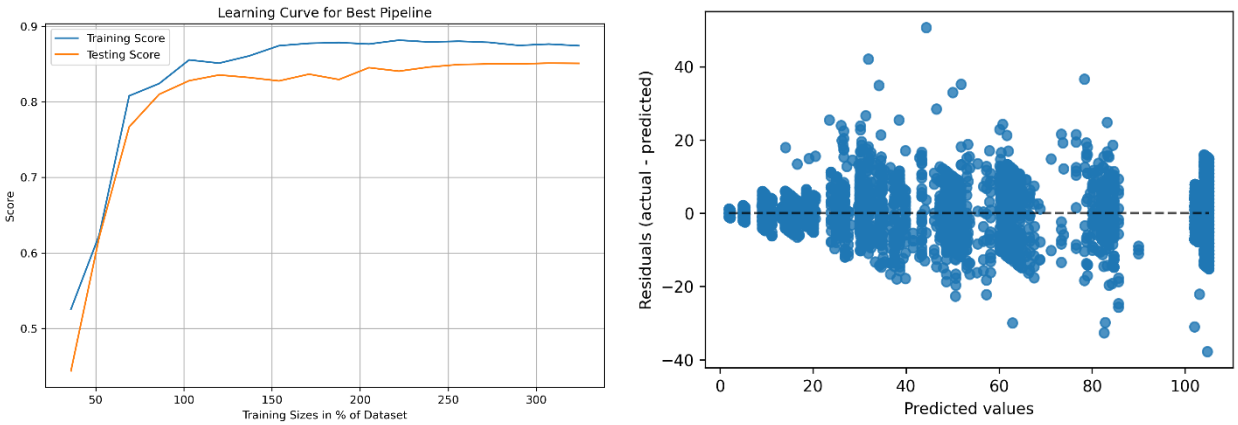


Figure 35 - Learning Curve and Prediction Error Plot of 60% Training Split

Once again just looking at model metrics alone does not paint the whole picture when it comes to the best training split; it is apparent that a 40% Training split is too low as evident by the scale of the Prediction Error Curves. Additionally, in this split, the r^2 Score of both Training and Testing Datasets improves dramatically until both datasets approaches 100 samples or more. However, the difference between the **60% and 80% Training Split** is a lot more subtle and requires more analysis through Prediction Error Curves Plots. The 80% Prediction Error Plot is a lot more concentric around the 0 residual point. Showcasing better overall accuracy shown with the below comparison. Moreover, the Residual values (-10, 10) is much denser than in the 80% Training Split

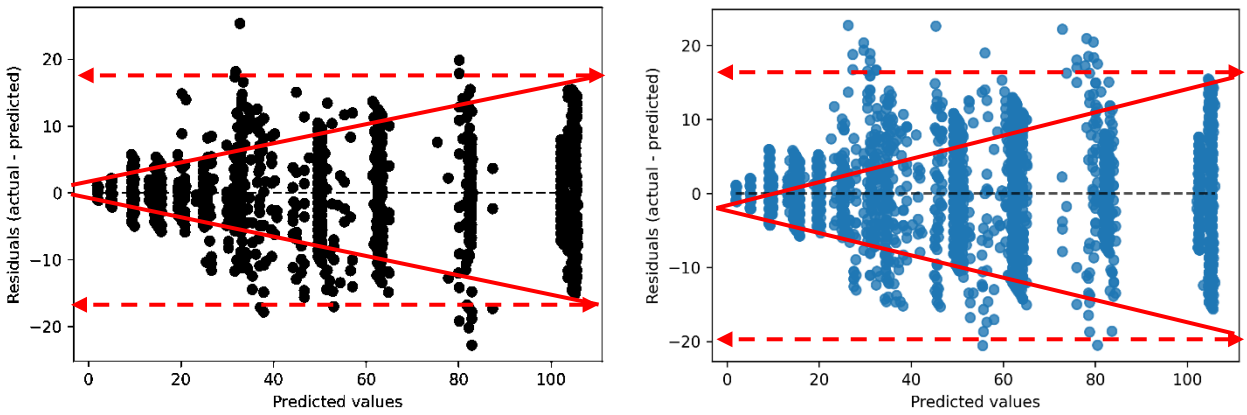


Figure 36 - Comparison of Residual Spread between 60% (Right) and 80% (Left) Split

Effect of different Missing Data Percentages

The final comparison is aimed at exploring the effect of missing data in the Dataset on the Model Performance. To handle missing data in the model there are 2 methods. The first is to drop any missing data rows from the dataset using the Pandas function "**my_dataset**.dropna()". This will parse through the entire dataset rows and drop any row that has an empty column. This approach is very hard to recommend since real world data will often be very messy and dropping any empty row will adversely affect model performance due to the massive shrinking of the dataset.

Therefore, the second approach is utilized which is dataset Imputation or prediction of missing variables done through the preprocessing library of Scikit Learn. For this approach a Pre-Processing Pipeline was made that uses the "**KNN Imputer**" Module to Predict the missing value according to a set amount **K = 2** of the Nearest Neighbours (NN). The setting of **K = 2** was a heuristic choice aimed at filling in the missing value with the closest related project as they will most likely have similar planned variables. The results of the Comparison are shown in 2 tables below. The default Value of the Imputer is **K = 5.**

Missing Data %	Top Regressor	Test Score	Score	r^2 Score	Med Abs Error in Days	Run Time in minutes
20%	ElasticNet Stacked with AdaBoostRegressor	0.87	-0.57	0.87	3.97	7.5
40%	Lasso Stacked with AdaBoostRegressor	0.81	-0.62	0.81	3.97	7.25
60%	ElasticNet Stacked with AdaBoostRegressor	0.61	-0.89	0.61	4.40	7.5

Table 6 - Comparison of Missing Data on Duration Model Performance

The approach of filling in missing data is not without its flaws as well, when the percentage of missing data is below 50% the regression efficiency is still acceptable because the imputer's effect on guessing the missing data is limited (i.e. more data is present than not). However, when the dataset has a very large number of missing data the imputer's effect is shown greatly in the regression results as it simplistically replaces the missing data therefore making it more difficult for the regression fitting process. This is shown in the graph below comparing the Testing Score to the amount of missing data.

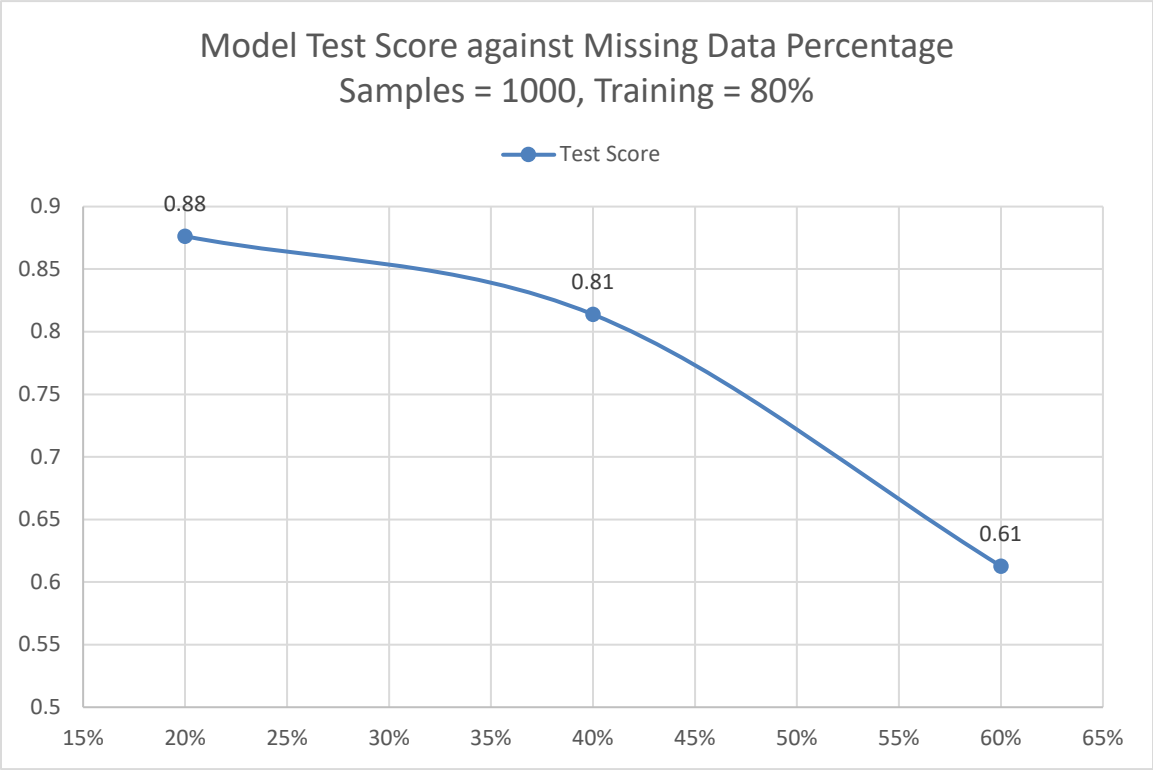


Figure 37 - Graphing Different Amounts of Missing Data vs Model Performance

Comparison with similar Models from Literature

Now that all different aspects of the Duration Model have been tested, it is good to do a comparison between similar efforts from the literature. In the research by (Sanni-Anibire et al., 2021) their Construction Duration Estimation model used for risk delay mitigation yielded an 0.69 R^2 as shown in the below figure summarizing their model results score.

TABLE 8: Performance of developed machine learning models

Model	Regression performance			Classification performance	
	R^2	RMSE	MAPE	Classification Accuracy	Misclassification Error
Duration model	0.69	301.76	0.18	-	-
Cost model	0.81	6.09	80.95	-	-
Delay risk model	-	-	-	93.75	6.25

Figure 38 - Duration Model accuracy (Sanni-Anibire et al., 2021)

In Comparison, the developed duration model utilizing a Generalized Stacking approach to increase regression performance ended up with a much higher r^2 score of around 0.8 to 0.9 depending on the dataset structure and a minimal amount of missing data.

Model 2: Activity Relationships Prediction

This model is aimed at understanding the relationships between activities in a Schedule. This is explained by a model being able to learn and predict accurately the successor activity in a schedule, or the necessary predecessor given the current activity. Additionally, the type of dependency should be included as well like a Finish-to-Start Relationship, Start-to-Start or Finish-to-Finish.

Synthetic Database Breakdown

For this model the same Synthetic Dataset approach was used with the necessary modifications to allow Classification of Activity Relationships. The Synthetic Database consists of 1000 schedules. There were two datasets used to test the models, one dataset only included **Activity Names/IDs, Predecessor Activity IDs, and Relationship Types**. The second one added the below features for each activity in the same project.

1. Buffer
2. Project Location
3. Project Type
4. Year
5. Season
6. Contract Value
7. Soil Type
8. Floor Built Up Area (BUA)
9. Number of Floors

Excel Formulas were used to randomize the Predecessor Activities and Relationship Types. For example, the **Plumbing Water Supply First Fix** predecessor activity was calculated using the following:

```
=CHOOSE (RANDBETWEEN (1, 6), B3, B4, B6, B3 & ", " & B4, B3 & ",
                    " & B6, B4 & ", " & B6)
```

Where B3 is **Earth Works**, B4 is **Concrete Skeleton Works**, and B6 is **Plumbing and Drainage First Fix**

After that, the Activity Relationships are determined based on the number of Predecessors using this formula.

```
=IF(ISBLANK(D5),"“, IF(COUNT(SUBSTITUTE(D5,",","")) =1, CHOOSE
(RANDBETWEEN (1, 3), "FS", "SS", "FF"), CHOOSE (RANDBETWEEN (1,
```

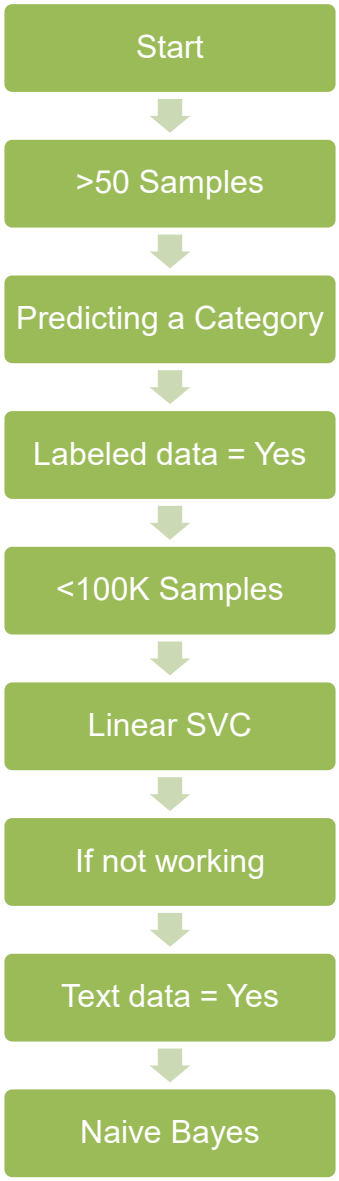
```
6), "FF, FS", "FF, SS", "FS, FF", "FS, SS", "SS, FF", "SS,
FS")))
```

The formula fundamentally counts how many Activity IDs are in the Predecessor ID Cell, and Assigns a random number of Relationship Types equivalent to it from all possible permutations of **FS, SS, and FF.**

The approach followed was trying to simply the problem as much as possible by focusing on each single activity. This yielded a simple classification procedure where each activity would have its own set of Features against a known Classification Target. The advantage of this approach versus the ones outlined in the Literature like using a LSTM NN (Amer and Golparvar-Fard, 2021) or a BD-LSTM (Amer and Golparvar-Fard, 2019) is that it should correspondingly allow for Dataset Expansion through adding multiple new Activities when needed. With the downside of training the model recurrently on each activity as a singular target

Approach 1: Utilizing Sci-Kit Learn Classification

The first and simplest approach was to utilize the similarities between the Durations Model Code to build another model that does Classification using Sci-Kit Libraries. For this approach the **Model Selection Justification** Subsection of the previous model was used leading to the following process



Linear SVC Model Performance

The Linear SVC Model was first tasked to predict only the Activity Predecessor ID Column of all activities, then the Dependency Type Column. Finally, the Linear SVC Model was wrapped inside a Classifier Chain Module to allow multiple predictions at the same time; meaning the model can now predict both the Predecessor ID and Dependency Type together.

