Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

Aneeq Ahsan

AI in projects- an investigation of use of project-based data for prediction of changes

Master's thesis in Project Management
Supervisor: Nils Olsson
June 2022

NTNU
Norwegian University of Science and Technology

Aneeq Ahsan

AI in projects- an investigation of use of project-based data for prediction of changes

Master's thesis in Project Management
Supervisor: Nils Olsson
June 2022

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Preface

This thesis is written as completion of two year master program in Project Management with specilization in production and quality engineering and it counts for 30 credit hours. It is written at department of Mechanical and industrial engineering with faculty of engineeing.

The topic of this thesis was selected due an oportunity provided by my supervisor Nils Olsson after specialization report in fall of 2021. The specilization report was focused on machine learning applications in transport sector with in depth understanding of machine learning principles. That report provided the baseline for learning and imrovement during the research in this thesis. By changing the topic to project data from transport data the data types remain same however, the challenges were more because of less amount of research in project management applications of AI and specifically in topic of interest.

Project management is a complex domain of knowledge and thus it was deemed more challenging and rewarding. Projects are normally thought of as old profession with less innovative applications than other disciplines such as computer science or engineering but nevertheless provides large spectrum of innovations. The transition to artificial intelligence applications on project management from transport data applications was a rewarding and innovative experience.

I am thankful to my supervisor for providing me the opportunity to work on intersection of project management and artificial intelligence and for continual guidance and support during the research process.

Acknowledgement

This thesis is a result of continual collaboration with several key actors including my supervisor, collaborating contarctor company and project management software company representitives.

My supervisor, Nils Olsson provided me the opportunity to understand the impact of combining project management and machine learning. He was continiously supportive and provided essential feedback and knowledge required to undersatnd the concepts in both domains. The collaboration between contarctor and software company is also result of his efforts.

I am thankful to the team at contarctor company who provided the data for this thesis. They made this research possible with providing essential knowledge in undersatnding the project operating procedure and data logging at the company. Their support in understanding project methodology used at the company and essential knowledge about change order procedures were vital in this thesis.

I am also very thankful to the project management software representitves who helped me undersatnd the data structure in the software and continually helped me in every phase of the research. This research is a product of collaboration between all of these key actors and without their support, opportunity and feedback this thesis would not have been possible.

I am also thankful to NTNU for providing me a life-time experience here. It has been a wonderful environment with perfect balance of fun and knowledge at the campus. I will be forever indebted to the people affiliated and this institution for my personal growth and opportunities provided.

Abstract

This thesis analyzes a single project data from energy construction contractor stored in a project management software for machine learning (ML) applicability. By provision of extensive data availibility of project plan the opportunity to explore ML and AI methods is assesed. Data quality and ease of use in ML applications is studied by geenration of three different datasets for this single project. Data exploration, cleaning and pre-processing needs for the data are also identified with discussion of limitations and data inadequacies in project management software. The data generation steps used are validated by prediction of a previously researched quantity in ML applications of project data. The validation step also provides insight into the data quality and data reshaping impact on the desired prediction. The study main objective is to find a ML and data format that is able to predict number of variation orders in a project and thus reduce uncertainity and scope creep of projects. The study also identifies limitations of the methods employed and provides interpretable insights fro deep learning models that provided more than 80% accuracy in prediction of variation orders. The most impact from the project data on number of variation orders are identified as total scope, current scope, days remaining in current late finish, days remaining till revised plan early start, remaining hours and earned hours. The model is interpreted by dividing project into four parts and interpreting features, timestamps of importance, and activities that influence the most on issuance of variation orders. These four parts division of data is done to produce actionable insights from model and see any bahaviour change of model as project progresses. A key insight from analyzing impact of features is that during the end of project the model emphasises expended quanity and performance factor which the model seems to be neglecting in early phases of a project. From time dependencies impact analysis it is seen that during the start of a project only last week is important for variation order issuance however, during the middle phase of a project previous three weeks are given importance almost equally and this behaviour changes again at the end of project where only last week seems to be important for the model. The activities impact is also dynamic and changes as project progresses, in the start of a project the activities that are near completion are given more importance while shifting the importance from most completed to least completed activities as project progresses toward the end. These insights however are not linear and thus model developed are not directly interpretable linearly. These insights however have high importance for reliability check of models and building of trust in the model. By sucessful implementation of these ML models and development of other models on project data the project change management systems can be improved substantially with equally important implications for planning.

Sammendrag

Denne oppgaven analyserer prosjektdata fra leverandører i energibransjen. Ved å analysere prosjektplaner vurderes muligheten til å utforske maskinlæring (ML) og artifisiell intelligens (AI) metoder. Datakvalitet og brukervennlighet i ML-applikasjoner studeres ved generering av tre forskjellige datasett for dette enkeltprosjektet. Datautforskning, rensing og forbehandlingsbehov for dataene identifiseres også inkludert diskusjon om begrensninger og datamangel i prosjektstyringsprogramvare. Datagenereringstrinnene som brukes, valideres ved prediksjon av en tidligere undersøkt mengde i ML-applikasjoner av prosjektdata. Valideringstrinnet gir også innsikt i datakvaliteten og dataomformingens innvirkning på ønsket prediksjon. Studiets hovedmål er å finne et ML- og dataformat som er i stand til forutsi antall endringsordre i et prosjekt og dermed redusere usikkerhet og omfangskryp i prosjekter. Studien identifiserer også begrensninger ved metodene som brukes og gir tolkbar innsikt fra dyplæringsmodeller som ga mer enn 80 % nøyaktighet i prediksjon av variasjonsordre. Den største innvirkningen fra prosjektdataene på antall endringsordre er identifisert som totalt omfang, nåværende omfang, dager som gjenstår i nåværende sen finish, dager som gjenstår til revidert plan tidlig start, gjenværende timer og opptjente timer. Modellen tolkes ved å dele prosjektet inn i fire deler og tolke funksjoner, tidsstempler av betydning og aktiviteter som påvirker utstedelsen av variasjonsordrer mest. Disse fire deler av datainndelingen gjøres for å produsere nyttig innsikt fra modellen og se enhver atferdsendring av modellen etter hvert som prosjektet skrider frem. En nøkkelinnsikt fra analysen er at modellen under slutten av prosjektet legger vekt på brukt kvantitet og ytelsesfaktor som modellen ser ut til å neglisjere i tidlige faser av et prosjekt. Fra tidsavhengighetsanalyse ser man at under oppstarten av et prosjekter bare forrige uke er viktig for utstedelse av variasjonsordre, men i midtfasen av et prosjekt er de tre foregående ukene gitt nesten like stor betydning og denne atferden endres igjen ved slutten av prosjekt hvor kun forrige uke ser ut til være viktig for modellen. Aktivitetspåvirkningen er også dynamisk og endres etter hvert som prosjektet skrider frem, i starten av et prosjekt gis aktivitetene som nærmer seg ferdigstillelse større betydning, mens viktigheten flyttes fra mest fullførte til minst fullførte aktiviteter etter hvert som prosjektet skrider frem mot slutten. Denne innsikten er imidlertid ikke lineær, og modellutviklede er derfor ikke direkte tolkbare lineært. Denne innsikten har imidlertid stor betydning for pålitelighetssjekk av modeller og oppbygging av tillit til modellen. Ved vellykket implementering av disse ML-modellene og utvikling av andre modeller på prosjektdata kan systemer for styring av endirnger i prosjekt forbedres vesentlig, noe som har viktige implikasjoner for planlegging.

Table of Contents

List of figures

List of tables

List of abbreviations

ABR Ada Boost regressor

AC Actual cost

AI Artificial Intelligence

ANN Artificial neural network

AOA Activity on arc

AON Activity on node

BEI Baseline execution index

CNN Convolutional neural network

CPI Cost performance index

CPM Critical path method

CV Cost variance

EAC Earned at completion

ED Earned duration

EDA Exploratory data analysis

EDM Earned duration management

ES earned schedule

ETL Extract transform load

EV Earned value

EVM Earned value management

GBDT Gradient boosted decision tree

GBM Gradient boosting machine

GBR Gradient boosting regressor

KPI Key performance index

LGBM Light Gradient boosting machine

LSTM Long short-term memory

MAPE Mean absolute percentage error

MIL Multi instance learning

ML Machine learning

MSE Mean squared error

NLP Natural language processing

NN Neural network

PCA Principal component analysis

PERT Program evaluation and review technique

PM Project management

PMBOK Project management book of knowledge

PMI Project management institute

PV Planned value

RNN Recurrent neural network

RQ Research question

SPI Schedule performance index

SQL Structured query language

ST Spatio-temporal

SV Schedule variance

TPE Tree structured Parzen estimator

VIF Variance inflation factor

VO Variation order

WBS Work breakdown structure

1 Introduction

This thesis investigates the applicability of artificial intelligence to project management plan data. Project management has a lot of data and potential use of this data for efficient project management planning and control is accessed in this thesis.

The first and foremost task is to extract the data and format it to an acceptable machine learning input format. There would be several steps involved in this process so that the steps are formalized and enables digital transformation for future use of project data. Company project data is stored in a software database and the contractor is interested in applicability of machine learning to improve change management of the project.

Machine learning has been successfully applied to several domains and continues to make its way to the rest. In project management several studies have focused on the prediction of earned value (Kuhl & Graciano, 2014), project control (García et al., 2017; Peña et al., 2019) and success of a project (Cheng et al., 2010) based on different attributes of a project. In the offshore sector the research has been done on large projects (Tanaka, 2014) . In the energy sector, contractors play an important role especially in EPC contracts (Wang et al., 2016). The industry is becoming more digital and several digital transformation projects such as Internet of things (IoT) (Kolloch & Dellermann, 2018) , data analytics (Kolloch & Dellermann, 2018), semantic technologies (Moser et al., 2011) and machine learning (Hanga & Kovalchuk, 2019) . Cost overruns and longer project durations are still considered to be large in these projects (Rui et al., 2017). The challenges in project management have been defined in several studies (Mullaly & Thomas, 2009). Change management and change control have been a major factor in project challenges (Shafaat et al., 2016).

Project management has not yet fully utilized the potential of technology and digitalization. The amount of data gathered by any industry or process has increased exponentially over the years (Sawadogo & Darmont, 2021). Similar to every other industry the amount of data that project managers have access to has increased exponentially. However, the use of this data with the current technologies has not yet been explored completely. This thesis will try to bridge that gap between utilization of data and project management needs.

Project managers are essential part of a project (Mir & Pinnington, 2014). A project manager is responsible for project execution and control of activities to meet the goals set by the project owners(Sołtysik et al., 2020). Soft skills are often discussed for project manager roles because they must reach out to several stakeholders and continually update them on progress and decisions. The role of a project manager is becoming more diverse with inclusion of soft skills, control and change management rather than just engineering technical roles (Alotaibi & Mafimisebi, 2016; Edum-Fotwe & McCaffer, 2000). The success of a project manager still depends on best practice methodologies such as timely execution and with minimal overruns. The context also changes with national environments depending on the maturity of project methodologies of governance (de Carvalho et al., 2015). The digitalization in project management has been more focused on conversion to digital methods of storage, communication, and development of tools to track project performance. It is only recent that various programs have been developed

to help project managers and the team in execution and control of projects (Bean, 2018; Schreck et al., 2018).

The complexity of projects is increasing because of global supply chains and procurement partnerships, and the challenge to deliver project on time and within the planned budget Is becoming more complex (Nady et al., 2022; Tanaka, 2014). The oil and gas sector has historically been in high demand however, due to recent fluctuations in oil prices the industries are moving toward renewable resources. The renewable industry has seen exponential growth in the last century (Renewables, 2020). The need for successful project management and change management is evident by the growth of number of projects and the required execution efficiency. Project management is becoming global (Keshavarzian & Silvius, 2022) and multiple bilateral agreements and incumbents exist in the ecosystem of a project. The industry strategy must be aligned to cut costs and optimize its efficiency (Tanaka, 2014). Implementing new technologies could be a solution to overcome these challenges. Most common applications have been in adoption of digital twins, multi-agent systems and smart programs to monitor and control the assets (Hajizadeh, 2019; Hanga & Kovalchuk, 2019).

Machine learning (ML) is better at identification of hidden patterns and trends in the data than humans (Pavlus, 2016). ML provides the opportunity to learn automatically and improve the results as it gains experience without the need to be explicitly told. The applications of ML are not however required in simple tasks such as sorting of few numbers and is considered only feasible when data is complex and deployment of algorithmic programs are not feasible (Choudhary & Gianey, 2017). A complex task such as filtering emails to identify spam emails would make it feasible to invest in ML development. In this a no specific rule exists for this filtering and thus we can use data and ML algorithms to make these types of decisions (Domingos, 2012).

Machine learning algorithms can perform better with increasing data dimensions however, more data is not an indicator of an algorithm performance (Adadi, 2021). These algorithms requires less or no missing data to function properly. The data in the oil and gas industry and project management is increased and comes from multiple sources (Hajizadeh, 2019). However, machine learning can be hard to customize for domain specific and specialized tasks and sometimes does not justify the investment in the infrastructure required for machine learning systems (Guyon et al., 2019; Schelter et al., 2018).

1.1 Problem description

The purpose of this thesis is to use project plan data to investigate how and to what extent ML methods can be applied to investigate variation orders (VOs). Problem is framed as prediction of count of VOs for next week and to predict if next week will have any VO. These types of problems are called regression and classification in machine learning theory (El Naqa & Murphy, 2015). To make the task organized and valid four research questions (RQs) have been formulated. These research questions have been formulated to make the reader understand the implications of machine learning to a single project data.

RQ1: To what extent project management software provide sufficient data quality of project plans for compatibility with ML methods?

RQ2: How valid is the data generation process employed for ML predictions?

RQ3: Can ML models predict number of variation orders (VO) for next week or classify a week as potential candidate for VO next week?

RQ4: What are the impacts of successful machine learning implementation for change management and what is the predictive power of a single project data?

The order of these research questions is formulated in such a way as to give reader easier understanding of this paper. First RQ concerns the applicability of ML methods on project data and to understand the data quality provided by PM software for a project plan data. The extracted data from PM software is analysed for discrepancies and data quality. A dataset generation process is then employed to generate datasets for experimentation which will be evaluated for validity by employing a different target for ML, this switch of targets improves confidence in the developed method if results are found significant. The main part of this thesis is to identify if VO is a predictable target using ML methods which is addressed in third research question and the effects of successful ML implementation for change management is identified in the final research question.

2 Theory

Theory section consists of sections explaining project management, project planning, machine learning, data types and models for different problem types. The project management section will explain how traditional project management is structured and what challenges exist in the current implementation of a framework employed by a company. Planning of projects section will explain how a plan is developed, which is an important part of project management. It will also show how and what tools project managers can utilize for better planning decisions and make estimations of cost and time. Machine learning section explains what it is, how it works and what are the requirements and limitations of different algorithms. Data types are important for machine learning systems and therefore they are explained in a different section. Several models for each type of task in this thesis are also examined and studied.

2.1 Project Management

Project management is primitive, but the exact timing of when this became a knowledge area is unknown (Walker, 2015). The expectations of clients in a project largely depends on what and how resources are utilized, given the current complex ecosystem of projects and clients these expectations are further enhanced and thus require more efficiency in terms of cost and time (Walker, 2015). In the early phases of a project, some of the important decisions are also soft decisions about modes of communication, visualization, availability of resources for management of projects and tools to use for project management (Mullaly & Thomas, 2009; Russell et al., 1997). Good project management skills are needed for every business, thus making it a well-established knowledge area including several best practices for different industries (Schoper et al., 2018). Project management book of knowledge (PMBoK) is published by project management institute (PMI) to help project managers understand different knowledge areas and explain classical procedure in project management (PM) (Project Management, 2017). Due to increased complexity the expectations from project managers are also changing (Edum-Fotwe & McCaffer, 2000). The knowledge areas provided by PMBoK is therefore further expanded to include other aspects (Alotaibi & Mafimisebi, 2016). To meet these new expectations non-engineering knowledge requirements are also included to achieve this desired level of performance. The incorporations range from stakeholder management, sustainable goals, negotiations, working with multi-functional teams (Alotaibi & Mafimisebi, 2016; Keshavarzian & Silvius, 2022). These additions have been made to stay within the defined cost and timed estimates for a project (Russell et al., 1997). (Mir & Pinnington, 2014) found a strong relationship between project success and project management, however, (Mullaly & Thomas, 2009) studied 65 projects and concluded that in spite of importance of project managers, organizational cultures and implementation that fits the need is also important. Some of the researchers have suggested that current knowledge of project managers are not adequate for successful management of complex relations that exist now (Ramazani & Jergeas, 2015).

2.1.1 Project management challenges

Several challenges exist in project management for different industries, for the construction in Oil & gas industry the biggest challenge is to transition towards a

renewable energy sources (Tumin et al., 2022). During this transition and digital transformation there is tendency for these technologies to move towards industrial engineering, which is an integration of systems, people, information, services, and resources to cope with increasing globalization (Blanchard et al., 1990). The projects from oil and gas industry however, generates an environmental sentiment therefore, (Badiru & Osisanya, 2016) argue that PM should be an integral part of planning, organization, and project control . (Michaelides et al., 2014) argues that the negative aspects in the industry include lack of shared responsibility, inadequate planning, unstandardized procurement and weak collaboration between subcontractors and firms. Some of these negative aspects can be improved or eliminated by using new digital collaborative technologies but the integration does not come without friction from the actors involved (Kolloch & Dellermann, 2018).

Due to increased globalization of supply chains, project managers now have to manage, collaborate, distribute information and manage across multiple stakeholders across multiple countries. Project performance also depends on national governance policies and countries with mature methodologies for PM (de Carvalho et al., 2015). As a consequence of this project complexity is increasing as assemblies and production are outsourced to low labour cost countries (Badiru & Osisanya, 2016). According to (Rui et al., 2017) oil and gas projects have not performed well in the past, more than half of the projects have experienced cost overruns of over a third of target cost.

In oil and gas industry, digital transformation has begun and as top management understands the value of new technologies, a project manager is able to implement these in projects (Angelopoulos et al., 2019). Integration of these transforming technologies can improve working conditions and environment of the project and thus create value for customer (Angelopoulos et al., 2019). However, in traditional PM approaches, project managers are hesitant to involve new technologies into the processes (Kolloch & Dellermann, 2018). One of the most emergent technologies is digital twin with the purpose of asset monitoring, maintenance, life cycle management, and project planning (Wanasinghe et al., 2020).