Model	Classification Target	Number of Samples	Features	Train %	Accuracy	Precision
Linear SVC	Predecessor ID	1081	11	80	0.440	0.440
Linear SVC	Dependency Type	1081	11	80	0.272	0.272
Linear SVC Classifier Chain	Both Targets	1081	11	80	0.118	0.118

Table 7 - Linear SVC Model Metrics in Classification of Activity Relations

As it can be seen from the Accuracy and Precision Results above, the Linear SVC model has middling performance at best. Prediction of Predecessor Activities seems promising but Dependency Types and Multi-Label Classification of both categories at the same time falls flat. Therefore, Linear SVC Model was designated as unsuccessful for Activity Relationships Classification. And the author decided to move on to the next recommendation which is Naïve Bayes

Naïve Bayes Classifier Model Performance

To prevent going into unnecessary further details it is sufficed to say that unfortunately, this approach also suffered as the Naïve Bayes Model ended up performing almost identically to the SVC Model as seen from the below table.

Model	Classification Target	Number of Samples	Features	Train %	Accuracy	Precision
Linear SVC	Predecessor ID	1081	11	80	0.440	0.440
Linear SVC	Dependency Type	1081	11	80	0.279	0.279
Linear SVC Classifier Chain	Both Targets	1081	11	80	0.091	0.091

Table 8 - Naive Bayes Model Metrics in Classification of Activity Relations

Alternate approach: Utilizing a different Dataset Format

Since all the traditional classification approaches were unfortunately not suitable for utilization to effectively predict the relations between activities; before delving into different approaches shown below. It was thought by the Author that modification of the dataset format to a more suitable classification format can lead to improved results. For this method. The dataset format was changed into a Matrix Shape with the following characteristics:

Activity Name	Activity 1	Activity 2	Activity 3	Activity 1 Link Type	Activity 2 Link Type	Activity 3 Link Type
Activity 1						
Activity 2	1			FS		
Activity 3		1			SS	

Table 9 - Activity Relations Matrix Dataset Format

Where in the above table, all unique activities are scattered horizontally, with all activities in existing samples scattered vertically. And the number “1” implies that the horizontal Activity is a Predecessor to the vertical Activity. While the second half of the matrix implies the relationship type between them. This approach was imagined playing on the strengths of Binary Classification abilities of Machine Learning Models aiming to find the most suitable boundary line given different features affecting the relationships between activities.

Afterwards, the Sci-Kit Learn **Random Forest Classifier** Model was selected to train on the dataset since it is the only model that can Multi-Label (Existence or none for *Relation and Link Type*), and Multi-Class (*a certain Activity can relate to different activities in different samples*) Classification. This resulted in very favourable performance to the model. A summary of the results is shown in the below table.

	Accuracy	Corresponding Activity
Maximum	1.00	Project Start: Mobilization
Minimum	0.594	Doors and Windows Installation
Median	0.749	-
Mean	0.7496	-

Table 10 - Performance Summary of Random Forest Classifier Mode

A more detailed per activity result, and training time needed is included in the Appendices. In the meantime, it is sufficient to say that this approach is more favourable, and relatively easy to implement for Activity Relations Prediction. However, the major downside is creation of the dataset which involves a lot of manual labour for annotation and transformation into the desired format. It is to be noted that more automated techniques can be used to transform the extracted schedules into the desired format.

Two more valid approaches were attempted for this Thesis with more great success. Building a Graphical Neural Network (GNN), and Fine-Tuning of a Large Language Model (LLM) for Classification Tasks

First Alternate Approach: Using GNNs

To preamble the reasoning behind this approach it is first important to explain the concept of Activity-On-Node (AON) Graphs for Scheduling in general, and AEC Scheduling in particular. This concept is taught in scheduling classes for easier understanding. In AON, each Activity is represented through a Node (usually Square, Circle, or Rectangle) and an outwards line is drawn from that activity to the successor one. This process is repeated until the final activity is reached for each path. Usually, additional nodes for Project Start and Project End are also created to combine all nodes. The figure below is an easy representation of such a concept in action.

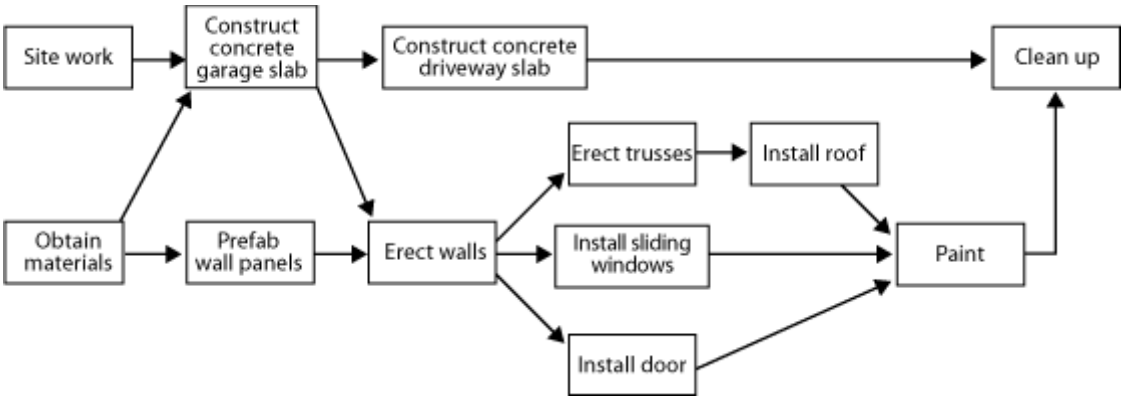


Figure 39 - Example of Construction AON Graph (RMIT international university, n.d.)

Utilizing the above information to move forward into GNNs. These Neural Networks work specifically on Input data easiest represented in a Graphical Shape. For example, Chemical Bonds between Molecules in pharmaceutical research; or connections between accounts in social media (Sánchez-Lengeling et al., 2021).

A graph can be represented by its parts through representing the Vertices, or Nodes, and Links between those vertices as Edges either with or without directions. Finally, the overall graph can be represented as a single point (for example a water molecule) in what is called a Global or Master Node for the whole graph (Sánchez-Lengeling et al., 2021).

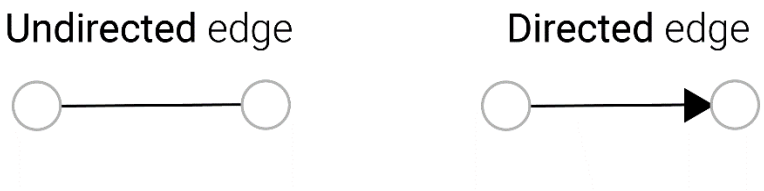


Figure 40 - Directed versus Undirected Graph Edge (Sánchez-Lengeling et al., 2021)

This allows the Graph Data to be represented as two matrices. The first Matrix is for the Number of Nodes, and Features for each node (For example Activity Durations, Resources…etc.). And the second is Adjacency Matrix; which represents all combinations

of Nodes that share an Edge or a Link together; noting that if the Link is non-directional, it is represented as **2 links A>B, and B>A**.

The ML Terms for this Data format is henceforth known as Feature Matrix and Edge Index. Additionally, if the links themselves have specific Features, a third input Matrix called Edge Attributes is available.

An important matter to discuss is how does a GNN "Learn" from the graph data. This is done through a mechanism called "Message Passing" and it consists of choosing one of the three ways shown in the below figure.

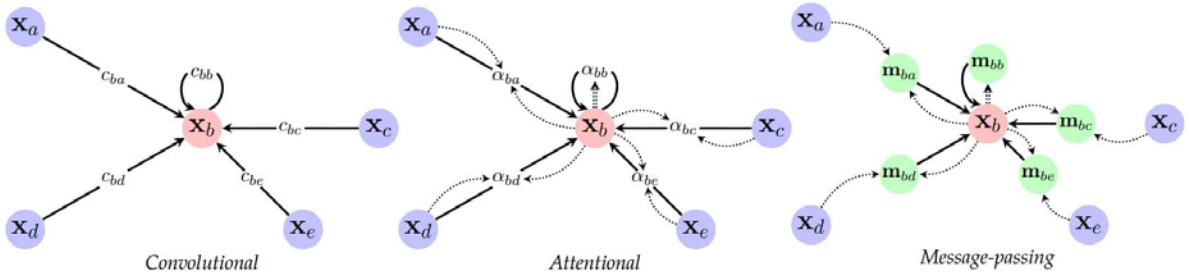


Figure 41 – How GNNs learn from Graph Data (Merritt, 2022)

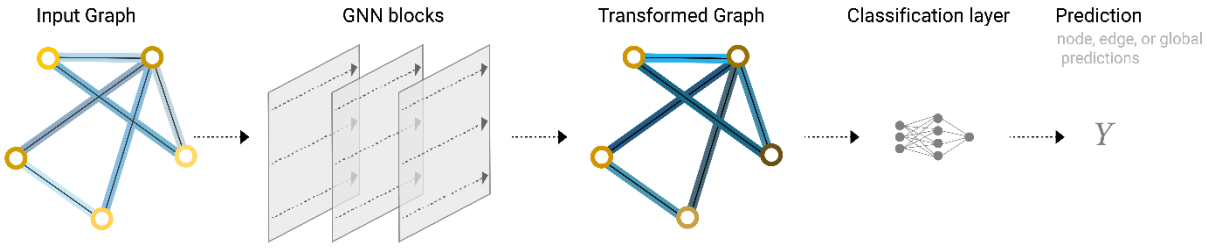


Figure 42 – Start to End overview of how GNN works (Sánchez-Lengeling et al., 2021).

Choosing the correct Learning Framework

The PyTorch Geometric library includes a variety of GNN architectures to choose from. Therefore, it is important to identify the use case of using a GNN, which will in turn feedback into the most suitable choice of framework. The purpose of this model is to classify the links between activities and the corresponding link type. This means that, given a list of Graph Nodes, or Activities. The model should be able to predict the Adjacency Matrix, in addition to the correct feature class for each edge.

The paper by (*Zhang et al., 2019*) providing a comprehensive overview of the types of GNNs and their current use cases provides the author with very good cases. For example, the researchers highlighted the use of GNNs in Chemistry and Biology whereby it seems like GNNs are used for overall Graphical Properties prediction like Drug side effects, and molecule properties. More information was also gathered from (*Karagiannakos, 2021)* which indicated that Graph Convolutional Networks have a problem of not supporting Edge Features like the Link type. Which removes them from the Learning Framework selection essentially.

A few selections from the Author's research stood out. First is Graph Attention Networks (GAT), This architecture attempts to compute weights for each pair of nodes embedded knowledge and organize them in the order of importance like how the current LLMs are trained (*Brody et al., 2021*). And GraphSAGE Networks, with a focus on something called "Node Neighbourhoods" whereby the model focuses on the neighbouring nodes for data In the Introduction paper of this Learning Framework by (*Hamilton et al., 2017*). It was specifically mentioned the main advantage of this method is generalizing to unseen Graphs/Nodes. Finally, an architecture called Knowledge Graph Embedding (KGE). A knowledge graph is a collection of real-world links between entities. It is made of 3 main components, the node or entity, the edge or connection, and the label or the connection type (*What Is a Knowledge Graph? | IBM, n.d.).* The easiest example is a social network knowledge graph whereby the nodes are the personal accounts, the edges are the links between accounts, and the label is something like "friends with", "married to", or "might know".

Model 2, Approach 1, Code Breakdown Structure

Phase 1: Data Preprocessing

To build this model it was first needed to figure out how the graph data is going to be pre-processed. For this goal, an open-source Python Library called Network-X was used. This library allows the construction of Graphs from tabular content given a Source and a Target (NetworkX — NetworkX documentation, n.d.). Additionally, it allows the visualization of the drawn graph; and extraction of the needed graph data for input to the GNN Model. An example is given below showcasing how the tabular data can be extracted from popular planning software like Microsoft Project and directly fed into Network-X.

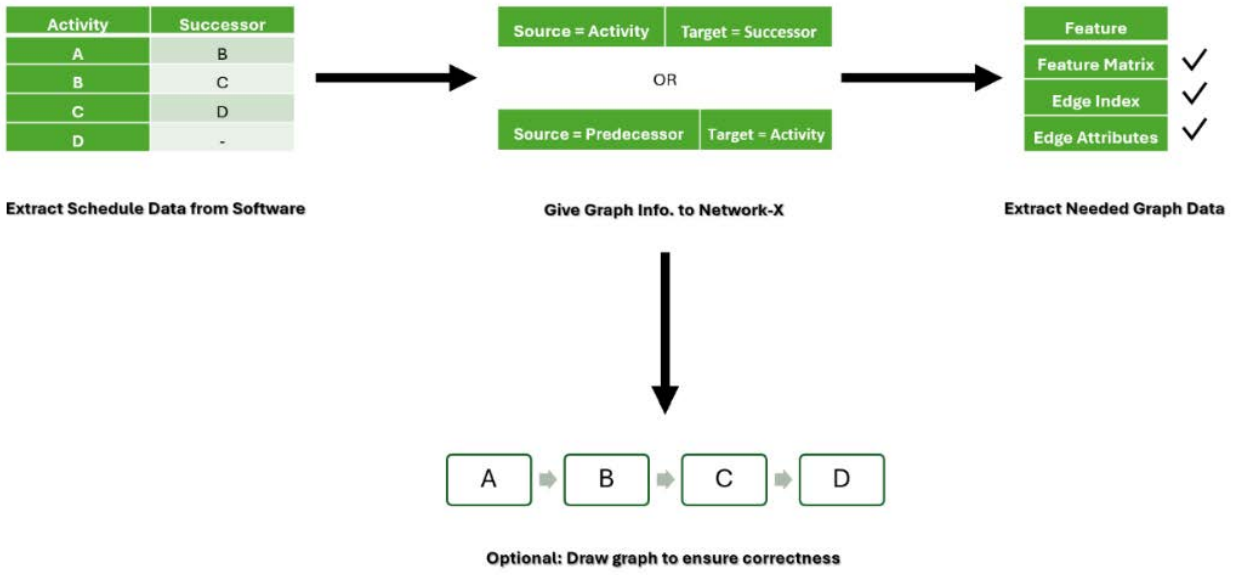


Figure 43 - Transforming Tabular Data into Graph Data with Network-X

After transforming the tabular data into Network-X Data. It is now possible to feed that input to the ML Framework used which is PyTorch; an open-source framework for building and deploying all types of models. A module specific to the GNN called PyTorch Geometric was used to store the graph information for each graph from Network-X

Phase 2: Model Building, Training Loop, and Validation Loop Code

The GNN is built around the approach of being given multiple graphs of varying lengths and complexities.