Other challenges in traditional project management are related to estimation of costs and schedule (Park, 2021). The uncertainty in the projects is inherent to its nature and must be managed properly (Taylor, 2008). Due to the increasing complexity of projects, traditional approaches used for estimation are insufficient to capture non-linear relations in the project (San Cristóbal et al., 2019). Due to globalization, businesses all across the world are spending more money on improvements of their predictions and estimations by decreasing the granularity of estimations (Banker, 2019). These estimations and predictions are time dependent and due to complex timelines it is hard to predict (Drezet & Billaut, 2008). Research conducted by UC Berkley (Tetlock, 2009) showed that expert predictions of almost 300 experts, showed only slightly better results than a random guess. However, experts that had a large portfolio of expertise performed better than a focused domain expert. As data increases, ML algorithms therefore might be better at prediction than humans. According to (Weinberger, 2019) Ai and ML could decrease the gap between our understanding and surpass our performance in prediction tasks. Humans are not able to calculate lengthy sequences and are not great at identification of data patterns (Daniel, 2017). Thus, computers and ML algorithms that can extract these patterns might be better at identification of these patterns and predictions.

Project estimations are usually done by three traditional approaches (Kerzner, 2022). In analog approach, estimations are done by comparison and approximations. The accuracy of estimation is therefore dependent on the knowledge of the person or panel making the estimations. In another method called order of magnitude, in which important aspects of a project are used to estimate a project. It is a simple method, therefore, it is used in the early phases of a project. The last method is called definitive method, much like order of magnitude method this method takes input of all aspects of a project and thus is more detailed and accurate in estimations. Because of the dynamic nature of projects a usual 10-15% of buffer is added (Pinto, 2010) . A method called triple point estimation is often used to quantify the stochastic nature of projects in estimations (Hong, 1998).

2.1.2 Project management methodologies

Project methodologies are highly important for project success and the choice of methodology is largely dependent on project size, type and company (Rasnacis & Berzisa, 2017). Management of projects and its success is highly dependent on choice of plan and procedure of these project methodologies. These methodologies standardize predictability, execution, control and data collection (Rolstadås et al., 2016). The choice of methodologies to select are many and depends on the detail needed. Overview models communicate some parts or complete discipline of project management, few popular methodologies are PMI, PRINCE2 and IPMA (Rolstadås et al., 2016). A methodology can also be based on segmentation of phases of a project. A simple development cycle might start from project definition and then drilled down to technical, functional, management and financial baselines. These individual baseline definitions are also customized to fit the type of project (Taylor, 2008). Process models are more focused on specific process such as estimation, uncertainty, risk or work breakdown structures (WBS) (Rolstadås et al., 2016). In these methodologies key performance indicators are often put forward and they provide a link between different phases of a project. These steps are iterative in nature and with the monitoring and control stage they are often updated and developed as required.

2.2 Planning of projects

2.2.1 Baselines

Baselines are used in robust managing of project schedule (Schatteman et al., 2008).According to (Tereso et al., 2019) baselines are one of the most widely used practice in private organizations. They also argued that it is one of the important factors in project success. A baseline compares a state with which future developments are compared with. It is a snapshot of what the future should look like, it is set at beginning and at specified intervals in a project lifecycle. A project manager then utilizes this baseline as a reference to compare actual progress with baseline planned progress. Therefore, It is essential to have a good initial plan or run baselines at specified intervals when more information is available (Besner & Hobbs, 2013). Through monitoring of actual progress and baseline planned progress, actions required are often easier to spot. Therefore, a good baseline is important. One method is to use project maturity model describe by (Kerzner, 2002). A baseline is also required to be accessible and often due to multiple baselines they must be distinguishable. Multiple baselines are common due to inherent nature of a project and uncertainty. As project progress baselines must be updated to keep track of changes, information, and activities of a project. Number of baselines depends on project complexity, size, and length of project (Thomas & Završki,

1999). These baselines are often generated and maintained in a project management software. Project management team develops a plan based on previous experiences, knowledge and expected durations of tasks, expected completion date and when an activity must be completed if it is a dominant criteria (Seely & Duong, 2001). The required resources and order of tasks are adjusted and optimized; this leads to a baseline that a project manager can do comparison with during the execution of a project to check variances. A PM toll often can generate a baseline. Although a computer can calculate the baselines, a human is the one who originally specifies the properties and requirements of a task and therefore uncertainty is inherent to the plan (Tereso et al., 2019).

Multiple baselines generation is valuable since it stores different snapshots of project and makes it easier to spot variances. This can lead to knowledge of required actions. Also, these baselines are good knowledge resource for project management practices, subcontractor performances or execution methods. A baseline can have different versions however, these versions should be kept as small as possible and ideally one. A version might be necessary in case of scope changes, cost changes, or force majeure events. It can also result from poor planning, and thus creates a ripple effect (Thomas & Završki, 1999).

A baseline can be of many forms depending on the needs for the project and management. One baseline might also have different aspects to signify certain aspects of a project (Taylor, 2008). These baselines might be cost, scope, schedule, or quality baselines. Using these baselines are main drivers of effectiveness. These comparisons of actual and planned progress, scope, or budget gives valuable knowledge and insights into project performance. Baselines are said to be learning tools for project managers as well (Seely & Duong, 2001).

2.2.2 Work breakdown structures

Work breakdown structure (WBS) is integral part of project execution and control. It connects the cost, schedule information, work estimates and actual costs and schedule (Colenso, 2000). Any complex projects , such as in energy industry is made manageable by breaking it down into individual components in a hierarchy (Devi & Reddy, 2012). Development of WBS improves project organization, resource allocation, control points, estimations and explain scope of the project. A work package is situated at the bottom of the hierarchy in a WBS and is the smallest manageable part of work that can be done and controlled. These work packages are deliverables independently, has shortest lifecycles, clarifies the relationship to other entities in WBS (Devi & Reddy, 2012). A WBS also incorporates milestones, they are defined as having no duration, signifies major progress points (Colenso, 2000). The lowest level is used for reporting schedule and costs (Rolstadås et al., 2016). A WBS provides information to plan, schedule, and control projects including baseline settings (Rolstadås et al., 2016). The deliverables of the whole project can be recalculated by rolling up from the lowest level.

In a WBS breakdown can be based on different characteristics such as functional components, physical components, geographical, organizational, departmental, sub-contractor etc (Webster, 1994). A WBS, however, requires an in-depth knowledge of project and activities before the start. It therefore, requires domain experts and large planning duration for the project. Another downside might be tunnel vision, which can limit the resource focus on one work package rather than the whole deliverable (Webster, 1994).The lowest element of a WBS, called a work package, should be defined in such a

way that it is measurable in results, start, finish and estimations such as cost and time. It should also be independent of other elements at the lowest level (Rolstadås et al., 2016). Activities are then assigned to the work package and baseline is usually set using a work package level aggregation.

2.2.3 Scheduling

Project scheduling is an important part of project. The activities in a WBS are dependent on each other in at least two ways, they compete for resources and order of activities is important (Hartmann & Briskorn, 2022). In scheduling, network analysis is done considering constraints, connections, resource limitations and calendars, the output results in an estimate of the project schedule, which can be set as first baseline. A good scheduling requires both analytical and artistic traits (Taylor, 2008). A good schedule is said to have a big impact on whole project (Demeulemeester & Herroelen, 2006).

Some of the challenges in scheduling are constraints, connections, time lags, release dates and deadlines, temporal constraints, logical dependencies, resource constraints (Hartmann & Briskorn, 2022). A time lag is a type of constraint which limits the activity start or finish before a predecessor is finished or started. Release dates and deadlines mark the earliest start or latest completion time depending on the feasibility of the activity (J. Cheng et al., 2015). Other types of constraints might be to limit start of activities at same time, finish activities at same time or limitations in activity overlaps (Hartmann, 2013). Flow network models are used as an aid for project managers to visualize activity links and their durations (Tavares, 2002). Two types of networks are activity on nodes (AON) and activity on arc (AOA), in the former activities are located on nodes and arcs signify the dependence between activities. AOA network has activities on arcs while start and finish are signified by head and tail (Ahuja et al., 1994). Another visual tool to help scheduling is Gantt diagrams, in which activities are located and sorted on vertical axis while horizontal axis signifies time (Dupláková et al., 2017). This tool makes it easier to visualize the project timeline, however, no or little information about resources are present or any dependencies between activities (Rolstadås et al., 2016).

Scheduling requires activities, events, and milestones. Activities are defined in a WBS at the lowest level of a work package (Rolstadås et al., 2016). Events are when an activity starts or finishes. The linkages between activities can be start-start, start-finish, finish-start and finish-finish (Ahuja et al., 1994). Number of activities in the network are directly related to the complexity of the project. Apart from these constraints, there can also be time duration constraints between activities. If the duration of all the activities are deterministic, critical path method (CPM) can be applied to find the critical path in a network (Rolstadås et al., 2016). However, in case of uncertainty in duration of activities program evaluation and review technique (PERT) can be applied, which requires triple estimates of duration namely, least likely, most likely and average durations and uses Beta distribution to model project duration. PERT calculates mean and variance for each activity and later uses these to calculate total duration of the project (Rolstadås et al., 2016).

CPM makes assumption of durations to be deterministic and start, finish time for whole project is predetermined (Rolstadås et al., 2016). The inputs for this method are activity durations and linkages between activities. If a project completion day is also given, then it is considered as input in this method. CPM method tells us which activities are critical for the implied schedule. The calculation steps consists of two steps, first events are calculated for each activity such as late start, early start, late finish, early finish and in

the second step activity durations are calculated. These are called forward and backward passes through a network. In a forward pass, early event times are calculated and in backward pass late event times are calculated (Rolstadås et al., 2016). The sequence for calculation is as follows: early start, early finish, late finish, late start in the same order for each activity. These calculations also give insight into two further durations called float and slack. Float is defined as the freedom that an activity has for moving around in a schedule such that the whole plan is not influenced by the movement. Activities having zero float are called critical activities. Free float is the term used to describe how much an activity can be moved in time such that no succeeding activity incurs any impact. A critical path is then a path of activities in a network that cannot be delayed without having effects on whole project duration. A network can however, have different critical paths (Rolstadås et al., 2016). An activity can also have precedence restrictions, for example, an activity A cannot finish before activity B starts with a restriction of one day, therefore activity A can only finish one day after activity B starts. The precedence relationships between activities can be more than one and there can be different precedence relationships between two activities (Rolstadås et al., 2016).

PERT method for scheduling takes into account activity duration uncertainties and model these uncertainties using some distribution such as beta-distribution (Rolstadås et al., 2016). This distribution uses triple estimates of an activity duration, namely, most likely estimate, least likely estimate and average estimate of activity duration. These estimates are then modelled with beta distribution to find expected values and variance of duration for an activity. These expected values can be then used to determine the probability of finishing a project within a deadline (Rolstadås et al., 2016).

Due to scarcity of resources in real life projects, scheduling is often constrained by these resource limitations (Hartmann & Briskorn, 2022). Resource requirements are plotted from a schedule and known resource requirements per unit time we can estimate periods during which resources are scarce, which is called an overload (Rolstadås et al., 2016). To avoid overloading conditions, levelling is done which moves the activities in a schedule which have floats. If floating activities are not present, then project deadline needs to be extended or resource availability increased. Resource constraints thus imposes two possible scenarios, one in which resource availability is changed and another where schedule is changed (Rolstadås et al., 2016).

2.2.4 Performance measures

One of the most commonly used performance measure in projects are EVM system (Bower & Finegan, 2009). It used Planned value (PV), earned value (EV), actual cost (AC) as common metrics. PV are derived from work packages in a WBS, while actual cost is the cost incurred during the project. Earned value is a measure that signifies how much value is earned during the project (Rolstadås et al., 2016). During the execution of a project these values are calculated cumulatively and used as a performance measure (Rolstadås et al., 2016). These cumulative values form a S-shape curve during the project execution and are compared with baselines or planned values to keep track of the performance of a project (Chao & Chien, 2010). Some of the derivatives from these metrics are cost performance index (CPI), which is the ratio between EV and AC (Warburton & Kanabar, 2008). Another derived metric is Schedule performance index (SPI) which is the ratio between EV and PV. Both derivatives are used to track progress of a project, where value of 1 means the project is on target, while less than 1 means worse than planned performance.

Some of the other derivatives are earned at completion (EAC) and expected duration (ED) which signifies the forecasted earned value at project completion and expected duration of project at completion respectively. Another metric is Baseline execution index (BEI), it is a metric to keep track of schedule of a project from activities point of view. This index demonstrated ratio of activities started to activities planned to start at any given time (Beck & Kovacs, 2018). It also gives an insight into the execution pace of project.

Earned value management (EVM) provides project managers with metrics to calculate and control cost and schedule. It gives an insight into schedule, progress and cost and provide opportunity to make necessary amends to projects strategy (Anbari, 2003). It is one of the most widely used systems to keep projects in check (Khamooshi & Golafshani, 2014). However, to make this method successful the prerequisites are to have clear scope, precise schedule, and detailed budget. In a WBS the lowest level work packages are needed to have a budget and schedule plan and this sets a baseline for comparison (Cioffi, 2006).

Several parameters are defined by EVM to monitor and control projects such as planned value (PV), earned value (EV), actual cost (AC), cost variance (CV), schedule variance (SV). Some of the forecasting metrics and project efficiency metrics are also cost performance index (CPI) and schedule performance index (SPI) (Anbari, 2003). However, over the years EVM has been criticized for its schedule metrics (Khamooshi & Golafshani, 2014). SPI has been shown to have some flaws due to its property of having SPI equal to 1 for any activity that has been completed, which does not take into account the actual performance of an activity (Khamooshi & Golafshani, 2014). Earned schedule (ES) is therefore proposed as a solution (Beck & Kovacs, 2018). This ES still has some flaws as outlined by (Khamooshi & Golafshani, 2014) , where if a critical activity has not been started both EVM and ES SPI indices tend to give high performances. Authors of (Khamooshi & Golafshani, 2014) proposes use of earned duration (ED) to make acceptable schedule metrics for project control. Based on findings of an empirical project data, however, EVM provides reliable projects cost estimates while behind 15 to 20 percent project completion (Fleming & Koppelman, 2016). EVM provides different metrics to forecast time, cost and durations.

2.3 Project control

Project control is not considered a knowledge area by PMBoK (Project Management, 2017) , however, project control includes segments from other knowledge areas such as cost control and schedule control. Project control is an important aspect, out of 39 processes required for successful project management 21 processes relate to planning (Globerson & Zwikael, 2002). The execution according to planning is dependent on control methods (Rozenes et al., 2006). If project control is given more importance project performance increases (Avison et al., 2001). Project control systems could be one dimensional or multi-dimensional (Rozenes et al., 2006). An example of multi-dimensional control system is earned value management (EVM). One-dimensional control systems aim to control and manage a specific issue to achieve a specific aim, one such tool is project scope management defined by PMBoK. Another one-dimensional tool is to control strategic project by connecting success factors with balanced score card approach (van Veen-Dirks & Wijn, 2002). Project finance control system using shareholder value analysis introduced in (Akalu, 2001) focuses on financial aspects of a project control.

One-dimensional control systems lacks integration required to completely achieve project management objectives (Rozenes et al., 2006).

Multi-dimensional project control systems provide the necessary integration of several objectives of a project. EVM is an example of such a control system, which monitors cost and schedule dimensions of a project (Rozenes et al., 2006). Another system is called multidimensional project control system introduced by (Rozenes et al., 2004), in which a comparison between planning and execution is done based on global project control specification.

Traditional project management attempts to control different principles separately, such as earned value, productivity, resource consumption, completion and forecasting (Rolstadås et al., 2016). Earned value will be discussed in detail in the sections below. Productivity measures how resource has been consumed in contrast to the estimated consumption. It can be measured in terms of hours or costs incurred. (Rolstadås et al., 2016) recommends multi-dimensional project control to gather complete picture of the project status including volume and costs. Progress of a project can be measured with respect to its duration such as schedule progress and its volume such as work completed. Therefore, it is important to measure an activity progress on both volume and time. Changes affect the project progress (Rolstadås et al., 2016).

2.3.1 Scope change management

Projects hardly run according to the plan (Douglas III, 2000). Scope change management is part of scope management (Mochal, 2008), where baselines are set. The term scope change is usually misunderstood as scope creep (Amoatey & Anson, 2017). Changing of scope is an official decision from project manager while scope creep is unofficial and propagates through a project lifecycle. (Kerzner, 2009) however, states that some project managers state scope creep as scope changes without approval. A change in scope usually always results in adjustments to project baseline (Shirazi et al., 2017). Project managers should make controlled changes to project scope (Madhuri et al., 2018). Project theory has considerable limitations for complete knowledge of project scope (Andersen et al., 2011).

According to (Olsson, 2006), changes are more likely driven by project owners or users however, project managers or contractors are reluctant to these changes. A scope change is a result of perceived increased effectiveness of a project (Olsson, 2006), he also points out the importance of reduction of negative impacts of these changes.

Project change generally could mean any sort of change in environment or scope. (Revay, 2003) defines project change as an event that modifies scope, execution, cost or quality of work. (Lee, 2007) defines change as "*any action, incidence, or condition that makes differences to an original plan or what the original plan is reasonably based on*". A variation order however, is any change that is incorporated into the contract (Anastasopoulos et al., 2010). Changes are inevitable in projects since by the nature of a project it is a unique endeavor and due to limitation of time and resources (Moselhi et al., 1991).

Project performance is also affected by changes and change management is considered an important component of project management (Zou & Lee, 2008). Changes also cause disputes between contractors and owners (R. M. Jones, 2001). Project change however, is highly difficult to predict due to the inherent nature of projects being different (Hsieh et al., 2004). The impact of VOs on project performance has been studied previously

(Ibbs, 1997; Ibbs et al., 2003; Moselhi et al., 2005) however, little research has been done on quantifying the variation order quantities (Chen, 2015).

Project change can be of two types, work change and modifications (Rolstadås et al., 2016). Work change increases the volume of work. Addition changes only affects the current activities in a schedule while modification changes the activities that are planned (Rolstadås et al., 2016). According to (Hanna et al., 2004) changes can be caused by many factors such as design changes, engineering errors or external conditions on the construction site. (Hsieh et al., 2004) found that most of the variation orders are caused by problems in design and planning. They also suggested that type of project can have an impact on causes of change orders. Variation orders can have two types of impacts direct and cumulative (Chen, 2015). Direct impacts are quantifiable and can be measured such as increase in required quantity of work. Cumulative effects are however difficult to measure (Chen, 2015).