As an important note, due to the difficulty of assessing the correctness of graphs rendered by Network-X in the Synthetic Dataset created for previous models; the author resorted to a simpler type of dataset like the ones used in Scheduling Courses to teach students how to draw an AON.

This dataset consisted of **500 Graphs** describing projects up to **25 Activities** long. The dataset was built using a Python Script generated with the help of LLM Agents. The script is given some additional parameters to ensure that for example there are no isolated nodes, and that the number of Link Types match the number of Links

KGE versus GAT versus GraphSAGE

First model architectures tried were the GAT and GraphSAGE Models. In addition, two approaches were used to try and make the model understand the links between nodes. The first one is the standard Score Matrix in the shape of [*number of nodes x number of nodes*] wherein every entry in the matrix represents a probability of an Edge or Link existing between these 2 nodes, commonly referred to as a Node Pair Scoring. If the predicted probability is higher than the threshold (typically 0.5 or 50%). The link is predicted as existing. The end model output is compared to the True Adjacency Matrix which is in the same shape; and each link is shown in binaries 1 or 0 format. A small example is provided below

	Predicted Probability Matrix				**True Adjacency Matrix**		
	Node 1	*Node 2*	*Node 3*		*Node 1*	*Node 2*	*Node 3*
Node 1	0.757	0.047	0.120	Node 1	0	0	0
Node 2	0.331	0.600	0.765	Node 2	0	1	0
Node 3	0.142	0.872	0.831	Node 3	1	0	1

Table 11 - Example of Score Matrix Output from GNN Model

Loss Function Problems

All models were instructed to do 10,000 epochs with the dataset split being 90/10 for Training/Validation and a with a modest $\frac{1}{10000}$ or ***0.0001*** Learning Rate for the Model Weights Optimizer. It was quite unfortunate to see that the model loss over epochs plateaus around the 0.66-0.69 mark. This loss value translated to the model being stuck at calculating the probability distribution for the existing versus non existing edges. The author predicts the reason for this could be the due to the adjacency matrix shape being like ***the Alternate approach: Utilizing a different Dataset Format*** discussed above using Sci-Kit Learn which makes it difficult for the model to separate the connecting versus not edges. Accordingly, when calculating the Accuracy score as $\frac{Correct\ Predictions}{Total\ Predictions}$ it may have been considered high; but a correspondingly abysmal Precision Score for correctly identifying the Existent Edges calculated as $\frac{Correctly\ Predicted\ Positives}{Actual\ Positives}$ had a maximum of 35% for validation graphs. Overall, the final Validation Metrics seem to indicate that utilizing a GAT or GraphSAGE for Activity Link and Link type prediction was unfortunately not a successful endeavour. There are of course improvements that can be made however the author was not able to find a working combination.

KGE Implementation

A very pleasant surprise came when attempting to utilize KGE for the same task as the model crushed the training graph with an almost 0 loss value over the epochs. It is important to note that Pytorch Geometric provides 4 types of KGE Model implementations, TransE, RotatE, ComplEx, and DistMult. All of them were experimented on a single trial graph picked at random from the dataset and saved for use afterwards. The result of each model is shown in the comparison graph and corresponding table below

Epoch	TransE	RotatE	DistMult	ComplEx
1	0.3586	0.4881	0.6884	0.6829
1000	0.2711	0.3541	0.6315	0.5628
5000	0.2265	0.1753	0.1872	0.039
10000	0.1991	0.1124	0.0753	0.0012

Table 12 - KGE Types Loss over Epochs for Link Prediction

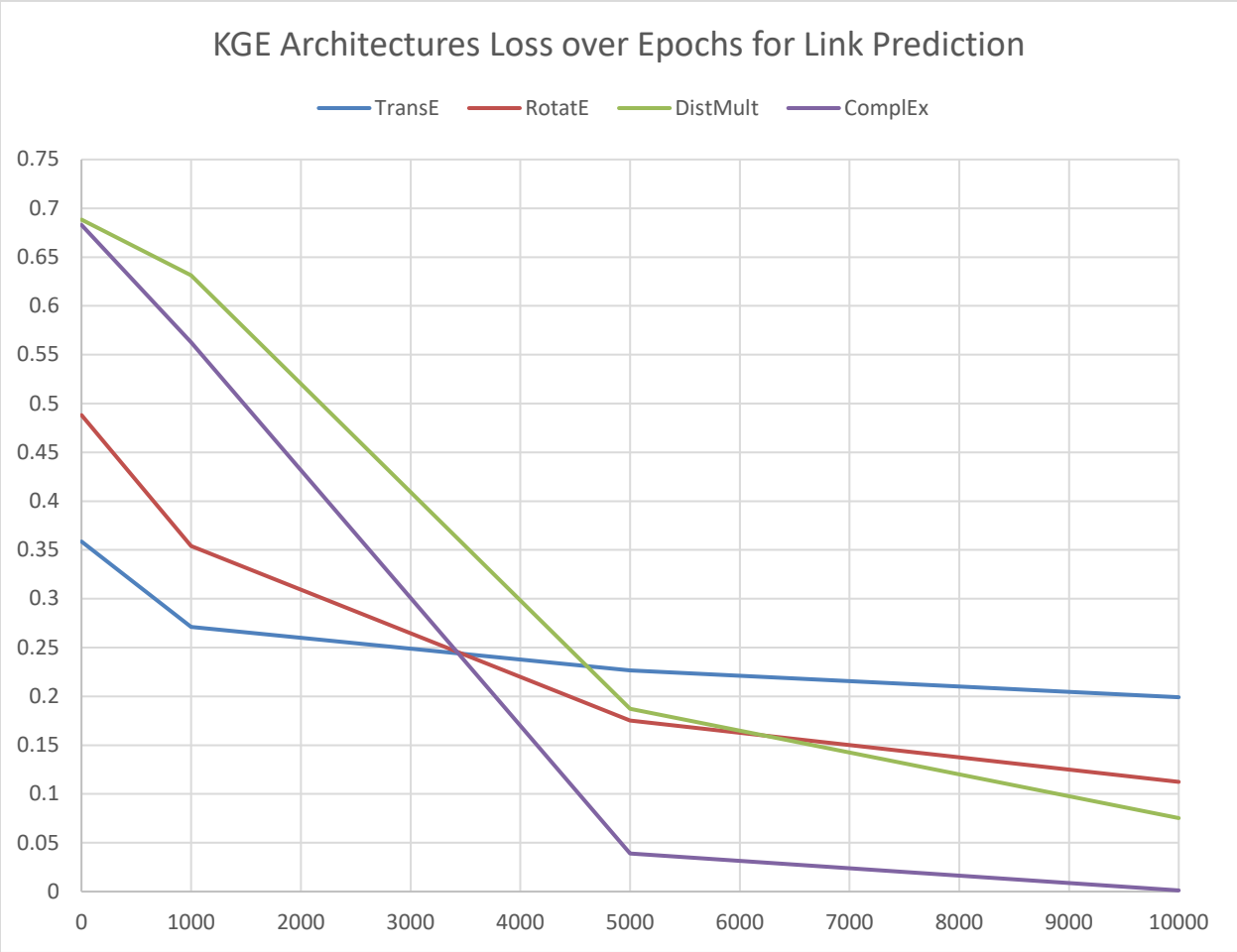


Figure 44 - KGE Types Loss over Epochs for Link Prediction

It can be seen from the above graph that even though TransE and RotatE types started out with the least amount of loss during initialization, they were very quickly overtook by ComplEx and DistMult types. With the overall seemingly best type being the ComplEx KGE. Finally, this model type was used on the full dataset. Comprehensive training and validation results are shown below

Epoch	Total Mean Loss	Link Prediction Loss	Link Type Loss
1	2.1682	0.702	1.4663
100	2.0655	0.699	1.3663
500	1.7405	0.689	1.0519
1000	1.4549	0.660	0.7953
2000	1.1067	0.509	0.5978
5000	0.8134	0.278	0.5359
7000	0.7997	0.2697	0.53
10000	0.793	0.2649	0.5281

Table 13 - Total Mean Loss over Epochs for Link Prediction using ComplEx KGE

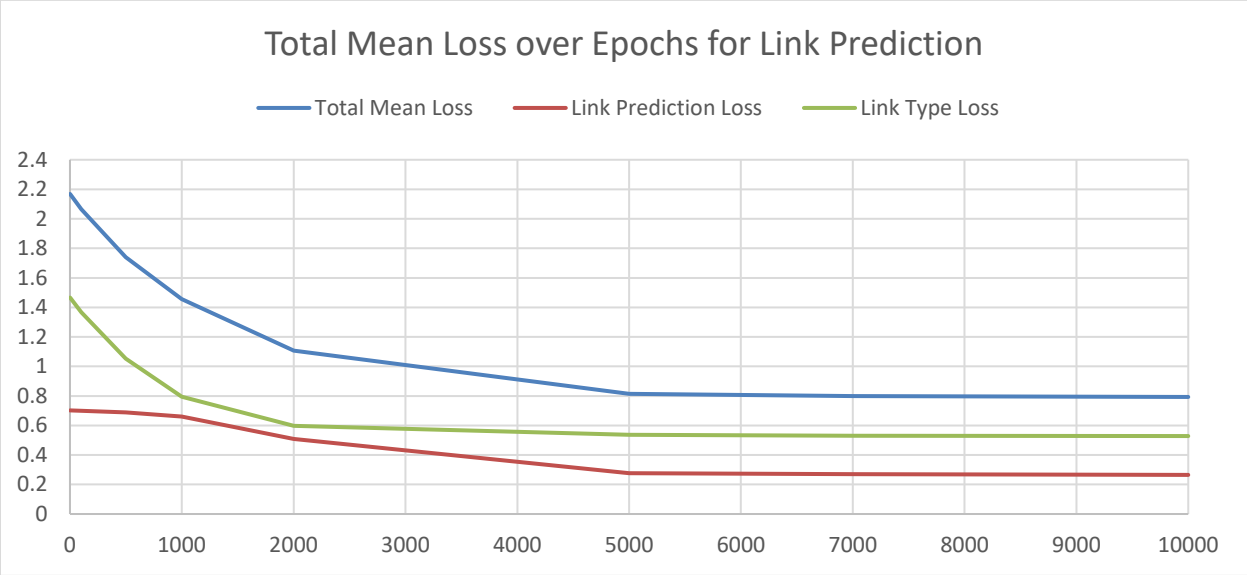


Figure 45 - Total Mean Loss over Epochs for Link Prediction using ComplEx KGE

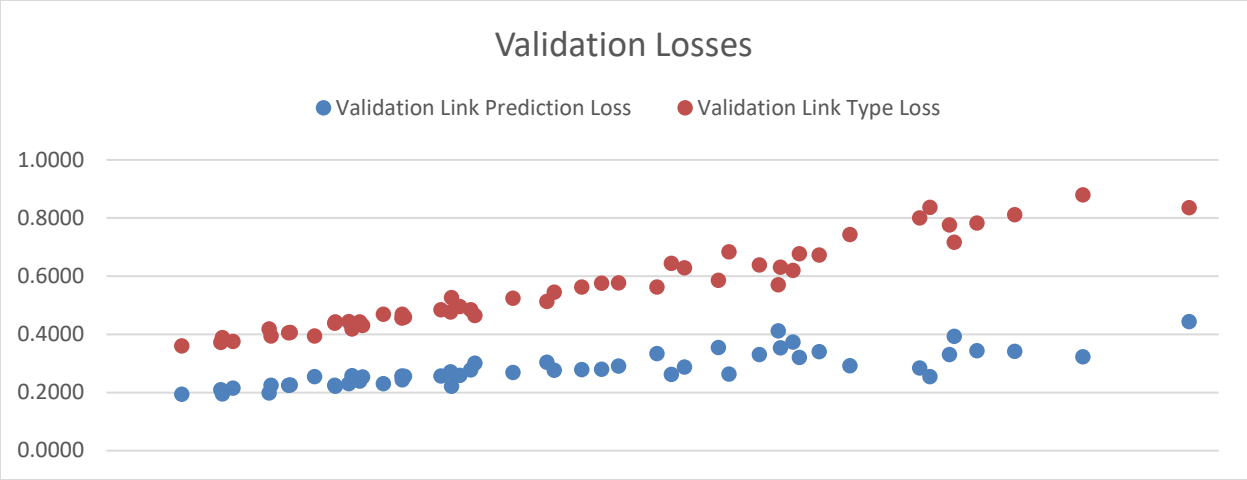


Figure 46 - Validation Losses Scatter Plot

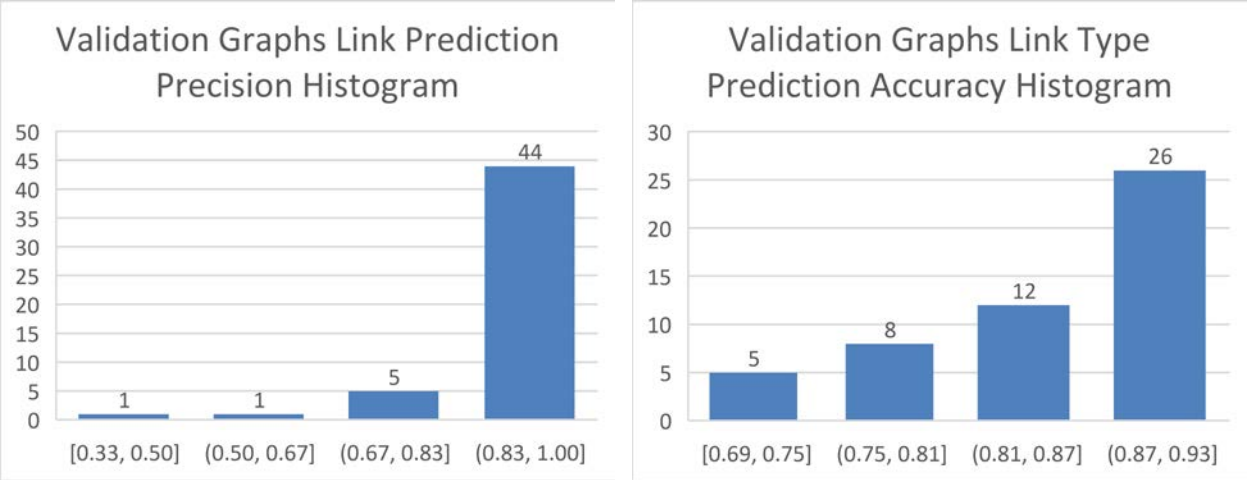


Figure 47 - Validation Graphs Precision and Accuracy Scores Histogram

The above metrics show very impressive performance for the final model being able to predict existing links with outstanding precision for True Positives. Overall. KGE GNNs are a versatile option for use in ML applications for Construction Scheduling

Second Alternate Approach: Fine-tuning an LLM

Transformer AI Models

To preamble the new approach it is first necessary to explain something called a Transformer Neural Network Model. A very simple definition is developing knowledge about context and sentence meaning through tracing relationships in sequential, or time-series data (Merritt, 2022). For example, “Pouring Concrete for Third Floor Slab”. The current working technique is attention or self-attention, to notice small ways that even distant data pieces in a sequence impact and depend on one another (Merritt, 2022). Transformer Models were first introduced by Google Researchers more than 5 years ago; and recently called Foundational AI Models by Stanford Researchers due to their powerful abilities to drive AI forward (Merritt, 2022).

This approach was discovered while navigating the Hugging Face Transformer Models Platform, “platform where the machine learning community collaborates on models, datasets, and applications”. (Hugging Face – The AI community building the future., n.d.) whereby the selection of an LLM to be trained and fine-tuned locally was made. The approach is widely known as Transfer Learning (TL) (Akinosho et al., 2020) and involves the reuse of models with strong classification or prediction performance with proper optimising and hyper-parameter tweaking.

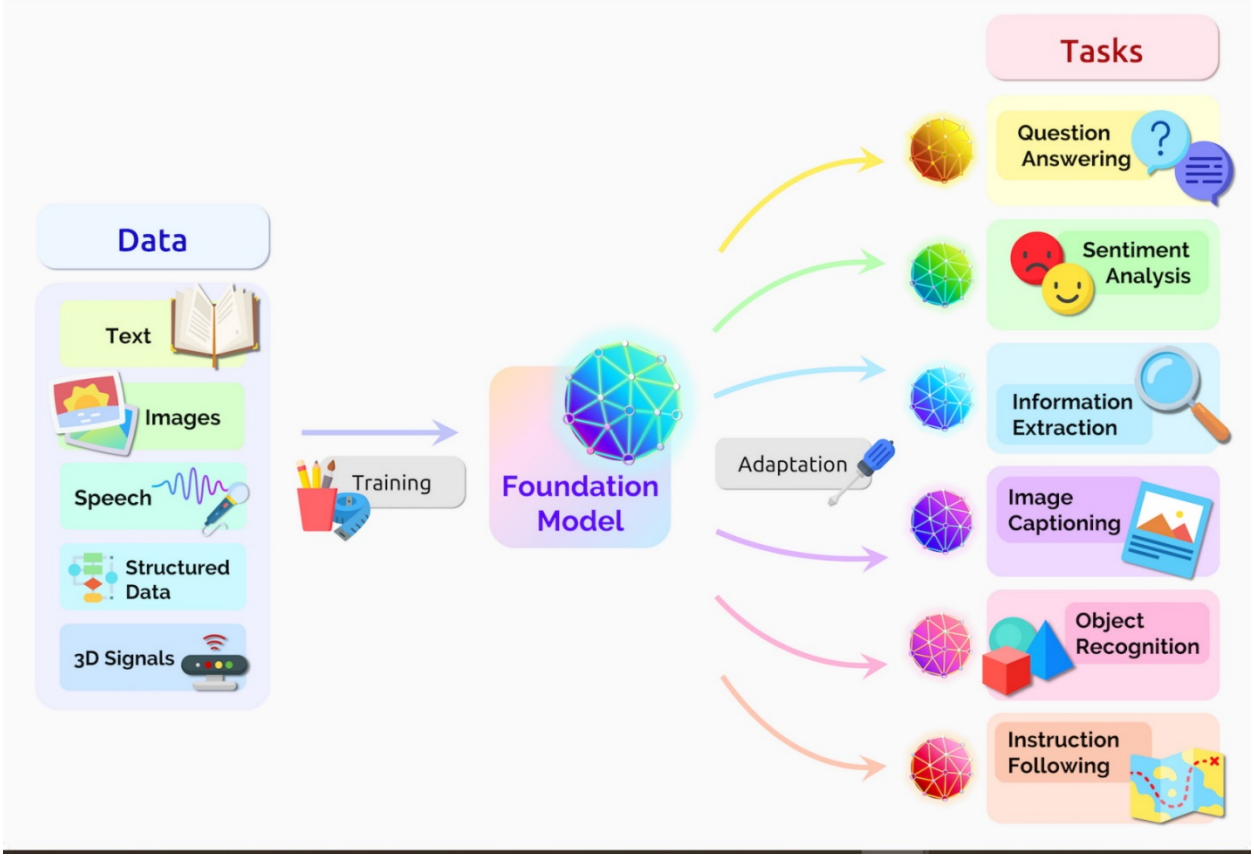


Figure 48 - Explanation of Transformer AI Models (Merritt, 2022)

Distil-BERT Pre-Trained Transformer

The criteria for model selection includes the ability to run on high-end consumer grade hardware. As most of the big models require more than 96GB of RAM to load the model before fine-tuning they were not selected simply for feasibility reasons. Consequently, the selection was made for a model called Distil-BERT. A Pre-Trained NLP Transformer Model that is created through distilling the original LLM Bidirectional Encoder Representations from Transformers, or **BERT** Introduced by Google in 2018 (Devlin, 2018). This model was chosen as a compact, and lightweight Transformer model that performs 60% faster, and retains more than 95% of BERT's performance according to the GLUE benchmark for Language (Sanh, 2019).

Model 2, Approach 2, Code Breakdown Structure

The Documentation Section for the model on the Hugging Face Platform (DistilBERT, n.d.) Included a use case on how to fine-tune the model for multi-label text classification whereby the model was fine-tuned to categorize hateful comments on the internet as one of multiple categories like hate, insult, threat, or a combination of multiple categories (Google Colab, n.d.).

The Google-Colab Notebook was followed according to these steps.

1. Importing Python Libraries and preparing the environment
2. Importing and pre-processing the domain data
3. Preparing the Dataset and Data loader
4. Creating the Neural Network for Fine-Tuning
5. Fine-Tuning the Model (Training on New Data)
6. Validating the Model Performance
7. Saving the model and artifacts for Inference in Future

Phase 1: Data Preprocessing

An important note about the data preprocessing for this model; since the notebook for this use case was done where the Model expects 2 Columns, A Text Comment Column, and a Label Column. The synthetic dataset had to be altered accordingly to fit those criteria. All Features were combined into a singular column where for example an activity is represented as:

```
30, Plumbing Drainage First Fix, 15.0, Munich, Office Building,
            2010, Winter, >50M, Rock, >1000<2000, >10
```

And according to the columns explained in the Dataset Breakdown Section above. To ensure that the model does not mess up in understanding the text in this column a special Token was added to the model to let it know how to split and tokenize the Sentences. This token is “, “

Additionally, the target columns (Predecessor ID and Dependency Type) were also combined and then turned into Binary Labels using the **Multi-Label Binarize** Module of Scikit Learn. This module turns each unique set of Labels into a separate class and each

row or activity, or sample has a single Binary Label of 1 according to its label and all other Unique Labels are 0.

For example, the above activity in Project 1 has a Predecessor ID and Relationship Dependency Type of

```
25, 35 and SS, FF
```

Where 25 is the ID for Plumbing Water Supply First Fix Activity, 35 is Exterior Walls Installation Activity, SS is a Start-to-Start Relationship with Activity 25, and Finish-to-Finish Relationship with Activity 35. Accordingly, this relationship set is saved as a unique Binary Label, and the Binary Label for This Activity is:

```
[58 Labels 0… 1, … Remaining Unique Labels until 312]
```

Indicating that [25, 35 and SS, FF] is unique label number 59 out of 312 labels.

Phase 2: Model Building, Training Loop, and Validation Loop Code

The model was first tested on only a singular Activity that has a large selection of different Predecessor ID and Relationship Dependency Types. This Activity is the Doors and Windows Installation. And it was selected as one of the Activities where the traditional classification model tested in the beginning does very poorly with only around 5% accuracy due to the randomness of the activity relations with other activities.

The model was instructed to do at first only 10 epochs, then 100 epochs over the dataset where a Loss Function calculates the model performance over the iterations. It is important to note 2 parameters for this model. Parameter 1 is the Train % of the whole dataset which was set to 95%. That still leaves the model with around 50 test projects which is enough for validation purposes. The second Parameter is the Dropout Rate, left at the default value of 0.1. This value indicates removing 10% of the nodes at random during training (Brownlee, 2019). This is an economical method meant to allow a single model to represent a combination of different models as a regularisation strategy for reducing overfitting and improving generalisation error in all types of deep neural networks (Brownlee, 2019).

Phase 3: Model Testing and Refinement

After the training process was done the model showed promising results of almost **92% Accuracy** for the selected activity for Predictions. The Loss Function over Iterations is plotted below.

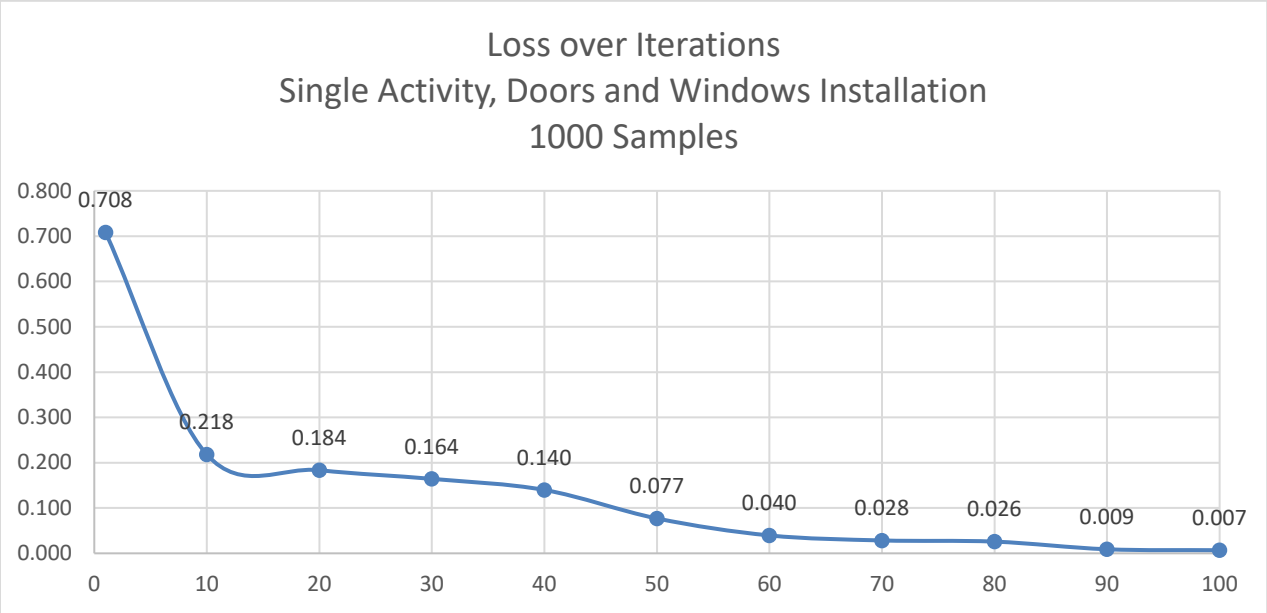


Figure 49 - Loss Over Iterations for Activity Relationships of Doors and Windows

Epochs	Validation Loss	Accuracy	F1 Score
10	0.05	0.0	0.0
100	0.006	0.922	0.941

Table 14 - Validation Metrics for Single Activity

After the Initial justification of the approach, and the fact that prediction accuracy is indeed contingent on number of epochs over the dataset (i.e. more epochs lead to better convergence) it was time to train the model on a massive dataset of around 1080 Projects where each project has 18 Activities for a total of more than 19,000 Rows.

It is worth noting that, the model learns each unique activity according to the number of samples it was given for it. As it learns from how the features for that specific activity affect its relation to the target prediction. Nonetheless, this step is meant to speed up the Training Process where the model can learn the relationships of multiple activities at the same time; therefore, allowing for Inference or Prediction of multiple Activities at the same time likely to be the real-world use case of such an algorithm (Prediction of a whole schedule at once).

Additionally, the same Training % was set as 95% and 5% for Testing and Validation. It is important to consider that Most Activities over multiple schedules are likely to have the same Dependencies and relations; therefore, more than 50 total projects for validation are a good indicator.