Management of changes has been extensively studied to analyze the impact of changes and change orders (Zhao et al., 2008). Dynamic planning and control method was developed by (Lee et al., 2006) to simulate impact of changes on a project. A web based change management system was introduced that supports different functionalities for team members in a change order workflow (Charoenngam et al., 2003). However, these approaches are focused on impacts of change orders. (Kartam, 1996) suggested that identification of changes early can have benefits and conflicts can be averted.

Change orders have direct impacts on performance and success of a project (Oyewobi et al., 2016). According to a study conducted by (Ghenbasha, 2018) for construction projects, a major reason for change orders are changes in design. A change order can also impact a projects schedule and cause delay which was reported by a study conducted by (Assaf & Al-Hejji, 2006). One of the major impact of a change order is also increase in project costs (Memon et al., 2010). There can be various factors that influence issuance of change order (Enshassi et al., 2010) (Memon et al., 2014). These changes happen because humans cannot entirely understand and describe a project at early phases of a project (Kerzner, 2022). (Cui & Olsson, 2009) points out that preparation for scope changes can lead to low costs and higher success rate.

The process of scope change management is first to generate a request for change and a committee appointed by the client and contractors either accepts the change or rejects the change (Kerzner, 2022). The requested and approved changes are both logged into a change register. A problem faced by the decision makers is to identify and analyze set of alternatives with some evaluation criteria. These evaluation criteria can be sticking to the cost planning, schedule or scope baseline of a project (Kerzner, 2022). Baseline is then revised according to changed scope if a variation order is approved. There could also be contractor generated scope changes called variation order request, and they need to be controlled in the same way as owner generated scope changes (Kerzner, 2022).

Project change prediction and identification is important in project management (Ibbs et al., 2001). Project plans, front-end planning, detailed engineering and project procurement phases can be improved by having a change management system (Anastasopoulos et al., 2010; Hsieh et al., 2004). In this thesis, VOs are considered to be any change that is initiated by contractor or owner, irrespective of approvals. In the light of theory, change orders or variation orders requests are also included in this study irrespective of origin and approval of change. A VO is therefore considered to include all changes initiated by any party whether it gets approved or not.

2.4 Machine learning

This section provides the necessary theory for machine learning, artificial intelligence, data quality and aspects necessary to discuss the results of models developed. Machine learning has found its applications in several fields of studies and businesses. The success achieved by its application in several fields entail its applicability in project management. Project management is a wide discipline and this section for machine learning is developed for the methods tested on one aspect of project management practice.

2.4.1 Artificial Intelligence

Artificial intelligence(AI), generally, is concerned with artificial behavior of man-made objects (Nilsson & Nilsson, 1998). Intelligence means to have reasoning, learning, communication ability and acting in complex scenarios. According to (Nilsson & Nilsson, 1998) AI ultimate goal is to be able to achieve all of these things and possibly more. AI is concerned with development of machines that can perform these actions and be able to explain their behavior.

Alan Turing in in 1950 developed a test called "Turing test" (Copeland, 2000), where the idea is for machine to be able to have a conversation with a human in such a way that a human could not make distinction between a machine and human's conversation. Such a system requires reasoning and inference steps which consists of flexible steps and rigid steps (Wang, 2006). One of the first steps in AI was taken when McCarthy developed a system called "Advice taker" (McCarthy, 1960) in which the system was not programmed but rather it was given instructions of what the system needed to know. Knowledge representation in AI still uses ideas from (Green, 1969) and it's variants. Perceptron as we know of today in modern AI was first introduced by (Block et al., 1962).

2.4.2 What is machine learning?

Machine learning is concerned with automatic improvement of systems through experience (Jordan & Mitchell, 2015). It falls under the umbrella of AI. Its numerous applications include computer vision, voice recognition, facial recognition, natural language processing, robot control and others. Many of the tasks considered only manual are automated using ML rather than programming by hand (Jordan & Mitchell, 2015). The data intensive industries especially consider the applications of ML for example, for energy predictions, predictive maintenance, diagnosis of faults, control of supply chain etc. (Jordan & Mitchell, 2015).

Learning is defined as improvement over a performance measure such as accuracy, by optimizing the model structure (Jordan & Mitchell, 2015). A very large number of ML algorithms exist that solves variety of problems across different domains of applications (Hastie et al., 2009; Murphy, 2012). A ML algorithm by using optimization searches through large domain of space during the training to find optimal function based on a performance metric or loss function (Jordan & Mitchell, 2015).

Machine is a branch of computer science that is based on probability theory, optimization theory and statistics (Hastie et al., 2009). Learning part in machine learning is called training in which any algorithm is able to change its structure and properties according to the given data (El Naqa & Murphy, 2015). The structure and properties of an algorithm is changed systematically using optimization theory (Zaheer & Shaziya, 2019). There can be several ML algorithms however, all of them can be decomposed into three basic

components, representation, evaluation and optimization (Domingos, 2012). Representation is hypothesis plane of a learner, for example in support vector methods the hypothesis plane is to find a linear boundary with maximum distance to the dataset. There can eb several representations such as logistic regression, naive bayes, neural networks etc. Evaluation is the optimization function or loss function, that can either be maximized or minimized (Domingos, 2012). Optimization is how an algorithm walks through its hypothesis plane to find optimal performance on evaluation metric.

A ML algorithm inputs the training data during optimization of model and changes the model structure, but the goal of ML algorithm is to generalize to the samples outside of this training data. The model is then evaluated for generality on test time using a separate set of inputs, the reason is that in real world data scenario it is highly unlikely that the same samples as input will be repeated (Domingos, 2012). The success therefore, on training data will not always mean success on testing data (Domingos, 2012).

There are different types of ML algorithm classifications, but they all fall under two types supervised and unsupervised learning methods. In a supervised method a ML algorithm has access to both the training inputs and training data outputs while training. The ML model then maps these inputs through a function to their respective outputs (Jordan & Mitchell, 2015). In an unsupervised problem the ML algorithm does not access to the desired output and the optimization is done to find pattern in data under assumptions about structural properties of data (Hastie et al., 2009). The RQs put forward in the thesis are only applicable for a supervised learning problem so unsupervised learning algorithms will not be presented.

2.4.3 Types of Machine learning

The two main type of tasks in machine learning are regression and classification problems. In a classification problem the outputs are discrete while in regression problems outputs are continuous (Domingos, 2012).

Classification can be done for supervised and unsupervised problems. In unsupervised learning the objective is to find classes by finding the optimal structure in the data. In a supervised problem labels are visible to the model during training. Supervised classification has several types including, logic based techniques (e.g. decision trees), perceptron based techniques (e.g. multi-layer perceptron), statistical learning techniques (e.g. Bayesian network), support vector machines and instance based learning (e.g. k-nearest neighbor) (Soofi & Awan, 2017). Most common metric for model performance is accuracy for binary classification problems (Dogan & Tanrikulu, 2013). Accuracy can give an estimate of how reliable a model can be on different samples when tested on data outside of training data. Classification models used in this thesis will be discussed in the chapters below.

Regression problems have continuous values as their outputs. Regression problems can be modelled using statistical techniques or machine learning (J.-C. Huang et al., 2020). Statistical methods include quantile regression (Hong et al., 2019), exponential smoothing models (Huang, 2018) etc. ML models are numerous and only the relevant models will be explained in detail in the following chapters.

2.4.4 Optimization theory

Optimization theory has advanced greatly since 1940. One of the reasons for its growth and applicability is the advances in computer science which enables its application to larger set of problems (Pierre, 1986). The concept of systems is essential to understand optimization theory. One of the important properties of a system is its describable nature. A system is any entity that receive inputs and produce certain outputs while maximizing an objective function (Pierre, 1986). A system, however, requires a concept of state for complete describing it. For example, a system with equations which takes set of inputs and maps them to a certain output, the knowledge of state of system at certain time is required to describe the system.

Many of the ML problems take form of convex optimization in which a loss function is minimized (Bubeck, 2014). A convex optimization assumes that a solution exists for a closed domain and there is a supporting hyperplane, for extensive details and mathematical explanations one can refer to (Bubeck, 2014).

2.4.5 Loss functions

Loss functions determine the improvement and performance of a ML algorithm (Granger, 1999). Different loss functions exist for different types of ML problems. For classification problems such as binary classification where the targets have two classes a simple 0-1 loss can be used (Wang et al., 2022). In a 0-1 loss function if sign of prediction matches that of the training label, then loss value of 0 is given and vice versa.

Classification loss functions are numerous but only some are covered here for understanding the importance of selection of loss functions while optimization of models. A perceptron loss function acts similar to 0-1 loss but instead of a constant 1 loss in case of mis-prediction it gives absolute value of prediction as loss (Wang et al., 2022). Another loss function used for classification is logarithmic loss, where the loss is dependent on sample prediction probability value. If the probability of prediction is greater for a true label, then loss is smaller in this case. This type of loss is similar to sigmoid loss function which is implemented in neural networks (Kohonen et al., 1988).

In regression, squared loss is one of the most common loss functions (Wang et al., 2022). This loss calculates the squared error between true values and predicted values. If a predicted value is too far from true value, then gradient for this loss is large which can result in faster convergence while performing convex optimization as explained in section 2.4.4. One of the drawbacks of squared loss is that it is prone to larger outlier impact on models (Wang et al., 2022). Another loss function used in regression is absolute loss which gives absolute value of error between predictions and true values as loss. An advantage over squared loss in this case is that it is more robust towards outlier impacts on the model (Wang et al., 2022). The gradient of absolute loss is constant, and it does not depend on the value of loss which would impact efficiency of optimization. Both of these squared and absolute losses can also be taken as objective function while training the models.

2.4.6 Data Preperation for ML

Data quality check and improvement is essential for ML model performance. The amount of data that is being generated by different industries is growing exponentially due to advancements in information technology (IT) (Naeem et al., 2022). Any knowledge discovery task put forward requires some data preprocessing, in which the goal is to clean the raw data (Eyob, 2009). Different methods exist in literature for complete

lifecycle methodology of data mining such as cross industry standard process for data mining (CRISP-DM) (Wirth & Hipp, 2000), knowledge discovery in databases (KDD) (Matheus et al., 1993) any more. Each of these methods have a step in the process for data pre-processing but they don't describe the problems of data quality in detail. Data cleaning and quality assurance of data is also dependent on type of task for a ML algorithm such as regression or classification.

Data quality is defined by (Morbey, 2013) as "*the degree of fulfilment of all those requirements defined for data, which is needed for a specific purpose*". According to this definition data quality is dependent on the type of task as well as type of algorithm used for that task. In the data mining research there is a principle called "garbage in garbage out" (Kim et al., 2016) which means that if data is of bad quality in inputs of an algorithm the results from the algorithm might be the same.

The first phase of data preparation is to understand the data. This can be accomplished by several techniques such as, missing value count and handling of these missing values (Aydilek & Arslan, 2013). Missing values can be a result of incorrect measurement, incomplete surveys, or incorrect data logging. Outlier handling is another technique in this data understanding step where an outlier is either removed or imputed with a reasonable value (Johnson & Wichern, 2014). An outlier is any observation in the dataset that is inconsistent with the rest of sample observations (Hawkins, 1980). High dimensionality of dataset is another issue that needs to be resolved, it refers to the fact that number of features might be too large for a model (Khalid et al., 2014). Redundancy is another problem that might be encountered during raw data processing, it means to have duplications in the dataset and is handled by removing the duplicated instances in a dataset (Huang et al., 2008). This effect is highly relevant for classification problem since it effects the classifier performance significantly (Bosu & MacDonell, 2013). Noise in a regression problem is defined as irrelevant data that reduces the predictive power of a regression algorithm (Chandola et al., 2009).

The next step in data preparation is to clean the dataset. Missing values are handled using imputations, which can consist of different approaches for imputation. Four of these approaches are deletion (Aljuaid & Sasi, 2016), hot deck (Strike et al., 2001), imputation based on an attribute (Grzymala-Busse & Hu, 2000) and imputation based on non-missing attributes (Magnani, 2004). High dimensionality problem of datasets is handled using dimensionality reduction techniques which reduced the number of features that represents the dataset. Different approaches exist for dimensionality reduction such as filtration, that selects features based on a criteria such as correlation (Ladha & Deepa, 2011). Wrapper methods are also used for dimensionality reduction which selects a feature based on the performance of an algorithm (Chandrashekar & Sahin, 2014). Projection methods of dimensionality reduction are also used for dimension reduction such as principal component analysis (PCA) (Jolliffe, 2002).

More data leads to better predictive power and it is therefore also important to be able to integrate different data sources and data fusion from multiple sources are important for better results (Bansal, 2014; Dong & Srivastava, 2013; Yang et al., 2014). The research question put forward in the thesis poses another data cleaning challenge called zero-inflated data (Tu, 2006) since the prediction is for the count of number of variational orders therefore, the targets must contain large amount of zeros. Many of the classical ML models assumes normal distribution of target variable (Rebala et al., 2019). The transformation of target variable is done to ensure normal distribution of residuals

(Atkins & Gallop, 2007). Several transformations exist such as square root and log transformations of target but in case of zero-inflated data these transformations would not distribute the zeros present in the target variable (Atkins & Gallop, 2007).

Another problem with real-life datasets are unsupported type of data in the dataset such as strings, these features can be handled using several methods including one hot encoding and label encoding (Sharma & Yadav, 2021). Project data contains several columns such as discipline for an activity that can be transformed using these methods for compatibility with ML algorithms. One-hot encoder converts each category into a new feature column with binary values specifying the presence of a category in an instance. A label encoder, however, converts the strings into values between 0 and number of categories in just one column. For example, using a label encoder on text data of [London, London, Paris] the transformation is then mapped to [0,0,1] in one column. While one-hot encoder will convert these into separate columns for each category. The choice of encoder depends on the ML algorithm and type of problem at hand. A typically overlooked problem is multicollinearity in the multi regression variables. Collinearity means to have linear relation between two variables (García et al., 2015). A common solution for removing this collinearity is calculation of variance inflation factor for each variable and then remove the highly collinear variables from the training features (García et al., 2015).

2.5 Types of data

Most of the real world processes are Spatio-temporal (ST) in nature (Atluri et al., 2018). RQs put forward in the thesis implies that the data is temporal since the target is prediction of next week variational orders in the time domain. Another dimension of the data is given by its features which are spatial features thus the data type is spatio-temporal data (Atluri et al., 2018). A special aspect of this data is that its measurements depend on the historical information. Many of the classical ML algorithms ignore these dependencies by making assumptions about independence and identical distribution of data (Atluri et al., 2018). These problems can however, be decomposed into spatial or temporal problems by transformations and assumptions about data, an example of this transformation is modelling of brain activity where the outcome is to predict the time at which similar activity of brain will happen, in this case the time domain is modelled as objects and measurements are modelled as features (Liu et al., 2013).

Different data types exist for ST data. Event data is the type of data which is characterized by location and time which tells about when and where an event occurred (Atluri et al., 2018). Point data is another type of ST data which consist of a space time field an measurements over it such as temperature, humidity etc. raster data is yet another type of data in ST data which denotes the measurements in time at fixed locations and at fixed intervals in a time space field (Atluri et al., 2018) example of such data include images and video sequences.

These types of data can be modelled in several ways for formatting into an acceptable pattern for an algorithm (Atluri et al., 2018). These data formats are point format, sequence format, matrix and tensor format (Wang et al., 2020). Our problem proposed in RQ is of raster type of data that can be modelled as time series, spatial map or ST raster data (Wang et al., 2020). These can be converted to different formats such as sequence, graph, matrix (2D) and Tensor (3D) as explained in (Atluri et al., 2018). Different models require different types of formats of data and therefore the modelling of data is important for different models (Wang et al., 2020). For example, in deep learning

sequence format can be modelled using LSTM, CNN etc. which will be explained later in the report. For the purpose of in-depth understanding of PM plan data three different datasets will be prepared to capture large range of model capabilities and verification of results.

2.6 Machine learning models

As pointed out in section 2.4.6 the target variable for regression is zero inflated and thus many ML models cannot be applied to this type of problem due to assumption if independent and identical distribution by those ML models (Atluri et al., 2018). However, tree-based models are highly non-linear (Kou et al., 2003).

2.6.1 Decision tree

A classification tree works by splitting a node on some criterion such as impurity in a space of all samples such that the resulting child nodes has lowest impurity (Loh, 2011). It starts at the root node and iteratively searches for nodes that minimize child nodes impurity while evaluating some stopping criteria. Two modifications of this basic algorithm are C4.5 and CART (Messenger & Mandell, 1972; Quinlan, 2014). C4.5 algorithm uses entropy as a measure of impurity of child nodes while CART uses Gini index as measure of impurity of child nodes.

For regression problems this algorithm works in the same way except that targets are now continuous and measure of impurity is changed to sum of squared deviations from mean (Loh, 2011). The resulting models are piece wise continuous (Loh, 2011) and thus zero-inflated data can be fitted using decision trees.

2.6.2 Ensemble methods

Ensemble methods combine multiple learners to produce an output. The learners in this method are called base learners (Sagi & Rokach, 2018). Typically these learners results are fused via some voting mechanism to achieve better performance than single base learner (Zhou, 2012). Ensemble methods can use any type of ML algorithm as base learners. The idea of ensemble is to combine multiple opinions about a decision and make a better decision than individual opinion (Sagi & Rokach, 2018). Several advantages exist by using ensemble methods, these include the advantage of decreased risk of local minima while optimization , complex representations, better handling of class imbalance, distribution independence, reduction of curve of dimensionality (Sagi & Rokach, 2018). Curse of dimensionality points to the fact that by having more features the search space for an algorithm explodes exponentially and thus the model is more likely to overfit and thus less generalize to the problem (Sagi & Rokach, 2018).

The modern ensemble methods are inspired by the work of (Mennis, 2006) who points out that for an ensemble method to be effective it needs to fulfil certain conditions such as diversity, independence, decentralization and aggregation. Diversity means that each base learner should have private information. Independence means to not be affected by decisions of other base learners. Decentralization means that a base learner specializes based on given conditions and inputs. Aggregation points to the fact there should exist a method of combining each individual opinion to a collective opinion (Sagi & Rokach, 2018).