The model took over **12 Hours to complete 100 epochs on the training data**. The result, however, was an outstanding **99.997% accuracy**. While that result might raise suspicions that the model might be overfitting to the data. It is not, since that number means a wrong prediction of 0.003 or around 3 wrong predictions in 1000 rows. Since the dataset has around 19,500 rows that would lead to 19 wrong predictions. An excellent performance still, but not perfected. The Loss Function over Iterations is plotted below.

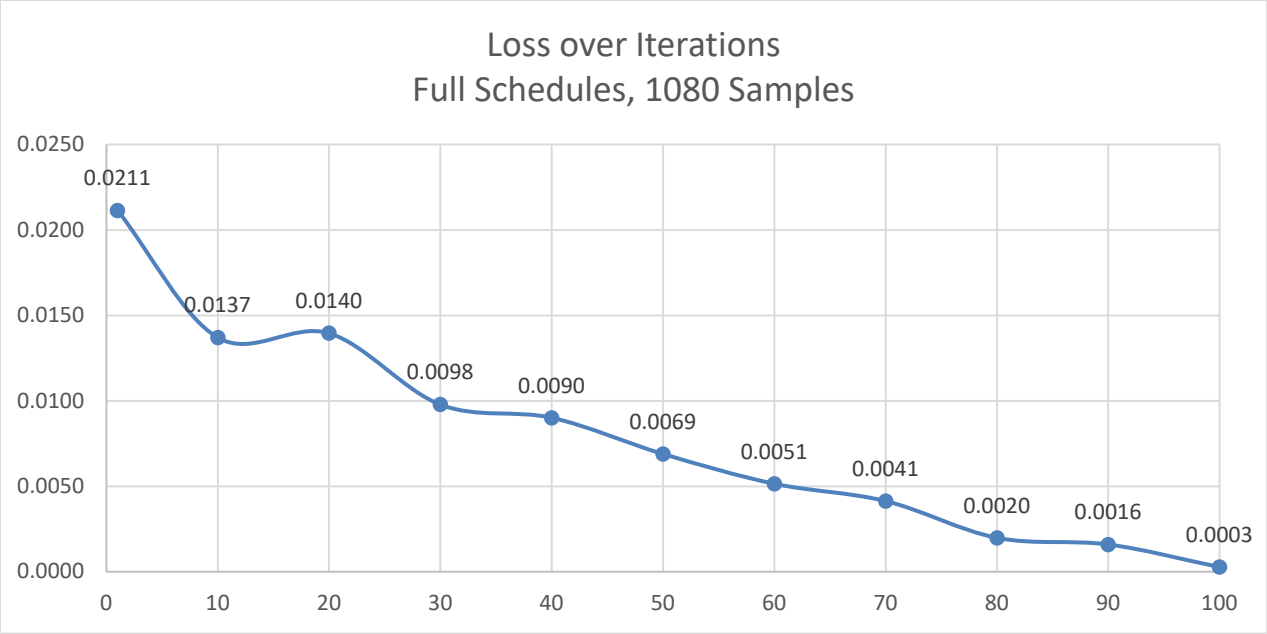


Figure 50 - Validation Metrics for Full Schedule Relations

Epochs	Validation Loss	Accuracy
100	1.08^{-5}	0.997

Table 15 - Validation Metrics for Whole Schedule Relations

It appears that the second approach for classification of activity relations is also considered a success with some hyperparameters that remain to be fully optimized.

Model 1: Most Probable Task List Generation

This is the final model developed in this Framework, but in real-world use case it will be the first model to be used for Schedule Generation. For this model the aim was to predict the most probable set of Schedule Activities that will take place given certain Project Features and Conditions. For example, if the Floor Area is larger than 2000 m^2. The Project is most likely to be split into 2 parts for effective construction. Additionally, if the Soil Type includes Clay that may mean the Water Level Table is high and Dewatering,

and or site wide retaining wall is needed. Finally, the Scope of works also affects the works on site as a Contractor, or subcontractor can be hired only for the Concrete Skeleton Execution for example.

For this model the same approach used in Fine-Tuning an LLM for Activity Relations Predictions was used since all the hard work was already done for that model but slightly modified for Activity Generation Prediction. Prediction of the most probable task list can be done in one of two ways, the first is Multi-Label Classification, where each set of project features has a set of Construction Activities. Each Activity is Encoded as a Label, and the model is tasked with prediction of all labels that most likely belong to that project's feature-set. The second approach is Multi-Class Classification since each Sample or Project will have a single block of Construction Activities linked to that sample.

Both approaches are valid for this task. Since the dataset will most likely be pulled as a task list from Planning Software, simple transposing it horizontally, or combining all tasks in a singular cell with a comma separator can be done easily. For this model, the second approach was used since the model expects a singular column on Target Labels; meaning that the Activity List will have to consolidated into a singular column for input to the LLM anyway.

Synthetic Database Breakdown

The same dataset outlined in the Duration Model and Activity Relations Prediction Models was again utilized was used in this Model as well. The following were the columns used to build the dataset:

1. Project #
2. Project Location
3. Project Type
4. Scope of Works
5. Soil Type
6. Floor Built Up Area (BUA)
7. Activity Names

For this dataset Multiple Activity Lists were created according to all combinations from different features of projects. And a lookup function is used to link the Feature Sets to the Activity Lists. For example, a project with the feature set

Scope of Works, Soil Type, Floor Built Up Area (BUA)

Concrete Works, and Rock, and >1000<2000

Equates to an activity list of

Project Start Mobilization, Earth Works, Concrete Skeleton Works, Embedded Piping Works

Another example of

Full Works/Turnkey, and Clay, and >1000<2000

Equates to

Project Start Mobilization, Retaining Wall Works, Dewatering Works, Earth Works, Concrete Skeleton Works, Plumbing Water Supply First Fix, Plumbing Drainage First Fix, Exterior Walls Installation, Exterior Plastering Works, Waterproofing for Wet Areas, Electrical Wiring First Fix, Ceramic Tile Works, Doors and Windows Installation, Elevator Installation, Interior Walls Installation, Interior Plastering Works, Painting Works, Electrical Wiring Second Fix, Plumbing Second Fix, Heating System Second Fix

The dataset contained Project Task Lists separated by the following table below.

Contractual Works	Soil Type	Floor Built-Up Area (BUA)
Full Construction	Sand	<1000
Finishing and MEP Works	Rock	>1000<2000
Concrete Skeleton	Clay	>2000<3000
Earth Works	-	>3000

Table 16 - Task List Generation Boundaries

The combination of all these factors yielded a possibility of 12 unique combinations (3 x 4) whereby in the case that the Floor BUA was in the third or fourth category (in other words more than 2000 sqm) The Construction Task List was split into two halves.

Model 1 Code Breakdown Structure

The model was trained using 100 epochs over the dataset. The Trained Model is then saved for Inference to predict new data.

The model performance was a perfect **100% Accuracy**. It was apparent that the dataset is simple in nature and can be easily Classified by a model since it is using a Lookup Function to classify and build the dataset initially, a very notable limitation of this model. However, in the construction of a real-world dataset the random and complex nature of real projects will simply avoid this problem unless explicitly simplified by the assigned team building the dataset.

Alternative Approach for Task List Generation: RNN-LSTM

Another approach is utilizing a Sequential Recurrent Neural Network that has the capability to hold sequential data in memory. This approach was targeted due to the existence of a similar approach by (*Amer and Golparvar-Fard, 2021*) in 2 literature sources.

The RNN is intended to take inputs of features that can be, but not necessarily, sequential in nature. And in return accurately predict a corresponding sequence of elements separated by time stamps. An example of this is Language to Language Translation

whereby the NN takes the first language text as input and provides the desired language output as a sequence of words. Another example from (*Amer and Golparvar-Fard, 2021*) is their work on understanding project scheduling activities. The authors took the text from previous schedules and manually annotated it into one of the following Categories: Action, Location, Object; for example, "*Pour Third Floor Slab.*" And the NN was tasked with Classification of each Item in the input sequence.

Recurrent Neural Networks (RNNs) are designed to process sequential data by maintaining a hidden state that captures information from previous steps. The Long Short-Term Memory (LSTM) architecture addresses issues like vanishing gradients and difficulty in capturing long-term dependencies by incorporating a memory cell and various gates that control the flow of information (*Jurafsky and Martin, 2024*). LSTMs are particularly effective for tasks that require modelling long sequences and capturing complex dependencies. They are commonly used in language processing tasks such as language modelling, part-of-speech tagging, sentiment analysis, sequence classification, and text generation (*Jurafsky and Martin, 2024*).

The same synthetic dataset was used for this approach. And the NN was tasked with prediction the whole sequence of tasks in a single prediction. There were 3 dataset formats tested and all of them yielded great success They are as follows:

1. Tasks in each Sample listed vertically. Repeating features per sample.
2. Tasks in each Sample list horizontally where each task is a column.
3. Tasks in each Sample list horizontally where unique tasks are column headers and each task in a sample is presented with a binary "1" or "0".

The first format is the main goal of this model since it highly resembles the data collection format from real task lists being copied to the dataset one after the other in a vertical format. All unique activities are then identified, encoded using Sci-Kit Learn's Label Encoder and stored in a Python Dictionary to be used to encode the whole dataset. A variable token was used to identify the Activity that signifies a new sample/sequence has started. In this case it was **Encoded Label '50'** which corresponded to **'Project Start Mobilization' Task**.

All Models used an input Feature Sequence of One-Hot-Encoded Features for each sample. For example, Project Location had 4 possible options, ***Berlin, Hamburg, Frankfurt, and Munich***. One-Hot-Encoding creates a Matrix of Binary 1, or 0 for each location and assigns the sample location to it. Therefore, sample n Project Location 'Berlin' will have a Matrix of shape [1, 0, 0, 0] Indicating. If the location was Frankfurt the Matrix would've been [0, 0, 1, 0] and so on.

This matrix is then fed to the model as input, the output Predictions are compared to the actual encoded labels using the Connectionist Temporal Classification (*CTC*) Loss Function and the Adam Optimizer for Optimization. This specific Loss function was chosen due to it being specific to RNN Sequences by being able to "*Calculate loss between a continuous (unsegmented) time series and a target sequence. CTCLoss sums over the probability of possible alignments of input to target, producing a loss value which is differentiable with respect to each input node*" *(CTCLoss — PyTorch 2.3 documentation, n.d.).* The figure Below shows the Loss over 2,500 Epochs. Model Validation on Test Set Sequences is included in the Appendices Link. The number of Epochs was chosen arbitrarily at 2500 since the Dataset Format creation required a large amount of time to separate the tasks, transpose them vertically, and repeat the features for each sample. Therefore only 500 Samples were made from the total 1000 initially created, and thus the number of Epochs was increased to account for the smaller dataset.

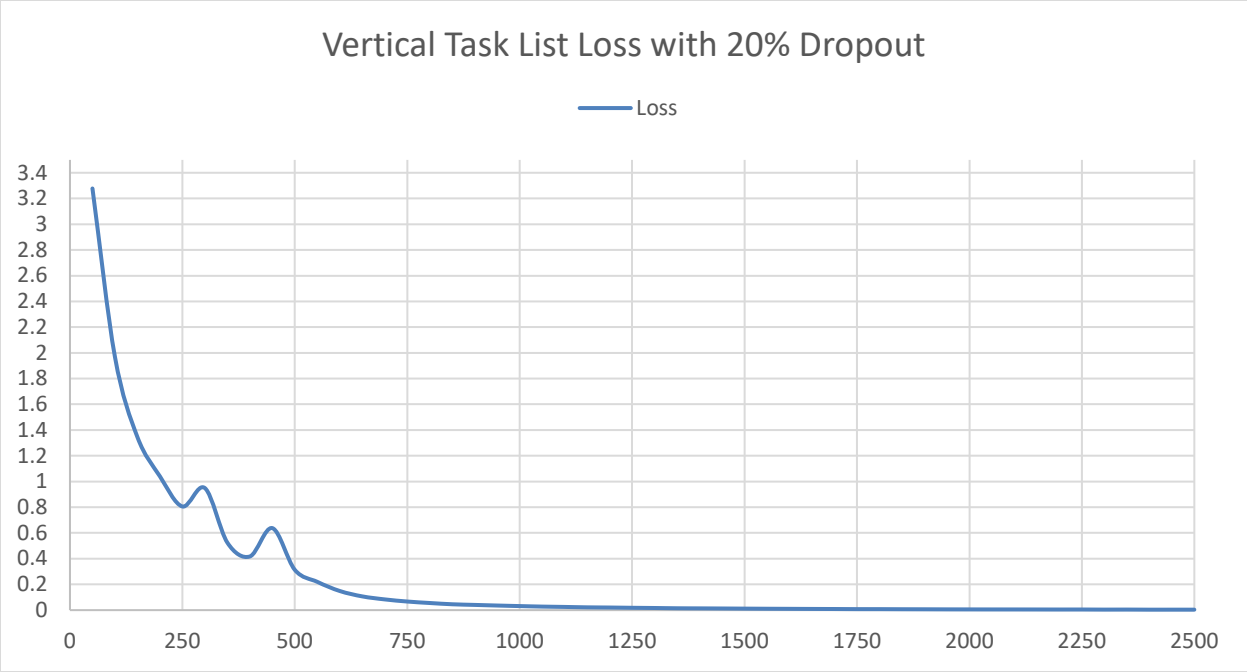


Figure 51 - Loss over Epochs for Vertical Task List Data Format

The second and third Dataset formats were explored as well since it may be used in building a real dataset with transposed activities list where each sample only takes up one row for being visually much easier to understand. The only difference between them is the approach of predicting a Binary Existence or None thereof for a Task in a Sequence. For the second Dataset Format, an additional Encoded Label for empty cells 'Nan' was created to indicate the end of the sequence. Below is a graphed comparison of the Loss over Epochs, and like the previous Model, Validation on Test Set Sequences is included in the Appendices Link. The Loss Function for both Datasets was Cross Entropy, and Binary Cross Entropy respectively.

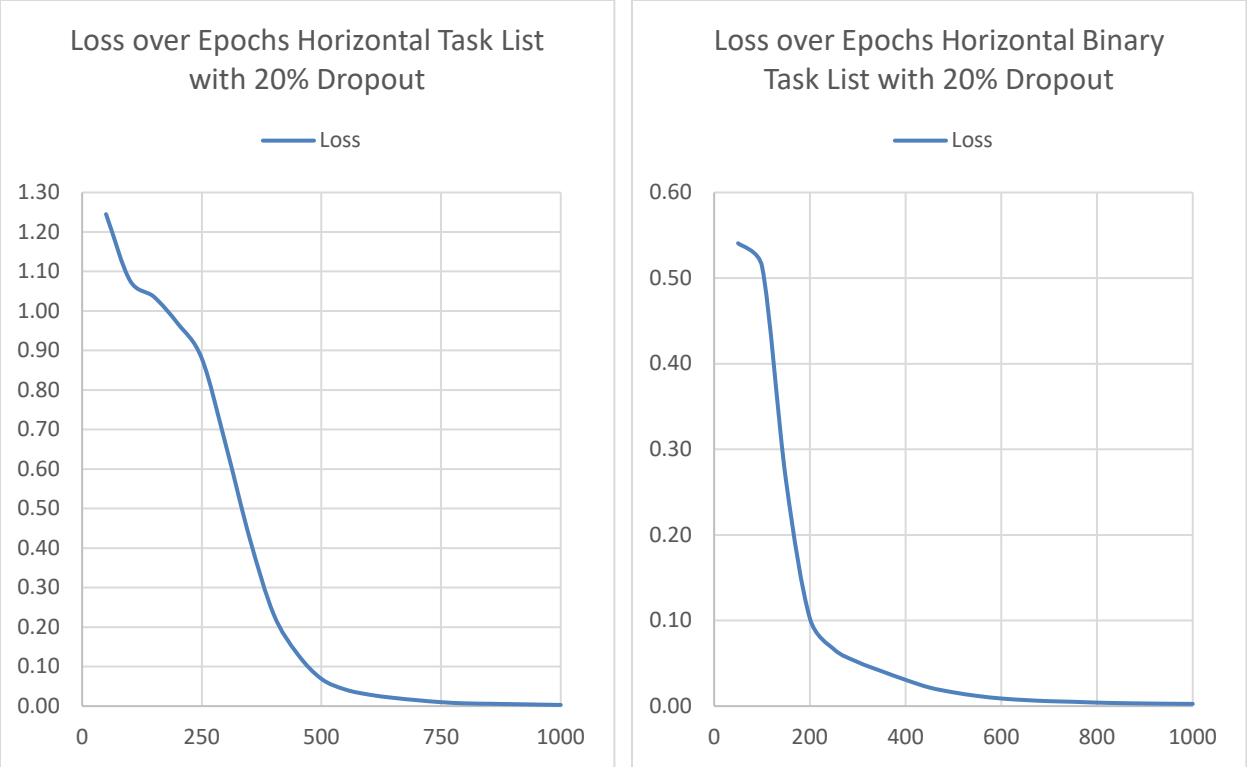


Figure 52 - Loss over Epochs for Horizontal Task List Data Format

Important note. An optimization approach to help the model converge faster on smaller datasets called Teacher Forcing was experimented with but unfortunately was not completed successfully due to a problem with utilizing the Total Loss for a single loop over the training dataset to update the model weights using the "backwards" Method.

Model 4: Webapp Deployment

In developing the model and selecting the coding language, deployment ability was heavily considered and researched. The deployment aspect was covered by (Sanni-Anibire et al., 2021) briefly through two main methods, a standalone application, or an add-on to popular project management tools.

In the case of this thesis's model, for an early rudimentary deployment, a standalone application can be extracted from the code as it already has user input features and expandability in mind during development. For this purpose, the deployed application be extracted as an executable file for windows platform (.exe), or a Graphical User Interface

(GUI) web-based application server that is hosted through a local computer or a cloud network. Nevertheless, the future commercial vision and application is best realized through development of an addon for popular scheduling applications like Microsoft Project.

Full Framework Integration

The final piece of the proverbial puzzle is the combination of all models after training to be used in the creation of an actual schedule file. This is done through the creation of an Inference File that loads the trained models and gives them the data required to make predictions. The following process explains the full framework integration.

It is worth noting that the Inference File for each model is already developed alongside it but the coherence of the models working together in a singular inference file is a step that was not taken due to time limitations. It is, however, a minor step all things considered. Below is the Code Breakdown Structure and further explanations of the Inference Files for each Model.

Development File 1: Model Inference for Most Probable Activities Generation

An important thing to mention is the Model Inference Output for each target is a list of numbers called Logits. A classification model provides a vector of raw (non-normalized) predictions, which are often sent to a normalisation function like SoftMax function. The SoftMax function then produces a vector of (normalised) probabilities, one for each conceivable class (Google for Developers, n.d.). The value of the Logits is a number for each unique label. If the number is a positive value, the model will likely assign this label as a prediction to this target, if negative it will likely not assign this label.

The Inference Schedule Features is fed to the Loaded Model and the Output Raw Tensor Logits were obtained. These Logits were first transformed back to Binary Labels using a transform Sigmoid Function that converts the predictions to a number compared to a standard threshold of 0.5 or larger. Then using a saved instance of the Label Binarize on the Original Dataset Training Labels to convert the Binary Output Labels to their Original Labels. Finally, the output data is then saved as a Spreadsheet file where the tasks are split and transposed using the following formula for Entry to the Relationships Model:

```
=TRANSPOSE(TEXTSPLIT(B2, ", "))
```

Where B2 in this case is the Activity List Cell

Development File 2: Model Inference for Activity Relationships Prediction

Model Inference using a GNN.

In the case of using a GNN for Activity Relationships, the data input to the model is an activity list as either the source; and the GNN will predict the Target and Link Type (Activity Successors) or the list as the target and the GNN will predict the Source and Link Type (Activity Predecessors). In either case, the tabular data will first be converted to graph data using Network-X, then Network-X will be used to construct all possible

combinations of Links between all activities as number of nodes x (number of nodes – 1). As a reminder, this approach is needed since the GNN model was trained using Negative or Fake Edge sampling to detect the most probable Links. Therefore, the GNN will get an input of all possible Links and it will filter out the most probable ones using its trained weights.

After prediction of most probable links and their corresponding types, the row data is normalised through a sigmoid function to turn them into binary existing or non-existing (1 or 0) Links. They are then compared to the original Adjacency Matrix created by Network-X to figure out which nodes are connected. Finally, the source node and target node are separated and saved alongside their link type as a .CSV File

Model Inference using a Fine-Tuned LLM.

The new schedule task list and features are loaded into the LLM saved model. And the raw predictions are obtained. Conversely, before translation of these predictions into their results there are two approaches to do so. The first is using a SoftMax Activation function that turns all predictions for a certain activity into a combined probability of 1. Then the highest probability of those is selected as the prediction. Another method is using the Sigmoid Activation Function with a 0.5 threshold. The benefit for this approach is that for some activities, predictions maybe very close together in multiple classes, and instead of choosing only the highest prediction, the model can output a singular label for the targets of low confidence, or multiple labels for other activities. The author views this as a positive outcome since not all activities will have the same constraints every time, for some of the more open activities, a model providing multiple options can help Planners decide between different scenarios using each option. Finally, the predicted labels are then inversely transformed to their original values using a saved instance of the Multi-Label Binarize Module.

Phase 3: Model Testing

The Inference Dataset was fed to the Loaded Model and the Output Raw Tensor Logits were obtained. These Logits were first transformed back to Binary Labels, then using a saved instance of the Multi Label Binarize on the Original Dataset Training Labels to convert the Binary Output Labels to their Original Labels. Finally, the Inputs for Inference

and Output Labels are saved in a singular Dataset with 2 columns for further use like importing to another Planning Software.

Development File 3: Prediction of new Data

Once the task list has been generated it is also loaded into the Durations Model for optimization. One of the added alterations to the ML prediction algorithm is the inclusion of Optimistic and Pessimistic durations. The calculation for these durations was made using the trained model's Median Absolute Error (MAE, in days) Metric defined as median of the absolute value of prediction errors (Deepchecks, 2023) rounded to the next integer day as the author believes it represents the best lower and upper bounds estimate of the predicted duration.

This Page was intentionally left blank as a separator between chapters.

Models, and General ML Limitations

It is widely known in the AEC industry that every project is, for better or worse, considered unique in terms of actuality of events. Therefore, each project also has a unique work breakdown structure (WBS), making scheduling data very unstructured. This fact restricts and limits the usage of ML effectively (Alice Technologies, 2018). Additional limitations are also discussed as follows.

Created Framework General Adaptability

The duration model is likely very adaptable at studying almost any type of dataset since it reads the columns of the dataset and allows user arrangement of Prediction Targets and Features. In theory it can be used to study any kind of regression relationship by appropriate selection of feature and target columns. Conversely, the Activity Relations and Generation Models are considered more fixed function than adaptable for other tasks in the AEC Industry since they are more locked down due to the lack of more specific coding knowledge and following the specified example.

Incompatibility with other trained models (no one-size-fits-all)

The purpose of this Thesis is to provide a holistic complete framework for generation of entire scheduling data using ML. Nevertheless, it is important to note that the approach is not flawless, and each company has different needs. Related to the first point discussed in this section, the automated scheduling can rarely be addressed through a general trained model. Each problem must be handled independently using a model that has been specially trained for that purpose and the company's historical data and relevant feature.

A related issue is linking the ML Model input and output to widely used Planning Software like MS Project and Primavera P6. This issue was partially taken into consideration throughout the development of the ML model for this thesis whereby the outputs are CSV file that be re-imported back to the scheduling software. However, the issue of the database creation and enabling the ML algorithm as a plugin or standalone application needs further investigation.

Talent acquisition hurdles with AI and AEC Industry Experience

One of the other challenges is simply related to the human talent with both AI and AEC Industry knowledge. It was seen in the Research Project by (Laing O'Rourke Centre for

Construction Engineering and Technology, 2019) that a collaboration between a scheduling software company, an industry giant contractor, and a major university was needed to pursue an automated scheduling project. Ergo, it stands to reason the difficulty to find engineers with sufficient knowledge in both domains since they are very much mutually exclusive. (Abioye et al., 2021) proposed increased governmental spending and increasing its investment in science, technology, engineering, and mathematics (STEM) education to address this problem. Furthermore, increased research efforts in general between AI and AEC experts in academic domains to fuse ideas and birth new breakthroughs that satisfy demands of the sector.

Initial and running costs of AI Systems Integration

Relating to the above point in terms of the likely compensation for Engineers with both AI and AEC Industry knowledge, there is also the issue of system implementation. The costs of the AI system can vary greatly depending on the requirements and use cases. The most expensive implementation would of course be a state-of-the-art implementation with powerful hardware to cut the Model Training time. Another solution to avoid the physical infrastructure is leasing cloud solutions like Microsoft's Azure ML or Amazon Web Services (AWS). The actual cost of applying deep learning is difficult to determine because it is dependent on skill needs and training gear. (Akinosho et al., 2020) advised for more research to identify cost-effective adoption methods in the sector. As the Author mentioned, the Activity Relations LLM Model took around 12 hours to complete training on a large dataset and utilizing the Graphics Processer for accelerated learning.

Blackbox Predictions and Gamblers Fallacy

Interpreting the outcomes of ML and DL models like ANNs remains problematic since the optimization of prediction difference function, typically called backpropagation, makes it impossible to relate and associate inputs and outputs (Fitzsimmons et al., 2022). One of the ways to improve the usability of ML models in Scheduling is through cutting up massive project schedules into smaller sub-schedules in a hierarchical manner (Wolejszo, 2020). While adapting the ML algorithm and dataset for that sub schedule area. Allowing for more specialized algorithms (for example Substructure Works) overarched by another specialized master schedule algorithm for optimizing major

phases' durations. Utilizing each scheduling algorithm when as the work's execution date approaches.

Another important limitation of predicting future values is the knowledge that the future is, eventually unpredictable in nature. The near complete dependence on prediction outcomes is known as The Gambler's Fallacy, or Monte Carlo fallacy. It is defined as erroneously thinking the outcome of a prior event or set of events makes a specific random future event more, or less likely to occur (Kenton, 2023). In the case of ML assisted prediction, there two sides of the coin to discuss. First, is that humans maybe sceptic about an ML algorithm output due to their bias that a certain streak of events may occur due to historical figures while the algorithm outputs a different, be it slight or complete, picture. In this scenario one of the recommended courses of action can be to utilize ML to reduce human inflicted gambler's fallacy as written by Stanford University Fellow Dr. Lance Eliot (Eliot, 2021) specifically related to Court Judges as ML can relate multiple sets of data for underlying relations much better than humans.

The second problem is related to the ML algorithm itself falling victim to Gambler's Fallacy. For example, a person, group, or a company might fall victim to Gambler's Fallacy in predicting future stock prices, even though the stock market is intrinsically volatile and unpredictable. To avoid Gambler's Fallacy, it's critical to implement ML predictions with an awareness of chance, unpredictability, data, and model limits. To illustrate, the predictions are only as good as the dataset being used, and as a generalization it is best to have contingency procedures that avoid forming predictions of future outcomes based entirely on previous happenings.

Data Collection, Pre-processing, Privacy, and Ethics

The process of finding relevant data, pre-processing it, and transforming it into useable format for the AI models is a gigantic task in its own right. As it was already briefly mentioned that certain approaches like the Matrix Dataset approach for activity relations, or the RNN formatting for Task List Generation requires a long amount of manual labour to enable the Model to learn effectively. Which could deter certain companies from utilizing the approach if the initial investment costs are prohibitive and with a fuzzy return

on their investment; that is to say the return is in terms of time efficiency and not financial in nature.