These conditions for a good prediction are fulfilled differently by each method. For example to make sure diversity condition is met by input manipulation, use of different

base learners, partitioning of data (Sagi & Rokach, 2018). Several methods also exist for output fusion such as weightings method and meta-learning method. In the following subsections some ensemble methods are explained that were used in the thesis.

2.6.2.1 Extra trees

Extra trees takes a classical drop-down procedure to build an ensemble learner (Geurts et al., 2006).It uses decision tree as a base learner. It takes two hyperparameters which specify how many random features will be selected for each tree and the other specify minimum sample size for splitting a node (Geurts et al., 2006). It the problem is a classification problem then outputs are fused using majority voting mechanism and if problem is regression, then base learner outputs are fused by using arithmetic average of base learner's outputs. The maximum number of trees can also be specified so that the model complexity is kept in check. However, in this method same sample of inputs are used to construct each base learner (Sagi & Rokach, 2018). It has been successfully applied to brain tumor segmentation problems (Soltaninejad et al., 2017).

2.6.2.2 Gradient boosted trees

Gradient boosting trees are a specific type of gradient boosting machine (GBM) (Friedman, 2001). In this method training of each base learner is dependent on other base learners that have already been trained. These GBMs work such that the correlation between negative gradient loss of base learners are maximized (Sagi & Rokach, 2018). Since the gradient of loss function is calculated between base learners that are trained on the residual of the last base learner in a sequential manner therefore, loss function must be differentiable in this case (Sagi & Rokach, 2018). Usually, the base learners in this case are shallow but larger in number and therefore it is important to select an optimal number of trees as base learners. If number of trees are large then the model is prone to overfitting, one remedy is to use stochastic gradient boosting method (Friedman, 2002). In the stochastic GBM the training is done on smaller sets of data extracted from original training data. These GBMs can be adopted for classification and regression problems both.

2.6.2.3 Ada Boost

Ada boost was developed by (Freund & Schapire, 1997) and is considered one of the most popular ensemble method. The focus in this method is given most to those samples in the training which had most loss (Sagi & Rokach, 2018). This focus is distributed across samples by using weighting and in first iteration of the algorithm each sample has equal weight, whereby in subsequent iterations weights are adjusted to put more emphasis on samples that are most incorrectly predicted. This weighting is also given to the base learners based on their performance. If a classifier is being trained then outputs of base learners are fused together by a voting classifier and in case of regression weighted median is used as ensemble models output (Drucker, 1997).

2.6.2.4 Cat Boost

Cat boost is a gradient boosted method for decision trees with two additional innovations, one is ordered target statistics and other is ordered boosting (Hancock & Khoshgoftaar, 2020). This method is particularly useful for datasets containing different types of data such as ordinal, categorical, and continuous data. Catboost is particularly successful in these types of heterogenous datasets (Hancock & Khoshgoftaar, 2020). Catboost was proposed by (Prokhorenkova et al., 2018), one of the major improvements in this method is how this algorithm handles categorical data based on their cardinality. This

algorithm is able to deal with high cardinality features. For low cardinality categorical feature catboost uses one-hot encoding. When categorical features have high cardinality then catboost does not use one-hot encoding instead it uses target statistics which is calculated from output values associated with particular categorical input (Prokhorenkova et al., 2018). Another technique employed by catboost is use of indicators to map categorical features to numeric values. Catboost also does not employ simple decision trees but rather it uses Oblivious decision trees in construction of an ensemble (Lou & Obukhov, 2017). Catboost also supports feature interactions which the original implementation called feature combinations. These feature interactions can better help understand how a model is interacting with the inputs and how two features interact to make a prediction in whole ensemble.

2.6.3 Light GBM

Light GBM is also a method using GBDT. It was developed to tackle big data challenges such as curse of dimensionality. It introduced two improvements over GBDT one is to use gradient based one side sampling in which any instance that has more error or loss contributes more to information gain criterion for node split (Ke et al., 2017). Another improvement implemented in the algorithm is feature bundling which bundles sparse features that contain large amount of zeros using a bundling algorithm, this step thus reduces the complexity of model and feature space (Ke et al., 2017). Light GBM official publication (Ke et al., 2017) does not mention the use of categorical feature encoding but their online documentation mentions a technique light GBM uses to encode categorical features (LightGBM) . This ensemble method has also been applied successfully in various domains.

2.7 Deep learning models for Spatio temporal data

Conventional ML models require large amount of engineering and considerable domain expertise to construct features and handle data quality problems to make a good performance model (LeCun et al., 2015). Deep learning methods are representation learning method developed to extract feature representations using linear or non-linear modules. Representation learning methods are those methods that can extract useful characteristics from raw data (Bengio et al., 2013; Huang & Yates, 2012). Subtasks of representation learning methods include feature selection including feature extraction and distance metrics calculations (Tu & Sun, 2012).

Deep belief network is thought to be a breakthrough in deep learning (Hinton et al., 2006). Artificial neural networks are the basis for development of deep learning (Hinton & Salakhutdinov, 2006). A neural network (NN) is made by components called neurons or cells which produce some output using activation functions and weighting of inputs. These cells are stacked horizontally and in layers to form a neural network which adjusts it's weights of neurons based on backpropagation algorithm (Liu et al., 2017). This back propagation gradient calculation is what gives deep learning models an edge over traditional ML models for feature extraction (LeCun et al., 2015).

In backpropagation gradients are calculated with respect to each neuron using an objective function which is just an application of chain rule for derivatives. A simple neural network architecture is a feed forward neural network which learns mappings from an input to output of fixed size (LeCun et al., 2015). In this network multiple layers are usually present where each unit or neuron calculates the weighted sum of its input and applies an activation function which is usually non-linear like rectified linear unit (ReLU).

The layers between input and output are called hidden layers (LeCun et al., 2015). Many deep learning architectures are present however, only those will be described here which are being experimented with in this thesis.

2.7.1 Convolutional neural network

Convolution neural network (CNN) receive 2D input data with channels information in depth such as images, an example of input is shown in figure 1. CNN can also handle 1D and 3D data if required since convolution operator used by this architecture is expendable to such domains. An assumption made by CNN is that features are spatial (Albawi et al., 2017). It has been successfully applied to many image tasks due to the property of independence of temporal domain, for example in classification of an image as a cat or dog would not depend on where the cat or dog is located. A CNN uses convolution operator to reduce number of parameters that a same functioning neural network will have for same task (Albawi et al., 2017). A convolution operator is mathematical operator that is applied locally to some region. The shape of convolution is specified for entire space and remain constant for an operator and number of convolutions is specified by a parameter called filters (P. Huang et al., 2020). To further reduce number of parameters of a CNN layer stride of more than 1 is given as a parameter. Stride means how the convolution slides on the received input. Stride of 1 means that the filter will move one step at a time through input space. One disadvantage of convolution is that by sliding the convolution operator on whole feature space the effects of corner features and their interactions are diminished, to remedy this padding is applied (Albawi et al., 2017). In this method zero values are padded to the input space such that the convolution operator is fully able to capture the feature space. After a convolution has been applied a non-linear function is applied same as neural networks. A typical convolution operation is shown in figure 2 where 3 depth input is convolved by two filters to produce two depth output with same shape as input due to zero padding.

A CNN is structured in a such a way that a convolution layer is followed by maximum pooling or average pooling (LeCun et al., 2015). This pattern is repeated multiple times though the architecture. The advantage of using several layers followed by these pooling layers is the capture of semantic relations between feature space. Stacking several of these convolution and pooling layers with different filters extracts different features from the inputs and hence in a CNN each layer might extract different information. An example of this is (Dumoulin & Visin, 2016; Lee et al., 2009) Which shows how different layers extract different information such as corners, colors etc. in each layer. These extracted features depend on the filter applied to the layers.

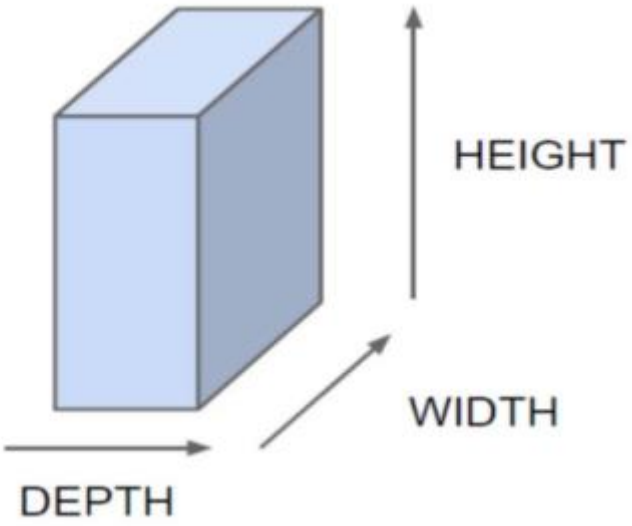


Figure 1: Input format for CNN

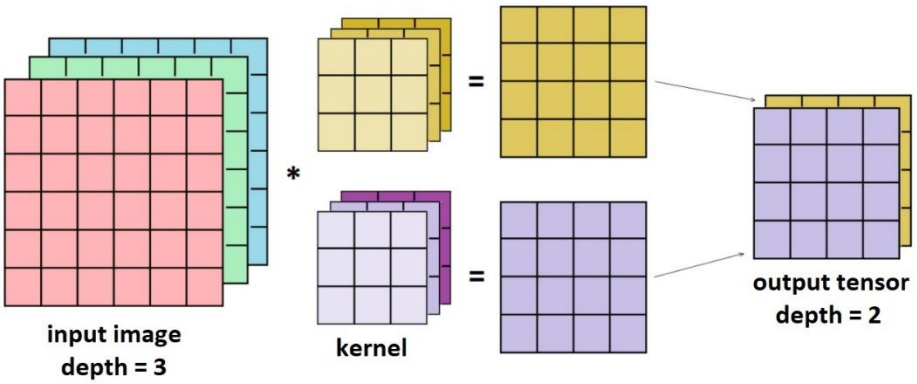


Figure 2: Convolution application to data

2.7.2 Recurrent neural networks

Recurrent neural network (RNN) are special types of neural networks where the state of network after a sample are not lost (Lipton et al., 2015). RNNs are capable of passing information from state to a different state selectively where the sequences are not independent. The inputs and/or outputs of RNN can be sequences. These sequences can be time sequences or time independent such as a sentence. Learnings for RNN are difficult because of long term dependencies (Lipton et al., 2015) and varnishing or exploding gradients during backpropagation (Bengio et al., 1994). A simple RNN cell is shown in figure 3 where input x_t and previous cell state $h_{(t-1)}$ is passed through a function to produce cell state $h_{(t)}$ at current position, this cell state h is again passed through a function to produce output $z_{(t)}$ at time t. Although they were designed to store long term dependencies they have been shown to not store information for long times (Bengio et al., 1994).

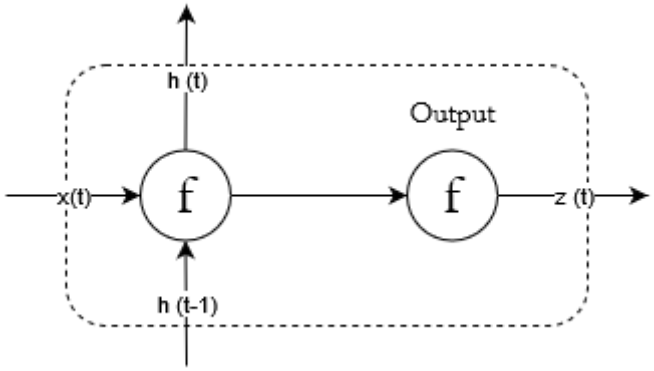


Figure 3: Basic RNN cell structure

2.7.3 Long short-term memory neural networks

Long short-term memory (LSTM) are special type of RNNs that uses a specialized hidden unit in a cell for remembering information (Hochreiter & Schmidhuber, 1997). It also incorporates a feedback mechanism from outputs of previous layer to whole network (Grossberg, 2013). The main ability of LSTM lies in the fact that it is able to capture information over time in a sequence (P. Huang et al., 2020). The original LSTM proposed consisted of a cell with two gates for input and output. Several modifications of original LSTM architecture exists and the most common is the one put forward by (Graves & Schmidhuber, 2005) which included sigmoid function as gate activation. A forget gate was proposed by (Gers & Schmidhuber, 2001) which was able to forget information that was unnecessary. A structure of LSTM cell is shown in figure 4 where x_t is the input at time t and C_t is output by a single cell at time t.

The working of an LSTM cell is governed by the gates in a cell. If input gates get activated then new information is written, similarly previous information is controlled by the forget gate and output gate controls what to output.

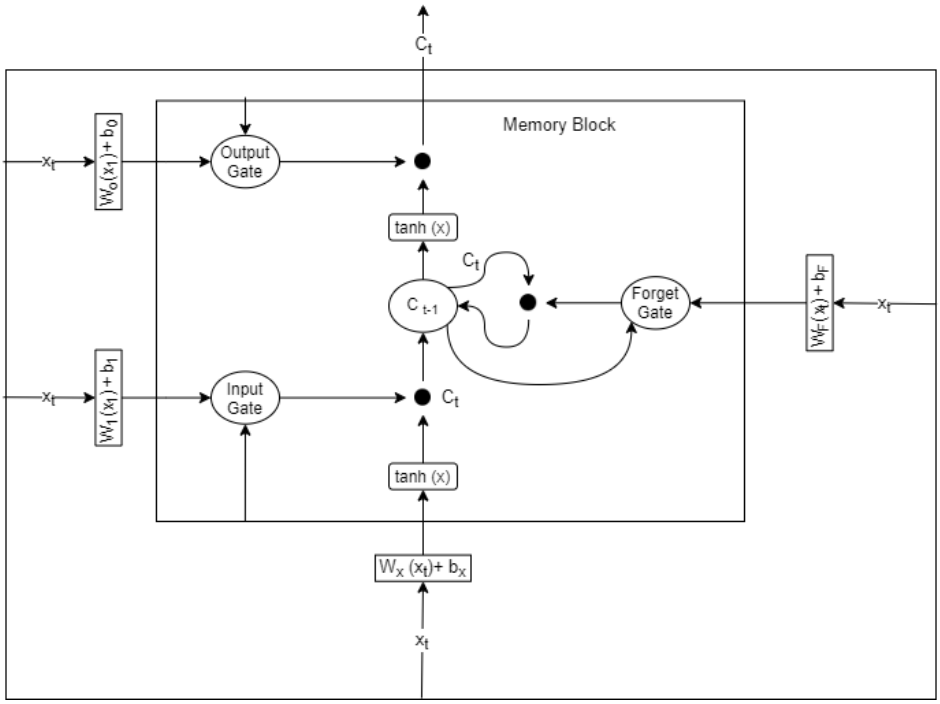


Figure 4: LSTM cell structure

2.7.4 Hyperparameters tuning

Hyperparameters for deep learning models are sensitive and difficult to tune (Han et al., 2020). Results such as (Schratz et al., 2019) show that models performance is dependent on hyperparameters of the model. (Bergstra et al., 2011) suggest that tuning and optimization of hyperparameters should be formally included in the learning process of a model. Several approaches exist for hyperparameter tuning such as sequential model based global optimization, gaussian process approach and tree structured Parzen estimator approach (TPE) (Bergstra et al., 2011). The algorithm used in this thesis is TPE algorithm. This approach considers tree structure generative process which means that the search space is discrete values with two nodes such as choose number of layers and then choice of cells in a layer. The optimization is performed using expected improvement method (D. R. Jones, 2001). This TPE method outperforms random and sequential search for hyperparameters tuning (Bergstra et al., 2011) and therefore, it was employed for hyperparameter tuning of the models developed.

2.8 Multi-instance learning

Multi instance learning (MIL) is a type of ML where instead of a single instance an event is represented by multiple instances (Dietterich et al., 1997). It is concerned with supervised learning where most of the research has been on binary classification problems. These binary classes are named positive and negative for the context of MIL problem. A special characteristic of MIL is its learning examples, where an example consists of a bag that is labelled as positive or negative. Each bag can have multiple instances and even different number of instances (Foulds & Frank, 2010). An example of such data could be project data where each week (a bag) consists of different number of activities (instances) and the task could be to classify a bag as positive or negative label. However, strict assumptions are made for different algorithms in MIL methods. Some algorithms make presence based assumption which means that a bag is only positive if it contains one or more positive instances (Foulds & Frank, 2010). Some algorithms make threshold-based assumptions which means that a bag is positive if it contains more than certain number of positive instances. Similarly count based assumptions are made by some models.

Two of the algorithms are tested for their performance on RQs namely multiple instance algorithm with a nonlinear kernel (MICA) and statistics kernel (STK) support vector multiple instance learning. MICA was proposed by (Mangasarian & Wild, 2008) which makes the assumption of having at least one positive instance in a positive labeled bag while in a negative bag all instances must be negative (Mangasarian & Wild, 2008). This is a strong assumption about our project type of data which might not be realistic, but the model was tested to by assuming that in each week where there is no variational order all activities must be of same characteristic. However, this algorithm has property of sparse solutions (Mangasarian & Wild, 2008) which holds for the case of count of VO prediction problem. STK algorithm applies kernel on the instances followed by support vector algorithm (Gärtner et al., 2002).

2.9 Model explainability

For a long time ML and AI methods were considered black box models since the model's internal structure for decision making were unknown (Wilson et al., 2019). Due to non-linear interaction of deep learning models the lack of explain ability makes then non-transparent. This the reason these methods are often called black box methods (Samek

et al., 2017). This lack of explainable behavior, therefore, is detrimental in certain cases such as medical diagnosis and drug prediction problems. Explainable behavior of a model is important for several purposes such as model verification, learnings from the model, compliance, bias, and discrimination among others.

One such method of explaining a model's prediction is to use attribution (Sundararajan et al., 2017). The purpose of which is to understand inputs and outputs behavior of a model. These attributions can be helpful in later development of rule-based system as well. A new method was developed to find these attributions called integrated gradients (Sundararajan et al., 2017). In this method a baseline is set, and gradients are calculated along a straight line from baseline to actual inputs while accumulating gradients along this path. A critical factor in this method is selecting baseline and the author of the method recommends baseline that results in near zero score. We will use these integrated gradients attributions to explain our best model and develop some understanding of the models. We will also employ another metric called linearity measure (Agarwal & Das, 2020) from "alibi" toolbox for python to measure instance based linearity. It tells us how linear a model is around an instance. This will help understand how linear our model is in its predictions.

2.10 Previous research

ML use in project management has been studied for past few years. Different PM areas are studied by different authors, (Mahdi et al., 2021) has studied software project management success and failures using ML methods, (Ma et al., 2021) has studied risk factors for construction PM, project delivery was studied by (Aldana et al., 2021), (Peña et al., 2019) studied project control, (García et al., 2017) also studied project control using ML techniques, many researchers have conducted studies involving prediction of earned value and improvement of project planning (Chen et al., 2016; Ling et al., 2008; Ling & Liu, 2004; Saglam, 2017). Recently researchers are focusing on the text based data of projects for predictions of project metrics such as project duration, a similar study was done recently by (van Niekerk et al., 2022).

The oldest research found was done by (Badiru & Sieger, 1998) where they attempted to use neural networks for prediction of project returns. The goal of the research was to improve the decision process of project financing using an integrated decision model. They also incorporated stochastic nature of investments (Pint, 1992) into the model to better account for uncertainties in project financing decisions. They concluded that using NN demonstrated better results than traditional simulation methods for project decisions (Badiru & Sieger, 1998). (Vahdani et al., 2014) studied project selection problem in construction industry using ML techniques. They also showed that the neural networks and ML kernel methods perform superior in prediction of project selection. (Chou et al., 2015) used AI methods to predict project award price. It considered contractor case for bids such that the contract award amount can be predicted by contractors. They employed multiple ML methods and ANN methods and demonstrated a superior performance in prediction of contract award price.

(Lu, 2002) also used artificial neural networks for enhancement of project evaluation and review technique (PERT) and were able to enhance beta distribution for better estimation of an activity duration. The merge event bias, which is a common problem with traditional deterministic methods can be solved by using PERT method of estimations (Ahuja et al., 1994), however, the authors show that the simplifications made by estimation of mean and variance using this method results in inaccurate duration

estimates. The maximum possible error using simplified version of PERT is 33% and 17% for mean and variance respectively (Cottrell, 1999). (Lu, 2002) demonstrated by using an ANN that the estimations of PERT can be improved by employing ANN.

Project delay risk has been studied by (Gondia et al., 2020). Since project delays are considered a global problem (Assaf & Al-Hejji, 2006), and traditional approaches require large amount of data such as activity durations, their probability distributions for simulations they employed decision trees to construct a classifier and identified key risk factors for a project delay and develops a dataset using identified risk factors. They included several project documents for construction of project dataset such as contracts, specification, change orders, baselines, status updates, resource calendars and risk registers. The total dataset consisted of 58 projects data and the target variable was time overrun. A limitation mentioned is that the model is applicable to the domain that it was tested on however, this is a general case with any data model (Le Roux, 2008). (Wauters & Vanhoucke, 2016) used AI methods to directly forecast final project durations, they employed ML methods and compared their results to traditional EVM and earned schedule (ES) methods for benchmarking. Their result demonstrated better performance than EVM and ES methods in early and mid-stages of a project. However, a limitation for their study is that the training and test set should be similar to one another. A novel method using minutes of meetings technique was proposed by (van Niekerk et al., 2022) which has the ability to predict project duration. They used project site minutes of meeting data to predict final project duration and showed accuracy of above 80%. The input data included several features such as field engineering queries, site instructions, delay information and several others including text data.

(Ko & Cheng, 2007) used a combination of neural networks, fuzzy logic, and genetic algorithms to predict project success. The model developed by them used continuous monitoring and assessment during project life cycle a selected factors dynamically for prediction of project success. They used a method developed by (Russell et al., 1997) called continuous assessment of project performance and improved the predictions of project success or failure. (Hsu et al., 2021) identifies and explains concept of inter project dependencies and builds deep learning model using ANN and RNN structures to capture these dependencies and predict project success. They compare ANN and RNN for their performance on capturing these dependencies between projects.

Software project management has also been an important driver for ML research in PM. (He et al., 2012) used classification methods of ML to predict if a project is likely to have a defect. Their study is important in the sense that they show cross project data is better at prediction of defects and that training data from different projects can lead to better performance for ML methods. They also studied automatic selection of training data in presence of cross project data for better generalization of ML algorithm. (Pospieszny et al., 2018) tested and developed three ML models and using ensemble method predicted effort and duration for software projects. They show the superior estimation ability of ML methods to traditional estimation methods. (Wen et al., 2012) reviewed 84 different studies that utilized ML methods for project prediction tasks.

Prediction of project performance has also gained interest in research community. (Ling & Liu, 2004) studied 33 projects of a specific contract type and identified 65 factors that affect the project performance and later using artificial neural networks predicts the project performance. They showed that at least six metrics were predicted accurately, including project intensity, delivery speed and equipment quality. (Ling et al., 2008) used

different projects data and suggested improved scope management technique using the models developed. They employed traditional ML algorithm for model development and predicted time performance, cost, and quality performance. The limitations are that only small number of projects are used in training data. (Saglam, 2017) applied ML techniques to predict earned value at completion of a project. (Aldana et al., 2021) studied forecasting using AI methods for individual projects and a portfolio of projects. For individual project data they improved the accuracy of cost prediction, final duration, and early warning duration. For portfolio of projects AI methods improved the cost forecast by 87% and improved the aggregated portfolio value calculations.

Project changes has been studied in software project management. A study conducted by (Malhotra & Bansal, 2014) suggests that prediction of changes in software releases is better if inter project data is used. They identified an improvement from 30% to 67% accuracy for a classification task by using inter project data. (Chen, 2015) studied extensively the effects and causes of change orders and developed a linear and decision tree model for prediction of change percentage in whole project. However, the study is focused on prediction of change percentage by looking at several projects' data and does not provide insight into the RQ put forward in this thesis. One of the conclusion of this study directly translated to the RQ put forward, the study argues that a tool for change prediction could improve project performance (Chen, 2015). He considers change in design, construction, and total change prediction problem. (M.-Y. Cheng et al., 2015) attempts to understand the impact of change orders on labor productivity and argues that combination of fuzzy logic with genetic algorithms predict impacts of change order on labor productivity better than purely ML and AI approaches. However, a limitation is that simple models are tested for comparison of performance in this study. (Choi et al., 2021) also studies change order impacts in module 4 of their model in which causes of change orders are identified. The input data used in this case also contains change order reports and other text data and therefore some natural language processing (NLP) techniques are applied to understand cause of changes. They classified these causes using ML classification methods. To the best of the authors knowledge no study has been done that attempts to predict the count of change orders using a single project data.

2.11 Summary of theoretical study

Machine learning has found its application in various industries and domains of knowledge including project management. ML methods consists of many algorithms that can be implemented in different settings based on the assumptions made on data some of these were presented in this section and will be explored in experimentation and discussed. The data format for different ML methods also differs and therefore, several datasets using PM plan data will be generated to satisfy the input requirements for these models. Project management literature suggests that most of its focus on applications using ML have been on performance metrics or scheduling problems, however, change management and project control has not been explored largely. Due to this research gap in this intersection of PM control, change management and ML methods most of the ML methods explained in this section will be explored and discussed. The choice of ML methods also depends on the RQ formulated, since this is a supervised problem therefore, only supervised learning methods are explored and discussed. Variation orders has historically been a driver for project changes and overruns, this identification from PM literature also increases the importance of the RQs formulated and the impact of implementing ML for project control. Uncertainty in projects can also be reduced by implementation of these methods developed in this thesis.

3 Method

This section is dedicated to explaining the steps and sources of data, data extraction, quality assurance and pre-processing of data.

3.1 Literature search

One of the most important parts of research is literature study and theoretical understanding of relevant concepts. To find relevant articles web of science was used along with google scholar. On web of science database advanced filters were employed to filter articles that were of interest for this research. One of the advanced filters used was "TS=(*heuristic* OR evolutionary* OR genetic algorithm*) AND ((TS="project control*" OR SO="project control*" OR CF="project control*") OR (TS="project evaluation*" OR SO="project evaluation*" OR CF="project evaluation*") OR (TS="project assessment*" OR SO="project assessment*" OR CF="project assessment*"))" to find applications of advanced methods for project control which returned 50 results. To find ML and AI applications on project control, change orders or earned value following search query was used "TS=(*project management* OR Artificial Intelligence* OR machine learning*) AND ((TS="project control*" OR SO="project control*" OR CF="project control*") OR TS="Earned Value*" OR SO="Earned Value*" OR CF="Earned Value*") OR (TS="Change orders*" OR SO="Change orders*" OR CF="Change orders*"))" which returned 946 results which were sorted for highest number of citations and skim reading was performed for abstracts before selecting an article for in-depth study. To find other articles on AI applications in project management following search query was used "((((AB=(Predict project)) AND (AK=(Machine learning)) OR AK=(artificial intelligence))) AND AK=(project)) AND TI=(project*)" which returned 99 records which were filtered by skim reading abstracts and selecting relevant literature. Project management literature was found mostly on google scholar with use of google dorks which matches the exact keywords in text or in title of an article, example usage to find keywork project scheduling is "intext:"project scheduling"" which will find any article containing the exact match in text of an article. The keywords used for literature search were "Project scheduling", "project control", "work breakdown structure", "project planning", "machine learning", "Artificial intelligence", "project performance", "data quality", "supervised machine learning", "spatio-temporal data", "convolution neural network", "long short-term memory", "neural network", "model explainability", hyperparameter tuning", "multi instance learning", "data pre-processing", "ensemble methods" and complex combinations of the above searches to filter the results that were relevant for the literature review.

A custom search was done on web of science database using following query to get a more focused search results on previous research done on selected topics "((((AB=(Predict project)) AND (AK=(Machine learning)) OR AK=(artificial intelligence))) AND AK=(project)) AND TI=(project*)" which returned 99 records of publications. However, this is a strict search and using google scholar search for "intext:"predict project "' a total of 110 results are displayed. A careful read to abstract is done to filter these research paper for only project management papers. The goal of this search and

inclusion of these studies is to motivate further research in applications of ML in project contexts.

The use of google dorks made it easy to filter results as needed for the literature study. Some of the references were used from internet searches such as from documentations and blogs where the relevant information existed. Large number of articles were skim read before selection for in-depth reading. This method provided the advantage of generation of new ideas for inclusion in this research such as multi-instance learning and model interpretation measures.

3.2 Data Description

Raw data for project was provided by one of biggest contractors in Norwegian construction industry. Data was provided for a single project. The data contained in a project management software with sufficient WBS level data. PM software used a database to store different values in several tables which can be extracted via SQL Server or through the software interface by running SQL queries. The data contained information about activity levels and milestones.

The data provided is of Spatio-temporal (ST) type. The natural progress of a project is in time domain and number of recorded variables are spatial features. ST type of data is available in several domains such as retail data, transportation data and economics (Wang et al., 2020).

Data contains period status tables for activities and resources, activities table, resources table, change register table and intermittent tables for joining the data and data aggregation at higher level. Activities table contain a snapshot of activities ata a specific time while period status tables contain historic information about activities with certain features being recorded. Similarly, resources table contain aggregated activities at specific time while period status of resource table contains historic information about resources. Change register table contains all the changes requested during the project and they are connected to resources table which in turn is connected to activities table via primary and secondary keys. Several types of data are contained in different tables such as text data about project such as discipline of an activity along with calculated features, flags and constraint information.

3.3 Data Preparation and Cleaning

Data preparation is an important step in ensuring the data quality is acceptable (Brownlee, 2022) since the data comes from several sources and actors involved in the project phases. Special consideration has been given to the step of data preparation to ensure that the principle of "Garbage in - garbage out" is refrained in the analysis.

Data exploration also called exploratory data analysis (EDA) is an essential step in machine learning (Song et al., 2022). It can give insights into underlying data behaviour, distributions and to understand data. It helps in testing the assumptions about data and thus helps improves data quality (Sánchez et al., 2019). A good EDA can lead to less iterations in the modelling phase of machine learning (Gupta et al., 2021).

Since the analysis and modelling of data will be done in python, therefore, the data is extracted from SQL server in several files and imported into a pandas dataframe (McKinney, 2011). Pandas provide an easy interface and functional possibilities for data exploration and data wrangling. From data mining literature several methodologies have

been put forward including cross industry standard process for data mining (CRISP-DM) (Wirth & Hipp, 2000).

Data preparation includes reading of data extracted from PM software database and merging them to make a one dataset that can be used for processing later. In the first step period status for activities table was read into a pandas dataframe with special care for date fields and date format. In the second step activities table was read to include important information for activities such as reference fields, flags, original plan dates, outline codes etc. Period status dataframe is merged with activities table with a left join such that all the information from period status a is retained and only the matching information from activities is extracted from activities table. This join operation was done on the common primary key for both dataframes.

The filtration of dataset was carried out using knowledge from domain experts. The resulting dataframe had 27764 number of unique activities. For example, description field from the activities table is filtered such that it is not empty since an activity in a project must have a description in the PM software. Cancelled field was analysed to filter activities that were not cancelled. If an activity is always on target, then that activity is flagged by a column in the dataset called 'on_target' therefore, a filter was applied on the dataset such that only those activities remain which is not a fixed activity. Duration field is filtered such that activities which have zero or no duration are eliminated from the dataset. The planning department of the company uses a word 'DUMMY' in the description of an activity that have no significance in the whole project schedule and therefore, any activity containing this word was removed from the dataframe. Reference fields carry important information, for example, in one of the fields any activity that is a milestone is marked and thus can be easily removed. Similarly, hour collectors are removed from the dataset.

The period status tables have two important date fields that are of most importance for this thesis research question, period start and cut-off date. Period start date signifies when a status update has been run and cut-off date signifies till when a status update has been run. According to the research question proposed, only weekly data is required and therefore, in the dataset any row having difference of more than seven days is removed.

In the next step, current progress field is filled. If an activity has not been started yet, then the data does not have any value in the current progress field which will be read as a not a number (NaN) value in the dataframe. Similarly, if the activity has been completed then the PM software logs hundred percent complete for all following instances of an activity which will be duplicates of the first value that occurred when activity completed. Therefore, only the last zero value of percent complete is retained while only the first hundred value is retained. Activities that are only appearing once in the dataset are removed. After application of the above filters the total number of activities remained are 12150 and 179 weeks of data. If an activity is not attached to a calendar in the PM software it is considered as bad planning or bad logging of activity, thus these activities were removed this phenomenon is considered noise in a dataset (Corrales et al., 2018).

According to the data exploration, some of the activities were started before their early start. This phenomenon was discussed with the planners at the company and was considered normal since according to the project conditions an activity can start before it was scheduled depending on the requirements and availability of resources.

Period status of resources is also an important table holding values of interest such as period earned value, expended quantity, planned quantity and contractual quantity. This data is merged with the resulting dataset after filtration with an inner join such that all the filtered activities from period status of activities table are retained and only the weekly data is joined. The resulting inner join will mean that if any activity has no resource attached to it then that activity will be dropped from the dataset. The resulting number of activities that remain are 10508 activities.

This filtered dataframe is saved as a reference for extracting the target variable since the target will only be extracted for activities that are actually available in the dataset. We can call this dataframe "filtered_df" for reference in the thesis. The resulting dimensions of this dataframe are 188731 x 99. Different features for date fields were calculated because most of ML algorithms cannot accept date-time formats as input to the models. Features such as days till early start, days till early finish, days till frontline date, late finish, current early start, current early finish. When a date was missing in the dataframe the resulting column was given a value of -100 so that the ML algorithm can differentiate these values during training. The resulting shape of dataset is now 188731 x 122.

Since for each week different number of activities are present, therefore the dataframe is aggregated on weekly data. The aggregation functions applied are sum, mean, minimum and maximum. These aggregation functions groups the data into weekly data and performs the specified function on given data. This step thus reduces the dimensions of dataframe to 179 x 529. However, by aggregating activity level data to weekly data some information is lost since regression assumes fixed level of aggregation (Berry, 1993) but this aggregation has different granularity level in each week. We can call this dataframe "filtered_df_weekly" for reference in this thesis.

Data quality assurance for regression has been studied extensively (Corrales et al., 2018). Data filtration was applied in this step to retain only relevant features and activities. Noise is removed from the data in this step. Data understanding comes from different explorations, such as counting missing values, visual outlier detection, dimensionality, redundancy, and noise. Data exploration phase in this thesis includes exploring different variables in the dataset that were thought to have most influence on the target variable. Missing values are handled, and features are plotted including multi axes charts to extract some features from important variables that were identified by the domain experts.

3.3.1 Preparation of three different datasets

For testing different ML algorithms, three datasets were prepared according to the input of an algorithm. First dataset is simple tabular data where each row has it's own target value and each row represents a week's aggregated data. We will refer to this dataset as dataset 1. The second dataset is shaped in a three-dimensional format such that each sample is explained by time dimension, spatial dimension, and number of features. We will referrer to this dataset as dataset 2. This type of dataset is suited for application of CNNs. The third type of dataset is special type of dataset constructed for application of multi-instance-learning (MIL). This dataset is a special type of dataset where an instance is represented by a bag and each bag can have different number of instances. A bag here represents a week and instances represent activities. These 3 datasets are described concisely in table 1.

Dataset number	Type of dataset	Explanation of dimensions in dataset	Difference between datasets	ML methods considered for dataset	Target data considered
1	2-D	First Dimension signifies aggregated data for a week while second dimension signifies features	This dataset is simple tabular data format significantly different than dataset 2 and 3.	Regression methods	Count of VOs
2	3-D	First dimension signifies previous time step condition, second dimension signifies activities in any week and third dimension signifies features for every activity in a week.	The difference between this dataset from dataset 1 is that this dataset does not aggregate values and maintains most of information about activities in a week.	Regression and classification methods	Count of VOs for main RQ and earned quantity for validation. For classification task presence of VO and absence of VO in a week is modelled.
3	3-D	First dimension signifies a bag, where each bag contains activities in second dimension and features in third dimension.	This dataset differs from dataset 2 in the sense that every bag contains different number of activities however, in dataset 2 the missing activities are padded with 0 values to make it a cube like dataset.	Multi-instance learning (MIL) Classification method	Presence of VO in a week is considered positive class and absence id considered negative class.

Table 1: Description of three datasets generated

3.3.2 Target data creation

The activities saved after application of filters are used to construct the target dataframe. From the PM software change register table is imported in pandas. The resources baseline table is read in pandas dataframe and duplicates are removed where for a single activity and resource a change order might be duplicated. This dataframe is joined with change register to extract issue dates for a change order. The saved activity sequences are read and joined with an inner join with this dataframe to get only relevant change orders. The data for change orders is the aggregated by counting how many activities had change orders on a given issue date of a change order.