With multiple researchers exploring this area. The training process of ML systems necessitates a large amount of data gathering and input. Input data ought to be gathered from multiple sources to reduce the chance of bias, including but not limited to, relevant prices of labour, materials, and equipment (theconstructor, 2022). The greater number of samples and features, the usually more accurate predictive performance of the Model. Data preprocessing can even be accelerated through specific AI algorithms for said problem. While AI Algorithms work best with larger datasets, size of the dataset may not always be the problem, it can be accessing data for a specific domain especially with concerns over data privacy, ethical implications & law problems with General Data Protection Regulations (GDPR) requirements (Akinosho et al., 2020). For the purposes of dataset size and limited datasets in general augmentation techniques with reasonable adjustments to the existing dataset can be used such as altering activity duration of a sample to create new samples. However, data augmentation might result in the loss of critical data or outliers required for training (Akinosho et al., 2020).

Recommendations for Future Research

Framework Specific

One of the first recommendations for future research efforts is optimization of the created models for more efficient training and inference. This includes for example the optimal number of epochs given a certain dataset versus the loss function improvement, and the best learning rate for the loss function optimizer. Additionally, the utilization of real-world datasets, perhaps built on top of the fully synthetic one created for this Thesis is also a top priority for improved credibility in the aim to expand the model to a country wide applicable prediction model for example, or even a more international model. Finally, the models and datasets can be further enriched with multiple features allowing expandability in prediction of other project aspects like Cost, Material Quantity Variances, Risk, Resources, and more.

From an overview, the GNN seems to be the most suitable for Construction Scheduling due to the way scheduling software handles task input through Activity-on-Node Network Diagrams. This Machine Learning Framework needs to be studied in more details with regards to Construction Planning

Incorporation of Fuzzy Logic

Fuzzy logic is a type of reasoning that works on imprecise or uncertain data. Unlike classical Boolean logic, which utilises just two truth values (true or false), fuzzy logic accepts a range of truth values ranging from 0 to 1. As a result, it is well-suited to portraying and reasoning about things that are difficult to explain in black-and-white terms. (Google Bard LLM). Fuzzy logic may be utilised to build expert systems capable of making judgements in difficult conditions where in a singular optimized duration or relation for an activity may not be feasible. Fuzzy Logic AI Algorithms can be implemented as an advancement to the Program Evaluation Review Technique (PERT) to include confidence intervals for each activity duration predicted as an example.

Search Algorithms for Optimal Scheduling (Alice Technologies, 2018)

Optimum construction scheduling by leveraging the power of search engines and "learning" all aspects of data in its application. Search algorithms are simply put solvers.

They analyse data and answer problems in a problem space. These algorithms examine many alternatives and limitations, employing a variety of methodologies to identify optimal pathways and, ultimately, solve issues. They utilize three main techniques known as Depth First or Breadth, or Width First. The third technique is the one of interest in Scheduling known as Heuristics. The algorithm rates many choices in this method. Information may be offered to enable an educated decision on which branch to pursue by rating the possibilities in a list.

The notion of utilising lists is a universal topic regardless of approach. A fresh framework is in dire need of a more advanced development for assigning the start and end dates to tasks to employ search in scheduling. The existing scheduling systems are based on notions from the 1960s and, as such, must be reconsidered. In this approach Four lists may be employed to define work in construction timetables, akin to Toyota's famed kan-ban technique. The initial list is mainly an activity level to-do list is the initial type of list. A to-do list allows organization of activities, limitations, related durations, connections, and resources. These lists establish the building plan's rules. The can-do list is next. The approach resolves restrictions in this case. For example, if one activity must begin before another, the first one cannot begin until the essential resources are available. When such resolutions are added together, they form a to-do list. Thirds is a “doing list” for future-state generation, which is the road towards the final list or the “done” list. This task's start and finish times are specified. The resource pool is then used to assign resources. They then go to the next job on the to-do list. While artificial intelligence evaluates the many sequence options, tasks are picked to go from a can-do list to a doing list.

The final list is the done or completed list. When all the things on the to-do list have been completed, the Schedule is complete in the most optimized way according to all restrictions.

Optimization of Overall Schedule Duration with 4D Simulation Software for Feasibility

The implementation of this optimization technique has a prerequisite of developing the “Classification of Activity Dependencies” first to obtain the complete schedule. Afterwards, the obtained schedule can be fed to a BIM simulation program (i.e. Navis-Works) to model the construction sequence for determining the feasibility. (Chen et al., 2013) proposes a

similar concept made by utilizing Discrete Event Simulation (DES) to “model and analyse construction processes, including the overall project duration as well as resource utilization, and what-if analyses”. Finally, a reinforcement learning ML algorithm can be developed to reiterate on the schedule duration-feasibility permutations until the optimum schedule is found.

General Related Research

Further Integration with AI Multimodal LLMs

This is a short recommendation for future research ability meant to reduce the development load requirement, experience needed to code proper data preprocessing and ML Algorithms in general. LLMs with multimodal capabilities, are LLMs that can process more than text alone. They can usually comprehend and create videos, images, sound, and further composition capabilities for larger segments of text (Meskó, 2023). The idea behind this recommendation is to utilize the capabilities of these models for interactive planning and scheduling, as well as historical schedules analysis for further optimization; for example, utilizing OpenAI’s GPTs feature allowing the creation of task specific versions of ChatGPT-4 (Introducing GPTs, 2023) for interactive planning and optimization.

Automation of Site Management Data collection and analysis

Regular project progress tracking is one of the most error prone tasks due to the involvement of multiple human inputs alongside the process chain. Even the most experienced teams will struggle to manage a huge construction project. Handling worksheets, schedules, and written communications from several subcontractors over the course of a lengthy project is bound to be overwhelming (Buildots, 2023). This is further compounded by frequent communication on site, which necessitates in-person trips that consume significant time. When dealing with huge projects, various teams, and tight schedules, data gaps are unavoidable (Buildots, 2023).

Automation, or at the very least partial automation of site progress tracking from a Project Management Documentation view is one of the areas AI, and especially Large Language Models (LLMs) can assist. The elementary outline of the idea was conceived while watching a video of a creative professional explaining their use of an online platform that

creates “Triggers” based on a defined set of “Actions”. To provide an example based on this creator, they were using this platform such that when they upload a voice note to a specified Cloud Storage folder (Trigger), a LLM can transcribe, summarize, and organize notes automatically (Action). And finally, the transcription and summary are uploaded on another platform (Action 2). This methodology can be adapted in a similar fashion to automate more mundane tasks on construction site such as daily progress reports. It is possible to utilizing a similar platforms' versatility in automating construction site data collection for management purposes and automated Report making. To substantiate, Construction firms utilizing using computerised tools were able to decrease costs by around 5%, and likewise 5% faster work rates due to around a quarter more labour productivity while also decreasing overall labour spending by the same amount (Buildots, 2023).

This Page was intentionally left blank as a separator between chapters.

Conclusions

The use of AI in the AEC industry is only set to stretch forth in all aspects in the future, from feasibility and project conceptual design to construction and facility management. It will significantly alter many aspects of the building process. Companies in this sector ought to plan on taking advantage of it sooner than later as initial supporters of this digital revolution will almost certainly generate growth and attain a competitive advantage enabling company leaders to choose the course and enjoy the most rewards (Srivastava, 2023) as it allows for better data organization and by consequence improved analysis and insights (AutoDesk, Ellis, 2023). There is a positive and linear correlation between publications and articles about ML in AEC Management processes. Despite the sharp decline in the last 3 decades where only 19 studies were conducted on the matter (Van and Quoc, 2021), a significant growth of the literature was published between 2014 and 2020 signifying the growing interest. Which can certainly encourage additional investigations on this issue for future study (Van and Quoc, 2021).

It is nonetheless quite unfortunate a majority view the AI of the future as an entity lacking or not needing human interaction for the most part. Indeed, a lot of individuals who may work in highly automatable jobs fear AI exponential advancements will only have a negative impact on the job. 32% of US workers believe AI would harm rather than aid the employment, while just 13% feel the opposite as shown in Pew's Research poll (AutoDesk, Ellis, 2023). JBKnowledge revealed from their Sixth Annual Report in 2017 covering Construction Technology that companies view the necessity of automation and technology as critical for growth. In particular, the reports states "If more construction professionals understood the work tasks that automation and Artificial Intelligence technologies can augment and enhance, they might focus less on the tasks they will 'replace.'" (Alice Technologies, 2018). Adding to this is the reluctance of most small-scale local contractors to upgrade their scheduling approach (Salleh, 2009). Therefore, seemingly complicated and Blackbox techniques like ML are a tough proposition.

This thesis covered only a small portion of the AI use cases that was specifically targeted at initial project planning and scheduling techniques delivering on the concept for a more technologically inclined and more automated planning system. An important principal to

highlight here also is that the acts of Planning and Scheduling should be thought of as two distinct tasks. Whereby Project Planning is an umbrella term that considers Scheduling within (Wolejszo, 2020). The AEC industry is hastily crossing the bridge of higher technology integration in the building process. Several research investigations have advanced our comprehension of human knowledge and showed the prospective benefits of employing such disruptive innovations (Karmakar and Delhi, 2021). There are more hurdles to come, for instance many technological systems are not adequately well-integrated with other systems or software packages resulting in data disconnect. As a result, there is a considerable need for more comprehensive in this area and creation of methodological and technical frameworks (Karmakar and Delhi, 2021).

The incorporation of AI scheduling in the AEC sector has significant potential as it can support expediting the scheduling process with enhanced accuracy, less inherent risk, and therefore greater chance of maximizing projected revenues (theconstructor, 2022). Predictive analytics can be used for forecasting fluctuations and allowing proactive measures instead of reactive ones. Thus, reducing the chance of unanticipated delays.

References

1. What Is a Knowledge Graph? | IBM (n.d.). Available at: https://www.ibm.com/topics/knowledge-graph.

2. Brody S, Alon U and Yahav E (2021) How Attentive are Graph Attention Networks? Available at: https://arxiv.org/abs/2105.14491.

3. Hamilton W, Ying Z and Leskovec J (2017) Inductive Representation Learning on Large Graphs. Available at: https://papers.nips.cc/paper_files/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html.

4. Karagiannakos S (2021) Best Graph Neural Network architectures: GCN, GAT, MPNN and more | AI Summer. Available at: https://theaisummer.com/gnn-architectures/#inductive-vs-transductive-learning.

5. CTCLoss — PyTorch 2.3 documentation (n.d.). Available at: https://pytorch.org/docs/stable/generated/torch.nn.CTCLoss.html.

6. Zhang S, Tong H, Xu J, et al. (2019) Graph convolutional networks: a comprehensive review. Computational Social Networks 6(1).

7. Jurafsky D and Martin JH (2024) Chapter 9: RNNs and LSTMs. Available at: https://web.stanford.edu/~jurafsky/slp3/9.pdf (accessed 22 April 2024).

8. NetworkX — NetworkX documentation (n.d.). Available at: https://networkx.org/.

9. Merritt R (2022) What Are Graph Neural Networks? | NVIDIA Blogs. Available at: https://blogs.nvidia.com/blog/what-are-graph-neural-networks/.

10. Sánchez-Lengeling B, Reif E, Pearce A, et al. (2021) A Gentle Introduction to Graph Neural Networks. DOI: 10.23915/distill.00033.

11. RMIT international university (n.d.) Network systems or PERT charts. Available at: https://emedia.rmit.edu.au/dlsweb/Toolbox/buildright/content/bcgbc4007a/03_dev

elop_track_revise/03_network_systems/page_001.htm (accessed 12 March 2024).

12. Sanni-Anibire MO, Zin RM and Olatunji SO (2021) Developing a machine learning model to predict the construction duration of tall building projects. Journal of Construction Engineering, Management & Innovation 4(1). Golden Light Publishing: 22–36. DOI: 10.31462/jcemi.2021.01022036.

13. Google for Developers (n.d.) Machine Learning Glossary. Available at: https://developers.google.com/machine-learning/glossary.

14. Brownlee J (2019) A Gentle Introduction to Dropout for Regularizing Deep Neural Networks. Available at: https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/.

15. Google Colab (n.d.) Fine Tuning DistilBERT for MultiLabel Text Classification. Available at: https://colab.research.google.com/github/DhavalTaunk08/Transformers_scripts/blob/master/Transformers_multilabel_distilbert.ipynb (accessed 16 February 2024).

16. DistilBERT (n.d.). Available at: https://huggingface.co/docs/transformers/model_doc/distilbert.

17. Sanh V (2019) DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. Available at: https://arxiv.org/abs/1910.01108.

18. Devlin J (2018) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Available at: https://arxiv.org/abs/1810.04805v2.

19. Merritt R (2022) What Is a Transformer Model? | NVIDIA Blogs. Available at: https://blogs.nvidia.com/blog/what-is-a-transformer-model/.

20. Hugging Face – The AI community building the future. (n.d.). Available at: https://huggingface.co/.

21. XGBoost Documentation — xgboost 2.0.3 documentation (n.d.). Available at: https://xgboost.readthedocs.io/en/stable/.

22. Shi AAK (2023) Gradient Descent vs. Gradient Boosting: A Side-by-Side Comparison. Available at: https://towardsdatascience.com/gradient-descent-vs-gradient-boosting-a-side-by-side-comparison-7067bb3c5712.

23. Khushaktov F (2023) Introduction Random Forest Classification By Example. Available at: https://medium.com/@mrmaster907/introduction-random-forest-classification-by-example-6983d95c7b91.

24. Deepchecks (2023) What is Mean Absolute Error | Deepchecks. Available at: https://deepchecks.com/glossary/mean-absolute-error/#:~:text=Mean%20Absolute%20Error%20(MAE)%20is,effectiveness%20of%20a%20regression%20model.

25. scikit-learn (n.d.) sklearn.multioutput.RegressorChain. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.RegressorChain.html#sklearn-multioutput-regressorchain.

26. scikit-learn (n.d.) 1.11. Ensembles: Gradient boosting, random forests, bagging, voting, stacking. Available at: https://scikit-learn.org/stable/modules/ensemble.html#ensembles-gradient-boosting-random-forests-bagging-voting-stacking.

27. scikit-learn (n.d.) Choosing the right estimator. Available at: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html.

28. scikit-learn (n.d.) 1.13. Feature selection. Available at: https://scikit-learn.org/stable/modules/feature_selection.html#l1-based-feature-selection.

29. Dodge Y (2008) Regression Analysis. In: Springer eBooks. 1st ed. NewYork, NewYork, United States: Springer, pp. 450–452. DOI: https://doi.org/10.1007/978-0-387-32833-1.

30. Eliot LB (2021) Using AI To Overcome Gambler's Fallacy That Pervades Human Judges. Social Science Research Network. 1 January. DOI: 10.2139/ssrn.3992208.

31. Venkatasubramanian V Data Science & Analytics Karthik (2021) AI and machine learning. Available at: https://www.oracle.com/construction-engineering/construction-intelligence-cloud/using-ai-and-machine-learning-to-predict-construction-schedule-delays/ (accessed 20 December 2023).

32. Kenton W (2023) Gambler's Fallacy: Overview and Examples. Available at: https://www.investopedia.com/terms/g/gamblersfallacy.asp.

33. Amer F and Golparvar-Fard M (2019) Automatic Understanding of Construction Schedules: Part-of-Activity Tagging. Computing in construction. DOI: 10.35490/ec3.2019.196.

34. Bang S, Aarvold MO, Hartvig WJ, et al. (2022) Application of machine learning to limited datasets: prediction of project success. Journal of Information Technology in Construction 732–755. DOI: 10.36680/j.itcon.2022.036.

35. Code Academy Team (n.d.) The Machine Learning Process. Available at: https://www.codecademy.com/article/the-ml-process (accessed 17 December 2023).

36. Tajziyehchi N (2021) A Machine Learning-Based Approach for Predictive Analysis of Cost Growth in Heavy Industrial Construction Projects. Available at: https://prism.ucalgary.ca/items/517f9db9-bfd5-407f-bdf8-106aca96e2eb.

37. Alekseytsev A and Nadirov SH (2022) Scheduling Optimization Using an Adapted Genetic Algorithm with Due Regard for Random Project Interruptions. Buildings 2051(12). DOI: 10.3390/buildings12122051.

38. theconstructor (2022) Harnessing AI To Revolutionize Construction Scheduling. Available at: https://theconstructor.org/artificial-intelligence/harnessing-ai-to-revolutionize-construction-scheduling/568900/ (accessed 12 December 2023).

39. Chen SM, Griffis F, Chen P, et al. (2013) A framework for an automated and integrated project scheduling and management system. Automation in Construction 89–110. DOI: 10.1016/j.autcon.2013.04.002.

40. Introducing GPTs (2023). Available at: https://openai.com/blog/introducing-gpts (accessed 12 December 2023).

41. Meskó B (2023) The Impact of Multimodal Large Language Models on Health Care's Future. Journal of Medical Internet Research e52865. DOI: 10.2196/52865.

42. Akinosho TD, Oyedele LO, Bilal M, et al. (2020) Deep learning in the construction industry: A review of present status and future innovations. Journal of Building Engineering 101827. DOI: 10.1016/j.jobe.2020.101827.

43. Fitzsimmons J, Lu R, Hong Y, et al. (2022) Construction schedule risk analysis – a hybrid machine learning approach. Journal of Information Technology in Construction 70–93. DOI: 10.36680/j.itcon.2022.004.

44. Faghihi V, Nejat A, Reinschmidt KF, et al. (2015) Automation in construction scheduling: a review of the literature. DOI: 10.1007/s00170-015-7339-0.

45. Abioye S, Oyedele LO, Akanbi L, et al. (2021) Artificial intelligence in the construction industry: A review of present status, opportunities and future challenges. Journal of Building Engineering 103299. DOI: 10.1016/j.jobe.2021.103299.

46. Van TN and Quoc TN (2021) Research Trends on Machine Learning in Construction Management: A Scientometric Analysis. Journal of applied science and technology trends 96–104(03). DOI: 10.38094/jastt203105.

47. Laing O'Rourke Centre for Construction Engineering and Technology (2019) Machine Learning for Construction Scheduling. Available at: https://www.construction.cam.ac.uk/news/machine-learning-construction-scheduling (accessed 26 November 2023).

48. Salleh R (2009) Critical Success Factor of Project Management For Brunei Construction Projects: Improving Project Performance. School of Urban Development. Faculty of Built Environment and Engineering. Queensland University of Technology. Available at: https://eprints.qut.edu.au/38883/1/Rohaniyati_Salleh_Thesis.pdf (accessed 26 November 2023).

49. Kumar K (2022) Critical Review of Common Inadequacies in Project Scheduling. IJERT. DOI: 10.17577/IJERTV11IS040015.

50. Rampini L and Cecconi FR (2022) Artificial intelligence in construction asset management: a review of present status, challenges and future opportunities. Journal of Information Technology in Construction 884–913. DOI: 10.36680/j.itcon.2022.043.

51. Wolejszo D (2020) Project planning and scheduling Literature Review. Available at: https://www.linkedin.com/pulse/project-planning-scheduling-literature-review-dariusz-wolejszo/ (accessed 12 November 2023).

52. Sanni-Anibire MO, Zin RM and Olatunji SO (2021) Machine learning - based framework for construction delay mitigation. Journal of Information Technology in Construction 303–318. Conseil International du Bâtiment. DOI: 10.36680/j.itcon.2021.017.

53. Bhatia APS, Han S and Moselhi O (2022) A simulation-based statistical method for planning modular construction manufacturing. Journal of Information Technology in Construction 130–144. Conseil International du Bâtiment. DOI: 10.36680/j.itcon.2022.007.

54. Amer F, Koh HY and Golparvar-Fard M (2021) Automated Methods and Systems for Construction Planning and Scheduling: Critical Review of Three Decades of Research. Journal of Construction Engineering and Management 147(7). American Society of Civil Engineers (ASCE). DOI: 10.1061/(asce)co.1943-7862.0002093.

55. Stewart L (2021) Artificial Intelligence is Revolutionizing This Contractor's Construction Scheduling and Risk Management. Available at: https://www.forconstructionpros.com/profit-matters/article/21509256/artificial-intelligence-is-revolutionizing-this-contractors-construction-scheduling-and-risk-management (accessed 5 November 2023).

56. Mikulakova E, Konig M, Tauscher E, et al. (2010) Knowledge-based schedule generation and evaluation. Advanced Engineering Informatics 24(4): 389–403. DOI: https://doi.org/10.1016/j.aei.2010.06.010.

57. Amer F and Golparvar-Fard M (2021) Modeling dynamic construction work template from existing scheduling records via sequential machine learning. Advanced Engineering Informatics 47. Elsevier BV: 101198. DOI: 10.1016/j.aei.2020.101198.

58. Alice Technologies (2018) Search Algorithms and AI in Construction: A Modern Solution. Available at: https://blog.alicetechnologies.com/ai-and-search-algorithms-a-modern-solution-to-construction-scheduling (accessed 30 October 2023).

59. Heaton A (2022) AI and Machine Learning will Revolutionise Project Budgeting and Scheduling – Architecture . Construction . Engineering . Property. Available at: https://sourceable.net/ai-and-machine-learning-will-revolutionise-project-budgeting-and-scheduling/ (accessed 30 October 2023).

60. InEight (2023) Construction Scheduling Software | InEight Schedule | InEight. Available at: https://ineight.com/products/ineight-schedule/.

61. Azevedo MA (2019) Peak Funding May Have Hit Construction Tech Startups. Available at: https://news.crunchbase.com/venture/peak-funding-may-have-hit-construction-tech-startups/ (accessed 29 October 2023).

62. For Construction Pros (2019) Artificial Intelligence Finds a Home in Construction. Available at: https://www.forconstructionpros.com/profit-

matters/news/21079345/artificial-intelligence-finds-a-home-in-construction (accessed 29 October 2023).

63. Rathmann C (2022) This Construction AI Project by a Canadian Contractor Has Implications for all Contractors. Available at: https://www.forconstructionpros.com/construction-technology/article/22056304/artificial-intelligence-for-construction-estimating (accessed 28 October 2023).

64. Tariq J and Safdar Gardezi SS (2023) Study the delays and conflicts for construction projects and their mutual relationship: A review. Ain Shams Engineering Journal 14(1). DOI: https://doi.org/10.1016/j.asej.2022.101815.

65. Mbala M, Aigbavboa C and Aliu J (2018) Causes of Delay in Various Construction Projects: A Literature Review. In: In Book: Advances in Human Factors, Sustainable Urban Planning and Infrastructure, pp. 489–495. DOI: 10.1007/978-3-319-94199-8_47.

66. Karmakar A and Delhi VSK (2021) Construction 4.0: what we know and where we are headed? Journal of Information Technology in Construction 526–545. Conseil International du Bâtiment. DOI: 10.36680/j.itcon.2021.028.

67. Ellis G (2023) The Rise of AI in Construction. Available at: https://constructionblog.autodesk.com/ai-construction/ (accessed 23 October 2023).

68. Buildots (2023) Buried in Spreadsheets? AI Has the Answer. Available at: https://buildots.com/blog/ai-in-construction/ (accessed 23 October 2023).

69. McKinsey & Company (2020) Rise of the platform era: The next chapter in construction technology. Available at: https://www.mckinsey.com/industries/private-equity-and-principal-investors/our-insights/rise-of-the-platform-era-the-next-chapter-in-construction-technology#/.

70. Barbosa F, Woetzel J, Mischke J, et al. (2017) Reinventing construction through a productivity revolution. McKinsey & Company. 27 February. McKinsey Global Institute. Available at: https://www.mckinsey.com/capabilities/operations/our-insights/reinventing-construction-through-a-productivity-revolution (accessed 23 October 2023).

71. Srivastava S (2023) AI in Construction: Redefining the Industry with Smart Solutions. Available at: https://appinventiv.com/blog/ai-in-construction/ (accessed 23 October 2023).

72. Pedamkar P (2023) Machine Learning vs Statistics. Available at: https://www.educba.com/machine-learning-vs-statistics/.

73. Bzdok D, Altman N and Krzywinski M (2018) Statistics versus machine learning. Nature Portfolio. DOI: 10.1038/nmeth.4642.

74. Stewart M (2019) The Actual Difference Between Statistics and Machine Learning. Available at: https://towardsdatascience.com/the-actual-difference-between-statistics-and-machine-learning-64b49f07ea3 (accessed 20 October 2023).

75. Peng J, Jury EC, Dönnes P, et al. (2021) Machine Learning Techniques for Personalised Medicine Approaches in Immune-Mediated Chronic Inflammatory Diseases: Applications and Challenges. Frontiers in Pharmacology 12. Frontiers Media. DOI: 10.3389/fphar.2021.720694.

76. Salian I (2018) SuperVize Me Difference Between Supervised, Unsupervised, & Reinforcement Learning | NVIDIA Blog. Available at: https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/ (accessed 19 October 2023).

77. Atul A (2023) AI vs Machine Learning vs Deep Learning. Available at: https://www.edureka.co/blog/ai-vs-machine-learning-vs-deep-learning/ (accessed 19 October 2023).

78. UC berkely (2020) What Is Machine Learning (ML)? - I School Online. Available at: https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/ (accessed 19 October 2023).

79. What is Machine Learning? | IBM (n.d.). Available at: https://www.ibm.com/topics/machine-learning.

80. Copeland BJ (2023) Artificial intelligence (AI) | Definition, Examples, Types, Applications, Companies, & Facts. Available at: https://www.britannica.com/technology/artificial-intelligence.

81. McCarthy CSDJ (2007) WHATISARTIFICIALINTELLIGENCE? Available at: https://www-formal.stanford.edu/jmc/whatisai.pdf (accessed 18 October 2023).

82. What is Artificial Intelligence (AI)? | IBM (n.d.). Available at: https://www.ibm.com/topics/artificial-intelligence.

83. Dockery CD (2022) 6 Ways to Imagine AI Transforming the Construction Industry. Available at: https://www.constructconnect.com/blog/6-ways-to-imagine-a.i.-transforming-the-construction-industry (accessed 9 October 2023).

84. For Construction Pros (2018) How Artificial Intelligence Is Changing Construction. Available at: https://www.forconstructionpros.com/construction-technology/article/21016665/how-artificial-intelligence-is-changing-construction (accessed 9 October 2023).

85. foresight.works (2023) Why Traditional Scheduling Software Isn't Giving the Results You Need. Available at: https://www.foresight.works/blog/why-traditional-scheduling-software-isnt-giving-the-results-you-need (accessed 8 October 2023).

86. Tariq J and Gardezi SSS (2023) Study the delays and conflicts for construction projects and their mutual relationship: A review. Elsevier BV. DOI: 10.1016/j.asej.2022.101815.

87. Merriam-Webster Dictionary (2023) Definition of planning. Available at: https://www.merriam-webster.com/dictionary/planning#:~:text=%3A%20the%20act%20or%20process%20of,a%20social%20or%20economic%20unitt.

88. HISTORY (2009) Benjamin Franklin - Biography, Inventions & Facts. Available at: https://www.history.com/topics/american-revolution/benjamin-franklin.

89. McKinsey Productivity Sciences Center (2015) The construction productivity imperative. June. McKinsey Productivity Sciences Center. Available at: https://www.mckinsey.com/~/media/McKinsey/Industries/Capital%20Projects%20and%20Infrastructure/Our%20Insights/The%20construction%20productivity%20imperative/The%20construction%20productivity%20imperative.pdf (accessed 8 October 2023).

Appendices

1. Training and Inference Models
    a. Task List Generation Model
    b. Activity Relations Models
    c. Optimised Durations Model
2. Datasets
    a. Task List Generation Model
    b. Activity Relations Models
    c. Optimised Durations Model
3. Model and Console Log Outputs
    a. Task List Generation Model
    b. Activity Relations Models
    c. Optimised Durations Model