This count data is joined with the filtered_df_weekly such that the issue date falls in between the period start and cut-off date. This gives us the count of change orders that were generated in that week. The filtered_df_weekly now includes the target variable as a column. Missing values in the dataframe are filled with zero values.

Due to change order count variable as target, the dataset target column contains large number of zeros and large peaks in certain weeks. It is established in project management practice to not issue change orders near the completion of a project and therefore last 32 weeks of data is discarded. The shape of weekly dataset is thus reduced to 147 x 529.

3.3.2.1 Switching target variable to check validity of data

The target variable is switched to perform validation of data generated. The target is selected to be earned quantity for next week, previous research on project management applications of machine learning have been focused on prediction of earned value but due to lack of cost information earned quantity is selected as target. This step is done according to the RQ put forward so that the results of other RQs are reliable. To change target variable earned quantity column was named target and it was shifted one week before the actual occurrence. In both dataset 1 and 2 these targets are then used.

3.4 Feature extraction

The PM software have reporting functionality that allows the users to extract several features such as baseline planned quantities, estimate at completion, estimate to complete, expended quantity, original plan quantity, remaining work, periodic scheduling factor, schedule variance, to complete performance index and earned quantity. These features are extracted and merged with 'filtered_df_weekly' since the extracted features are on weekly level.

Some of the features were calculated for the data such as progressing activities. Progressing activities were found by counting how many activities had actually started and subtracting count of activities that are finished in the respective week. Time series features are modelled as sine and cosine of their day, week, and month. The transformation using cos and sine is done to capture cyclic nature of time features. For example, day number 30 should be closer to day 2. The sine and cosine components are extracted to capture this relationship (Kazemi et al., 2019). Dataset is again filtered to exclude any weeks where there are no progressing activities.

Due to large number of zeros and peaks in target variable, logarithmic transformation is applied to the target variable. Since simple log becomes infinity at zero therefore a constant number such as 1 is added to avoid infinite values in target variable. The distribution of target variable before and after transformation is shown in figure 5. By taking log transform and adding constant of one the count of zero values in target is preserved.

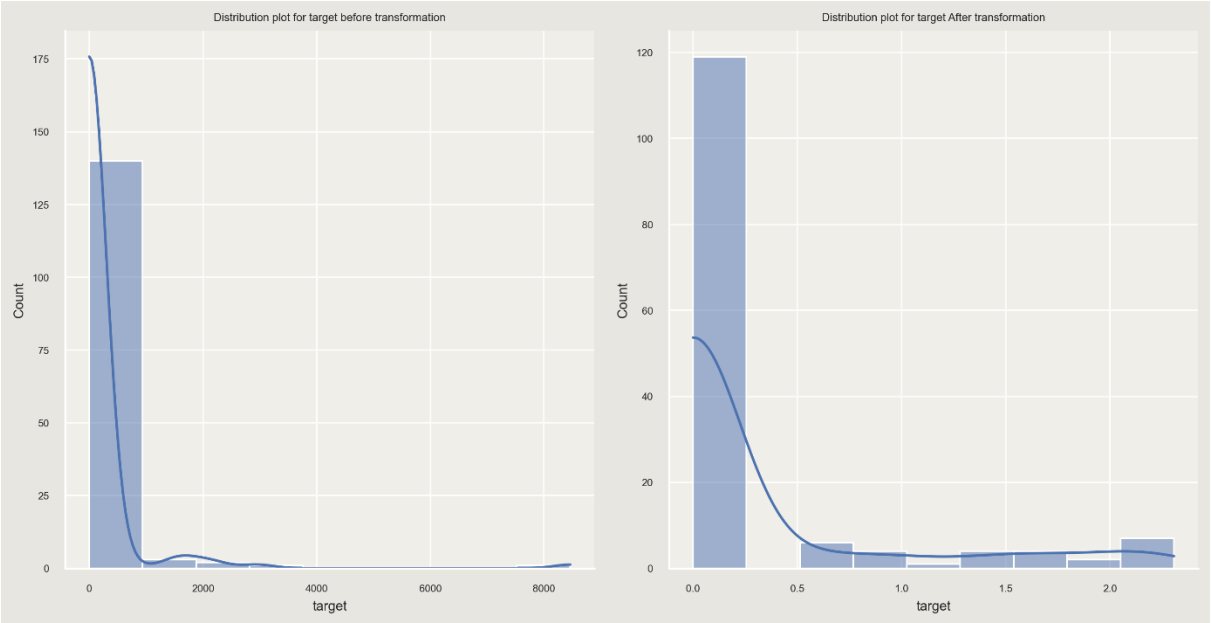


Figure 5: Distribution of target variable before and after transformation

Other features were calculated in the dataset such as actual quantity change over a week, contractual quantity change, total float change, free float change, change in activities started, change in activities finished, count of resources changed, duration of activities change, hours per day, remaining hours for activities in a week, earned hours, performance factor. Change in estimate to complete, change in remaining work, change in schedule variance, and change in earned quantity. These features will later be assessed for usefulness when we remove multi collinearity. The shape of dataframe after feature extraction for weekly dataframe is 147 x 578.

3.5 Data Pre-processing

Data pre-processing step involves removing duplicate columns, removing multi collinearity and removal of high correlation factors. Total 166 columns in weekly dataset are found to be duplications and are thus removed. Columns holding count values are also removed which only signifies how many values of a feature are present in a week. It is then deemed necessary to remove columns that contain only one unique value, this column does not add any information to the dataset since it remains constant. Columns holding dates information and strings are also dropped. These steps reduce the shape of weekly dataset to 147 x 324.

Variational inflation factor (García et al., 2015) is used as a measure of collinearity in dataset. Before the application of variance inflation factor (VIF), Pearson correlation (Benesty et al., 2009) is calculated between every column of dataset excluding the target variable. To apply VIF target column is dropped and VIF values are calculated to each of the features. A threshold of 10 is set for VIF factor and any column having more than 10 VIF value is dropped. These steps reduced the number of features from 324 to 57 features excluding target variable.

3.6 Regression modelling

3.6.1 Preprosessing the data

Dataset 1 is used for regression, dataset is preprocessed to make it compatible with assumptions of regressions models and distribute data between training and validation sets. A training set is only used in the training process and the data must not leak between training and testing sets (Samala et al., 2020). Since the regression data comes in two different datasets that were prepared, one with weekly aggregated values and the other shaped as a matrix therefore, it is necessary that both datasets are split similarly to avoid any data leakage. The split of data is done such that last 20 percent of data is used as testing data, the rest of 80 percent is used in validation and training data. Scikit-learn commonly referred to as sklearn (Pedregosa et al., 2011) is used to split to the data and standard scaler module of sklearn is used to scale the data. Scaling of data helps algorithms converge faster (Bhanja & Das, 2018; Singh & Singh, 2020). Since the models will be developed for LSTM as well a custom function is made to make data of acceptable shape for an LSTM. For LSTM 3 previous time stamps data is given as input.

Dataset 2 is also used for regression but can also be used for classification in which the format of data is number of weeks x activities x features thus making it a 3-D data and suitable for CNN models. The following logic was used to transform the 'filtered_df' which was of shape 188731 x 122. First separation is made for each week and each week data is extracted into a separate dataframe. Then maximum number of activities was identified for different weeks and maximum 2951 activities were found in any week. Thus, an assumption was made that each week will have maximum of 3000 activities and if any week has a smaller number of activities than 3000 then zeros are inserted for these activities. Any feature that had one unique value was dropped. Similar to the dataset 1 last 32 weeks of data was removed since no change order was issued near the completion of project. To make this dataset's results comparable to dataset only those weeks of data was taken that was used in training, validation and test set of dataset 1. After these steps the shape of the dataset was 147 x 3000 x 83 where 147 is number of weeks, 3000 are activities and 83 are features to be used in models. To capture the temporal features of the dataset it was further reshaped to include last 3 weeks of data this step reduces the number of weeks to 144 since first 3 weeks are used in the first sample of data. The resulting data shape is now 144 x 3 x 3000 x 83. The data is split into training, validation, and test sets where 15 percent of data is used in testing, 15 percent for validation and the rest for training. Special consideration has been given to the distribution of data here since it contains large number of zeros therefore, it is necessary that validation and test sets both contain some positive numbers to better capture the model performance.

3.6.2 Models

Several different models were built and tested. For hyperparameter tuning using hyperopt library for python. First model that was tested and optimized is LGBM for which several hyperparameters were optimized including learning rate, maximum depth of model, number of leaves, sub sample, regularization parameters and minimum child samples. Next model optimized are extra trees model with tuning of number of estimators, minimum samples split and maximum depth. Gradient boosting regressor was also optimized for learning rate and number of estimators. Ada boost regressor was optimized next. Another model called cat boost regressor was optimized and feature statistics were plotted, and feature importance were analyzed to get a sense of

understanding for occurrence of change orders. Feature interactions were also analyzed which explains how two different features in a dataset interacts to help predict count of change orders. LSTM model was developed next with hyperparameter tuning of number of LSTM cells, number of LSTM and dense layers and depth of network.

Dataset 2 were tested with two different model architectures. Both models used CNN layers, LSTM layers and dense layers. The architecture of CNN and LSTM model 1 is shown in the figure 6. The input dimension of this model is 3x3000x84 signifying 3 timesteps, 3000 activities and 84 feature for every activity. The first layer has 32 filters applied, second layer has 64 filters followed by batch normalization and maximum pooling layer. The rest of shapes for layers are visible in the figure. Figure 7 shows the model architecture for model 2. The input has same dimension as model 1 while number of filters in first layer is 256, second CNN layer has 512 filters followed by batch normalization and maximum pooling layer. A bidirectional layer of LSTM is used before self-attention layer and the output of self-attention layer is fed into regular SLTM layer before dense layers. Model 2 for dataset 3 contained a special layer called self-attention layer (Zang et al., 2021)The self-attention mechanism for LSTM makes the model more focused on key parts of the sequences relevant to output and creates a filtration effect on remaining inputs (Zang et al., 2021). This self-attention allows interaction of inputs between themselves and thus improves the ability of features to express data. Due to large number of zeros a custom loss function was developed which gave more penalty to the model weights if a prediction of positive number were worse than a prediction of zero valued target.

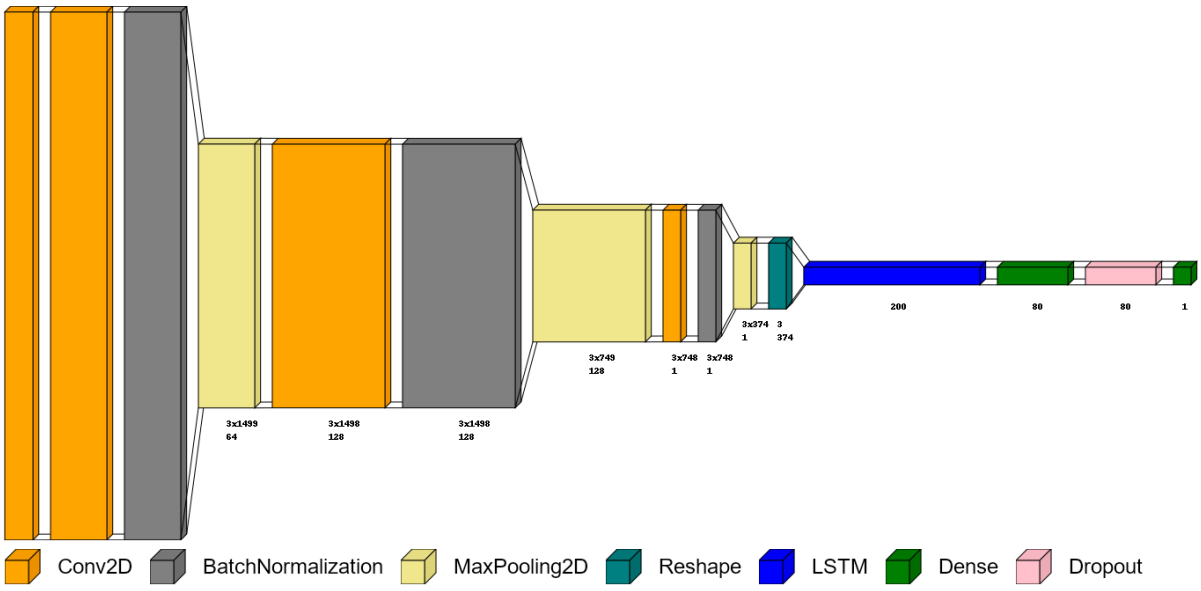


Figure 6: CNN + LSTM Hybrid model architecture

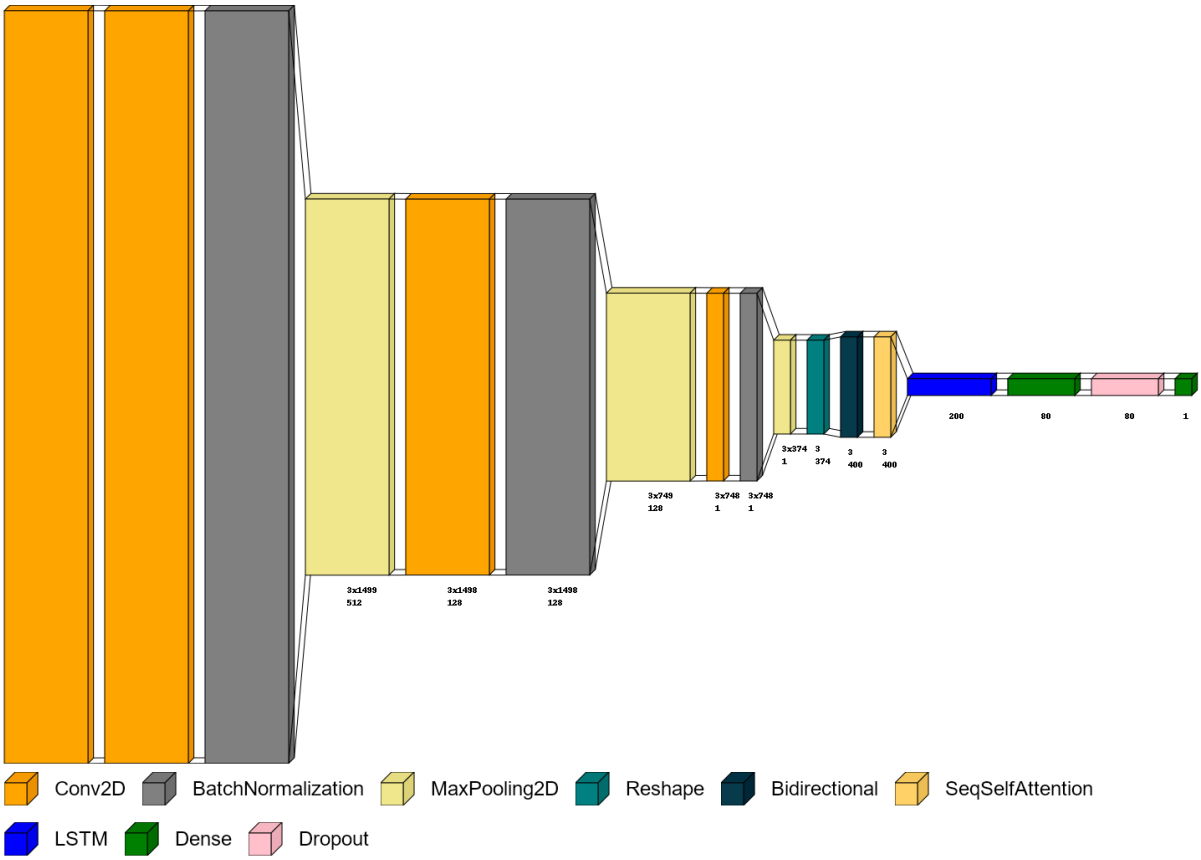


Figure 7: CNN+LSTM+Self-attention model architecture

3.7 Classification modelling

Classification modelling is done using only those ML methods which can predict the probabilities since the RQ stated in the thesis is concerned with the probability of having Vos for next week.

3.7.1 Preprosessing the data

Classification modelling was done using dataset 1 and 3. Dataset 1 was used in classical classification models and target variable was transformed into a classification problem by replacing any row having more than 0 change order to 1 thus making it a binary classification problem. For dataset 3 the target variable was transformed such that 0 values are replaced by -1 and any positive value is replaced by 1 because this dataset will be used for MIL model. MIL models assume a negative class and a positive class to make distinctions between different bags. A bag here represents a week of data which includes data about activities and their features. The advantage of MIL is that in between different bags there could be different number of activities or instances and thus this makes this algorithm a strong candidate for classification for given problem.

3.8 Performance metrics

Performance metrics are used to evaluate some measure of error or accuracy for a task. It measures how the actual values and predictions differ (Botchkarev, 2018). In this report, we'll use mean squared error (MSE) metric to keep track of model performance (Wang & Bovik, 2009). MSE is a common metric for regression tasks and on count data. It is chosen because of its symmetric property. An assumption made while use of this

metric as loss function for optimization is that all the samples are equally important. Due to this assumption a custom metric was also developed since the target for regression tasks contains large number of zeros and more weight should be given to non-negative targets. A weight of extra 10% was assigned to the targets where actual target was not 0. For validation task mean absolute percentage error was chosen as performance metric for comparison with past results on same metric. The formula for custom metric used in optimization of model 2 on dataset 2 is as follows:

$$loss = (y_{pred} - y_{true})^2 \; if \; y_{true} = 0 \; else \; 1.1 \; X \; (y_{pred} - y_{true})^2$$

For classification tasks, the target have imbalanced classes therefore regular accuracy metric cannot be used to assess the quality of models (Lipton et al., 2014). F1 score is used as a metric to keep track of classification models performance.

3.9 Reliability of results

Reliability is important for generalization of methods across different situations (Field, 2013). In the research method proposed by (Field, 2013), the steps included are to generate research questions, identify and study theoretical concepts, generate hypothesis, generation of predictions and analysis of results. Validity ensures that the process employed measures what it was expected to measure, and reliability then ensures that the results can be generalized and reproduced. Any process or method can only be valid if it is a reliable one(Field, 2013). A concept within reliability is test-retest which means to measure same results under same conditions for same set of population samples. In AI and ML this test-retest reliability can be tested by setting a seed value for functions being approximated and measuring the output from methods. During the development of models listed in this thesis, seed values were given for every process to ensure reproducibility of results. However, if a process evolves over time and values change different statistical tests can be employed to test reliability but in case of this study and proposed RQs the output variables doe does not change in time, which means that the number of VOs or earned quantity in a week will be fixed when measured today or tomorrow for a week. In AI and ML literature reliability is also measured by the ability of models to explain its behaviour (Tjoa & Guan, 2020) which will also be discussed and explained in this thesis which further increases the trust and confidence in the methods developed.

4 Results

4.1 Results of Regression

The baseline is set by measuring the mean of training examples and predicting this value for training, validation and test sets. The mean comes out to be 0.02327 for logarithmic transformed and scaled target. The errors for baseline model on training, validation and test set is shown in table 2. The evaluated metric is mean squared error.

Set evaluated	Training	Validation	Test
MSE	1.0224	0.7857	0.4690

Table 2: Baseline mean squared error for regression tasks

These values are used as a reference to measure the improvements of different models over this baseline model.

4.1.1 Dataset 1

LSTM model was optimized for hyperparameters and architecture of the model. The best configuration was found to be a single LSTM layer with 400 cells and connected to dense layer with 100 neurons and a final layer of 1 cell with a linear activation. Table 3 highlights the MSE errors on training, validation and testing sets.

Set evaluated	Training	Validation	Test
MSE	0.1351	0.7832	0.7399

Table 3: LSTM Model MSE on dataset 1

The results were plotted for on a graph to check the fit for training data and to compare test results. Testing and training data is compared graphically and is shown in the figure 8 and 9. In all of the results plotted in this section x-axis is named index which signifies the number of weeks in respective data and y-axis signifies predicted and actual values of transformed number of variation orders.

Figure 8: LSTM Model actual and predicted values on training data (x-axis named index signifies the number of week in training data and y-axis signifies predicted and actual values)

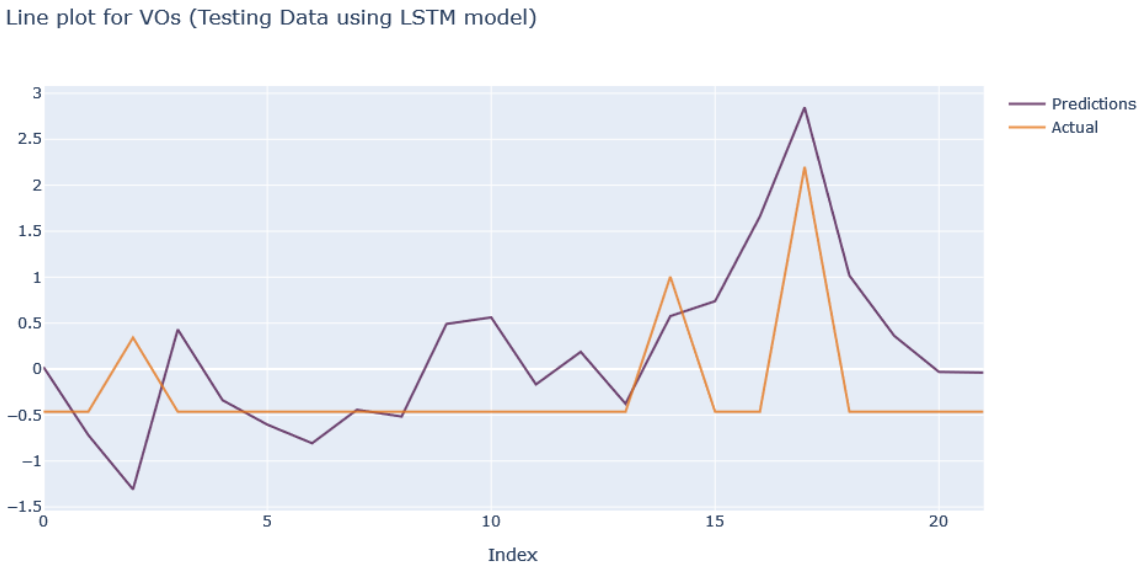


Figure 9: LSTM Model actual and predicted values on testing data

LGBM model was optimized for hyperparameters using unscaled data since LGBM is not dependent on scaling of data. Total of 100 optimization rounds were done and on each round lowest loss on training set was stored. The best model was extracted by using the saved parameters of these evaluation rounds. Table 4 lists MSE of LGBM model on training, validation, and test sets.

Set evaluated	Training	Validation	Test
MSE	0.0371	0.7603	0.7592

Table 4: LGBM Model MSE on dataset 1

These results are compared graphically for training and test sets in the figure 10 and 11.

Figure 10: LGBM Model actual and predicted values on training data

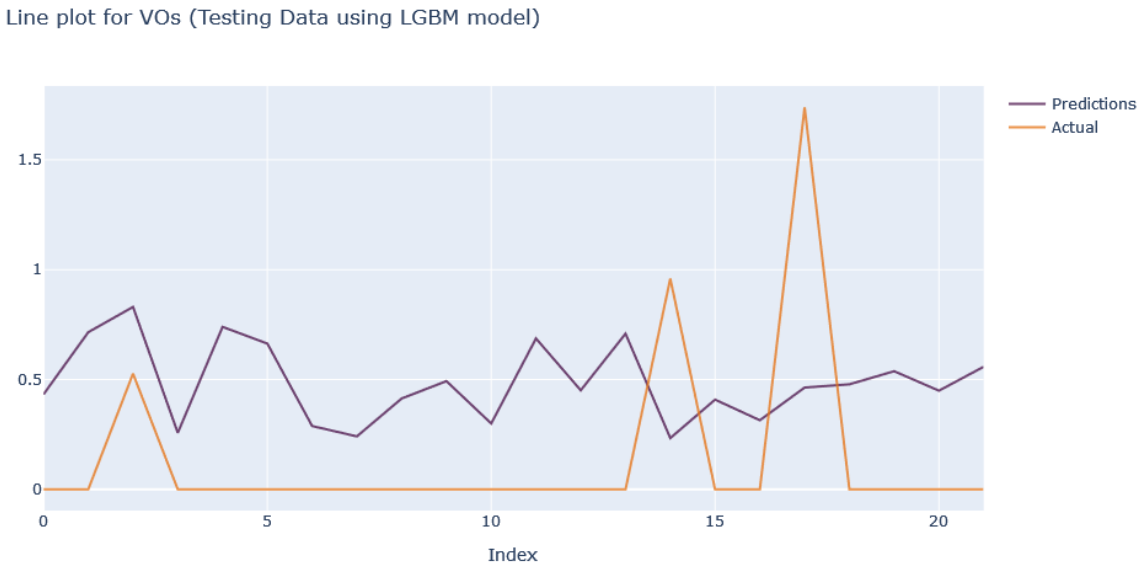


Figure 11: LGBM Model actual and predicted values on testing data

Extra trees regressor hyperparameters were optimized using 20 iterations where the objective is to minimize validation error. The resulting errors on training, validation and test datasets are listed in the table 5.

Set evaluated	Training	Validation	Test
MSE	0.0000	0.7178	0.6143

Table 5: Extra trees regressor model MSE on dataset 1

The data perfectly fits the training data and therefore the comparison on a graph for training data is not shown in the thesis. A graph was plotted for testing data to see the fit visually which is shown in figure 12.

Figure 12: Extra trees Model actual and predicted values on testing data

Gradient boosting regressor (GBR) results using same evaluation criteria is listed in the table 6. The graphical comparison for testing and training data is shown in figure 13 and 14.

Set evaluated	Training	Validation	Test
MSE	0.0933	0.7836	1.3289

Table 6: GBR Model MSE on dataset 1

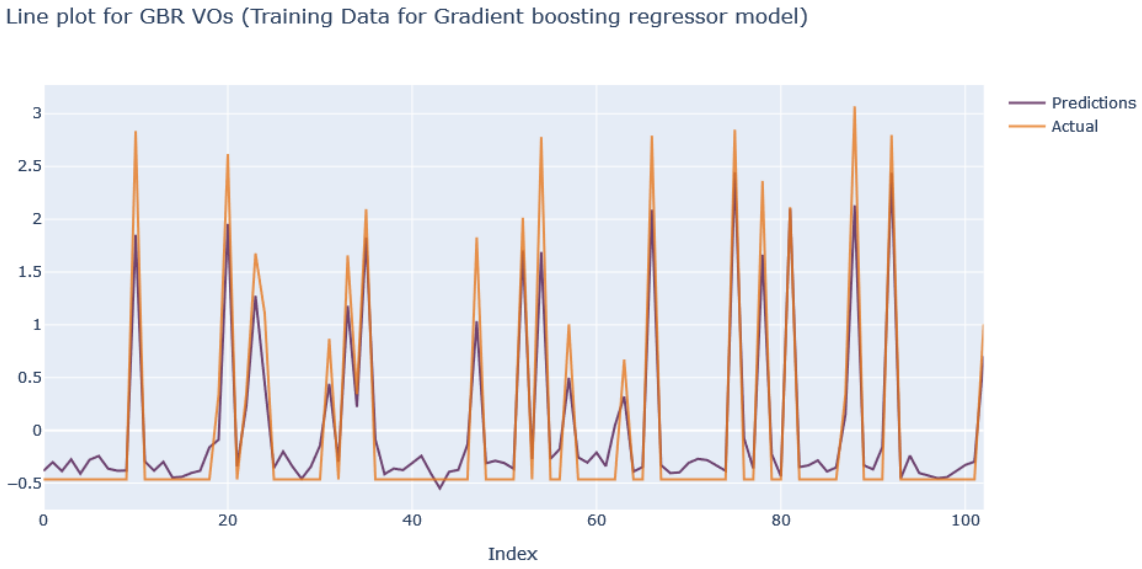


Figure 13: GBR model actual and predicted values on training data

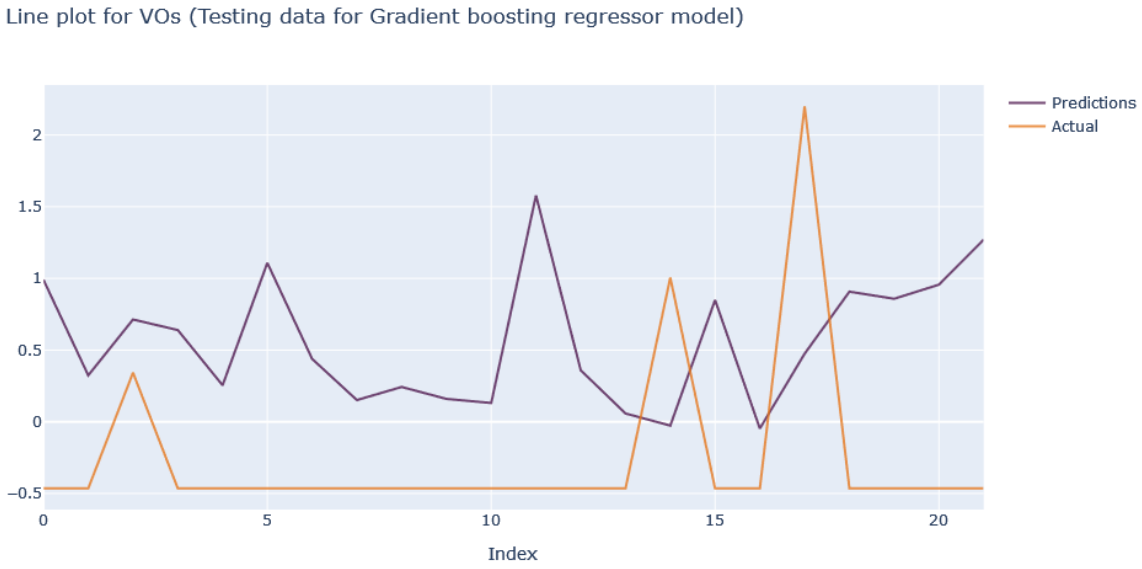


Figure 14: GBR model actual and predicted values on testing data

Ada boost regressor (ABR) was trained and hyperparameters optimized for lowest mean squared error on validation data. The following table highlights results obtained after hyperparameters tuning.

Set evaluated	Training	Validation	Test
MSE	0.7508	0.6850	0.4028

Table 7: ABR model MSE on dataset 1

These results were plotted on a figure for training and testing data and are shown in figure 15 and 16.

Figure 15: ABR model actual and predicted values on training data

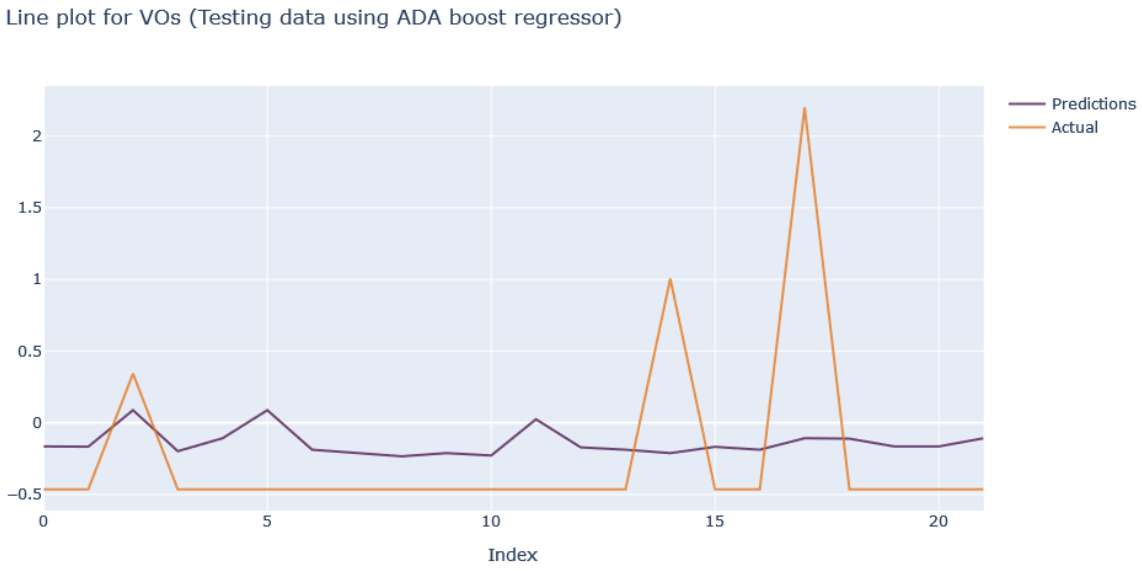


Figure 16: ABR model actual and predicted values on testing data

CAT Boost regressor was optimized for training loss using mean absolute error on training set and evaluating the fit on validation data. The total iterations done for hyperparameter optimization were 20. The results of mean squared error on training, validation and test set are shown in the table 8.

Set evaluated	Training	Validation	Test
MSE	0.0035	0.8721	0.4003

Table 8: CATBoost regressor model MSE on dataset 1

Since the training data is nearly perfectly fitted therefore, the testing data was plotted and is shown in the figure 17.

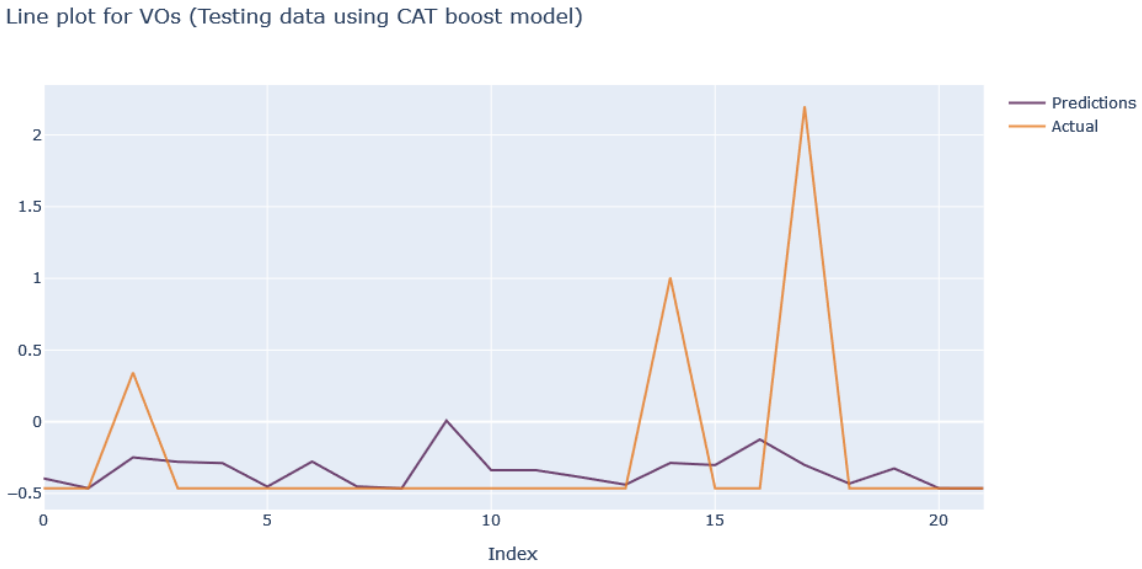


Figure 17: CATBoost model actual and predicted values on testing data

4.1.2 Dataset 2

Dataset 2 is of shape (sample, time steps, instances, features) where sample is number of weeks for which a predictions are made, time steps are 3 previous weeks of data, instances are activities in the respective weeks for time steps and features are features for every week tie steps. This is split into training, validation, and test sets such that no data leakage is possible between these sets. Two different models were developed and their hyperparameters optimized using grid search over a specific range of values. The last dense layer had both 'softplus' as their activation function. This activation function has property of output which gives values only above 0. Softplus activation function is used since in target values no value is less than zero and the models are trained on unscaled data.

Model 1 is structured to extract spatial features with help of CNN followed by an LSTM layer to capture the temporal features of data. CNN layers filters were only applied to spatial domain of data and no pooling or kernels were applied to temporal domain of data. For this model maximum pooling layers were used to extract only relevant features and to decrease model complexity and make it less prone to overfitting the data. The architecture of Model 1 is shown in the figure 6.

Model 1 mean squared errors on training, validation and testing sets are listed in the table below.

Set evaluated	Training	Validation	Test
MSE	0.1865	0.9309	0.2823

Table 9: CNN+LSTM model MSE on dataset 2

Model performance on training and test sets are visualized graphically and is shown in figure 18 and 19.

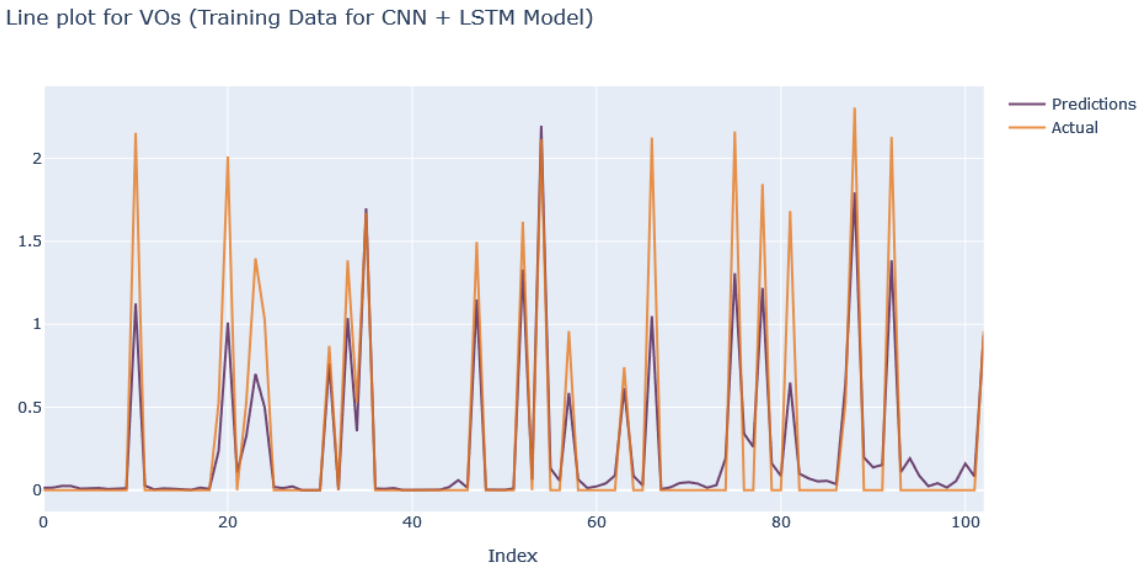


Figure 18: CNN+LSTM model actual and predicted values on training data of dataset 2

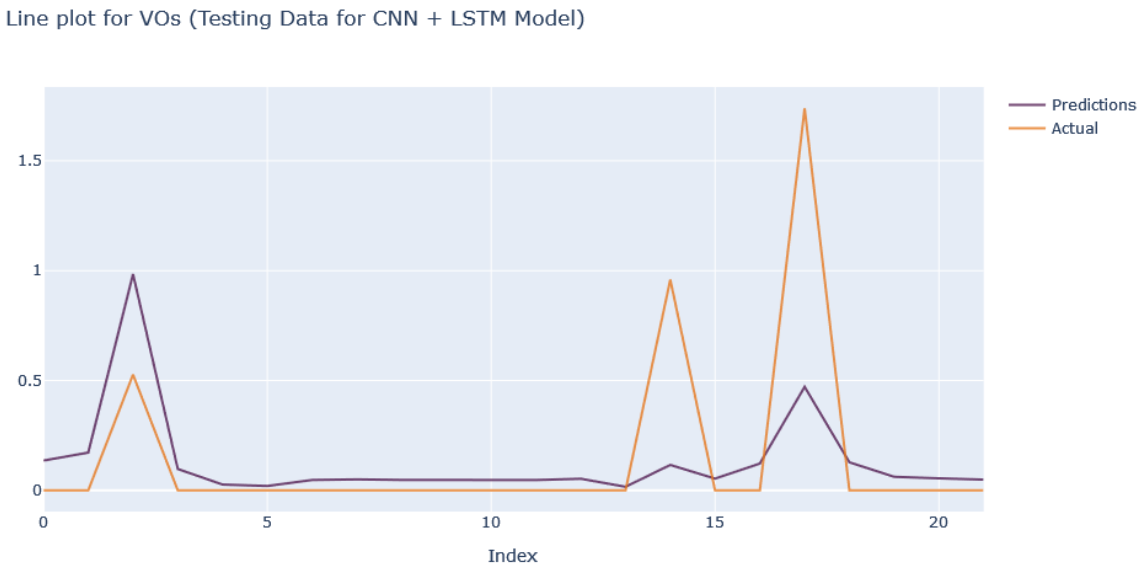


Figure 19: CNN+LSTM model actual and predicted values on testing data of dataset 2

Model 2 is structured to extract similar spatial features followed by a bi-directional LSTM layer, a self-attention layer and another LSTM layer before dense layers. CNN layers are applied with more depth and therefore, can extract more rich features from the data. The self-attention layer is explained in detail in the paper by (Zang et al., 2021). A custom loss function was also developed for this model to enhance its performance on zero-inflated data as explained in section 3.6.2. The model architecture is shown in figure 7.

The performance metrics from this model is listed in table 10 and predictions are compared with actuals in figure 20 and 21.

Set evaluated	Training	Validation	Test
MSE	0.5178	0.9619	0.4583

Table 10: CNN+LSTM model with self-attention layer MSE on dataset 2

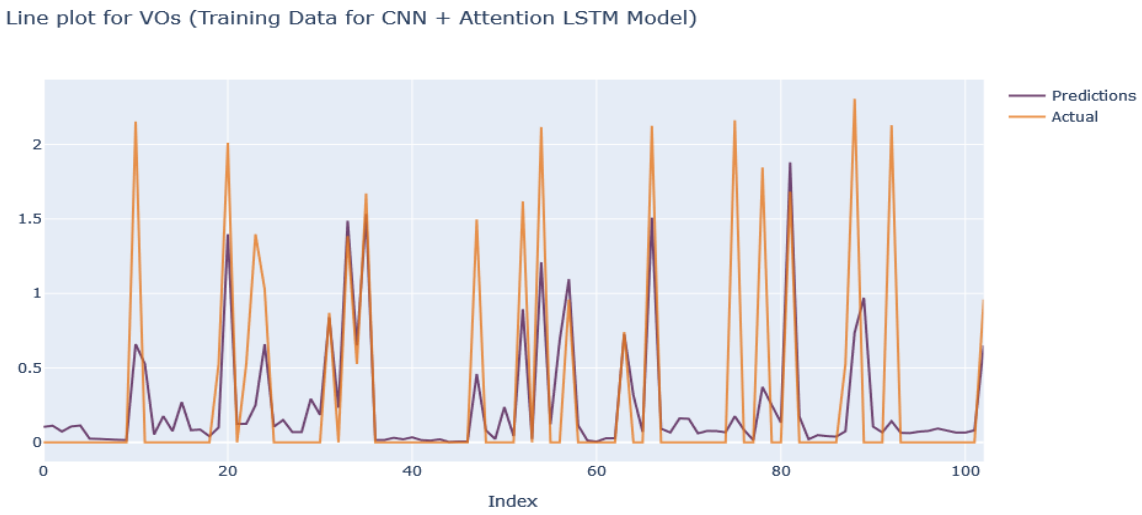


Figure 20: CNN+LSTM+self-attention layer model actual and predicted values on training data of dataset 2

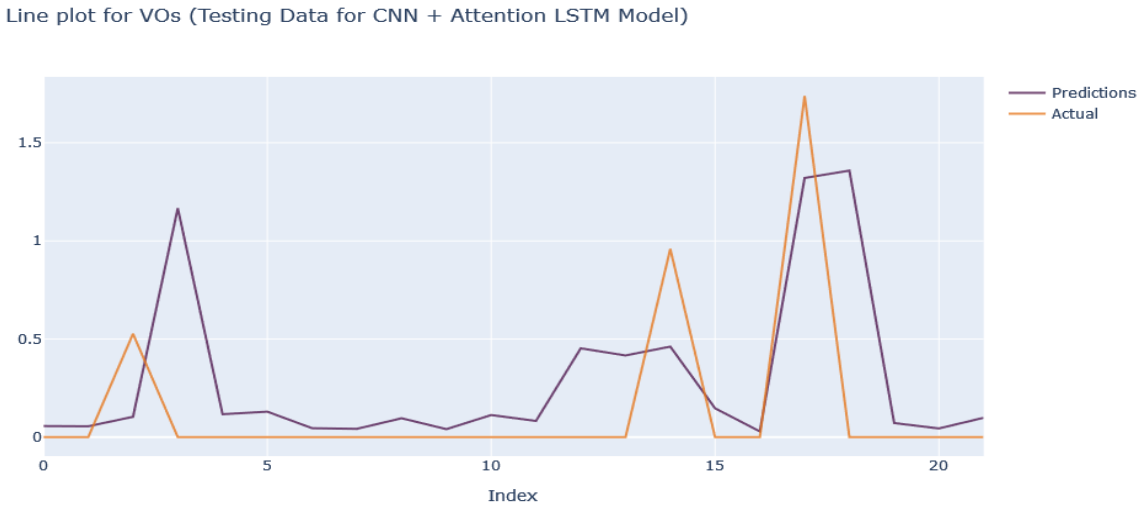


Figure 21: CNN+LSTM+self-attention layer model actual and predicted values on testing data of dataset 2

4.2 Results of classification

4.2.1 Dataset 2

A baseline was set for classification for comparison of models. The baseline score was calculated using F1 measure when every prediction is 1, that is, a baseline predicts every week as having a VO. These scores are listed in table 11.

Set evaluated	Training	Validation	Test
Baseline F1 score	0.3137	0.2400	0.1999

Table 11: Baseline F1 score for classification tasks

Figure 22 and 23 shows how classifier behaved while classification task in which the 1 value represents that next week will have a VO and zero represents no VO. The performance metric for training, validation and test set are listed in table 12.

Set evaluated	Training	Validation	Test
F1 score	0.8235	1.0	0.8888

Table 12: F1 score for CNN+LSTM model on dataset 2

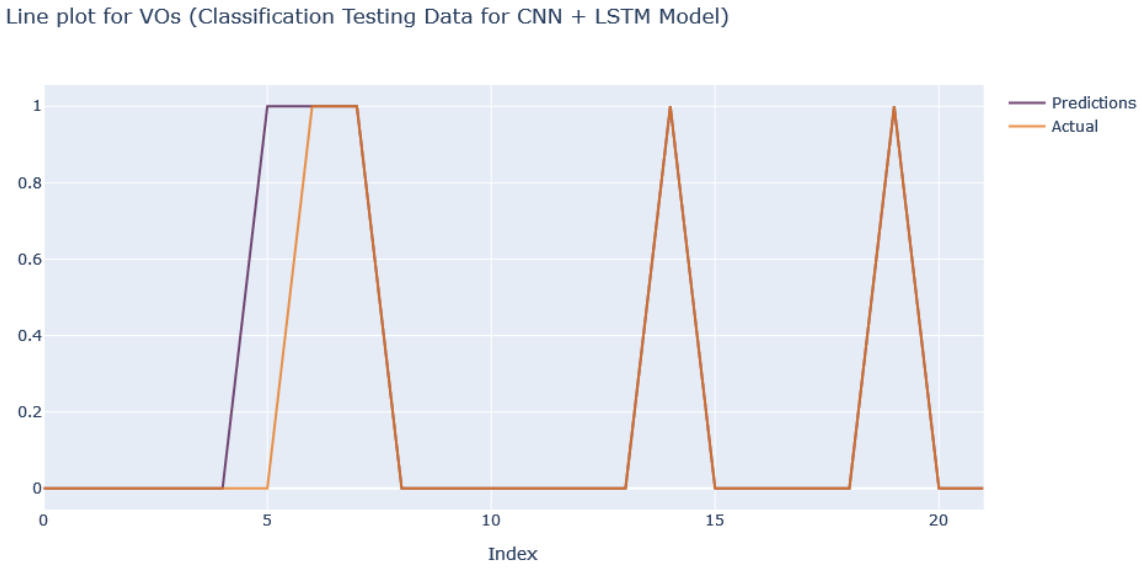


Figure 22: CNN+LSTM model actual and predicted values for classification on testing data of dataset 2

Figure 23: CNN+LSTM model actual and predicted values for classification on training data of dataset 2

Next hybrid model of CNN-LSTM with self-attention layer was tested for classification and results for F1 score are shown in table 13. While graphically the results and original target values are compared on training and test set in figure 24 and 25.

Set evaluated	Training	Validation	Test
F1 score	0.8571	1.0	0.8888

Table 13: F1 score for CNN+LSTM model with self-attention layer on dataset 2

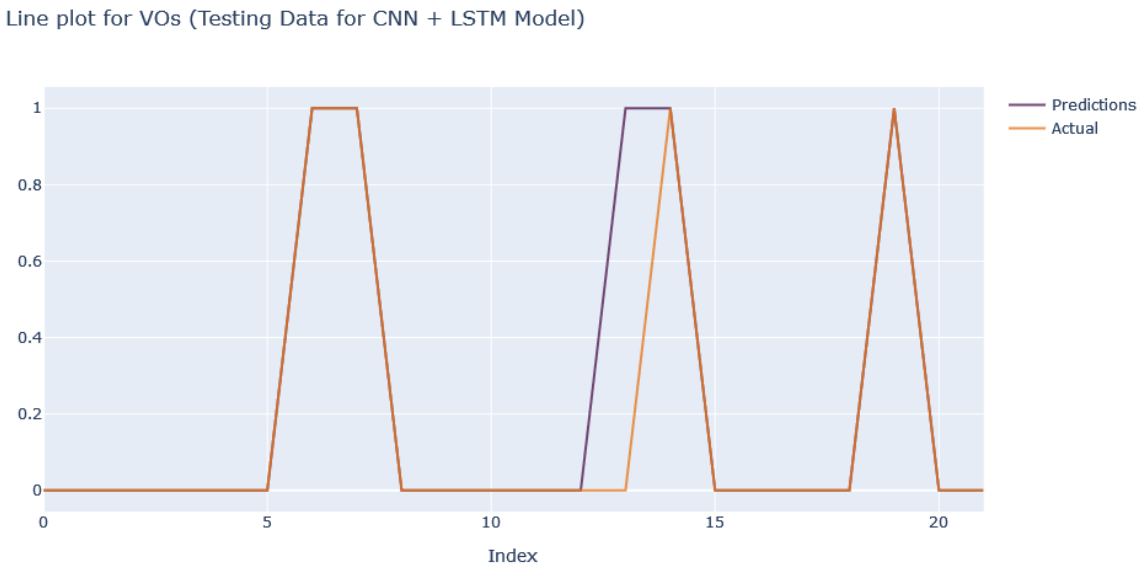


Figure 24: CNN+LSTM+self-attention layer model actual and predicted values for classification on testing data of dataset 2

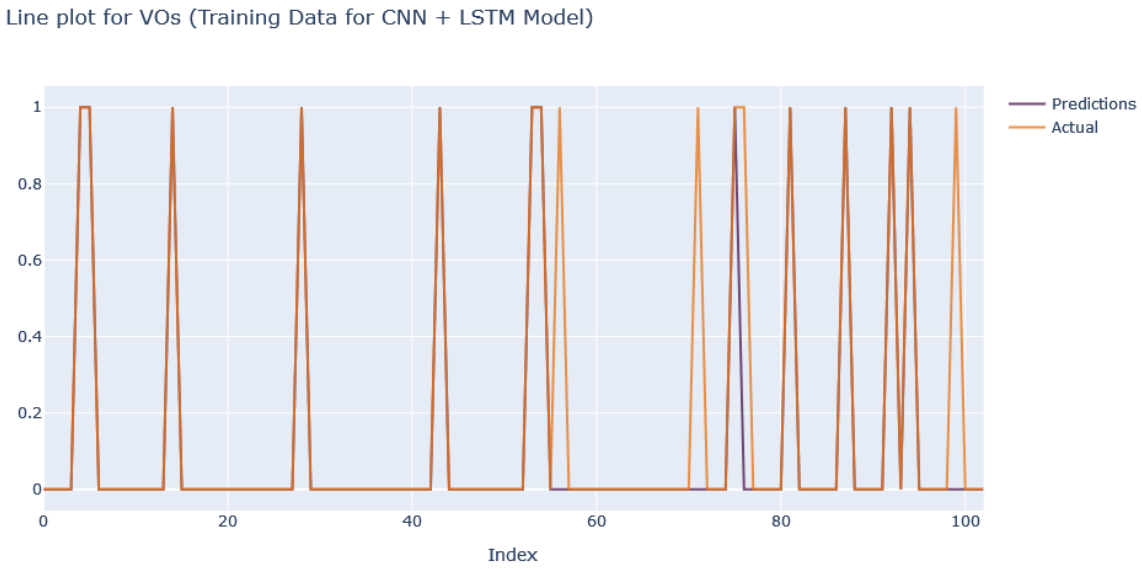


Figure 25: CNN+LSTM+self-attention layer model actual and predicted values for classification on training data of dataset 2

4.2.2 Dataset 3

Multi instance learning algorithms are developed as discussed in theory and method due to special type of data, that is, weekly project data. Table 14 shows F1 score for MIL algorithms tested for dataset 3. The F1 score for MICA MIL algorithm is 0 since it always predicts one class only for every example in train and test set. STK performed slightly better than baseline model.

Set evaluated	Training	Validation	Test
MIL- STK	0.4571	0.3333	0.7272
MIL- MICA	0	0	0

Table 14: MIL F1 score on dataset 3

4.3 Results for validity of data check

For validity check of dataset target variable is changed to earned quantity and models were trained and optimized using training, validation, and test set. This is a regression problem and therefore the best result obtained in regression is referenced and therefore dataset 2 was selected for validation, MAPE performance measure for this dataset is given in table 15 and results are shown graphically in figure 26 and 27. X-axis shows number of weeks in respective dataset while y-axis shows actual and predicted earned quantity values. The CNN and LSTM hybrid model architecture was kept same as the one used in regression models of VO prediction. Dataset 1 was also validated with the same earned quantity prediction performance and compared with the performance on dataset 2 for validating the data generation process. The performance measures for validation on dataset 1 is given in table 15 and the results are plotted graphically in figure 28 and 29. The LSTM model consisted of one LSTM layer with 300 cells followed by 3 dense layers with 100,80 and 1 cell respectively. The input consisted of last 3 weeks of features for LSTM layer.

Model	Training MAPE	Validation MAPE	Testing MAPE

CNN+LSTM (Dataset 3)	0.3337	0.1780	0.5252
LSTM (Dataset 1)	0.2719	0.1651	0.7707

Table 15: Validation results on MAPE measure for dataset 1 and 3

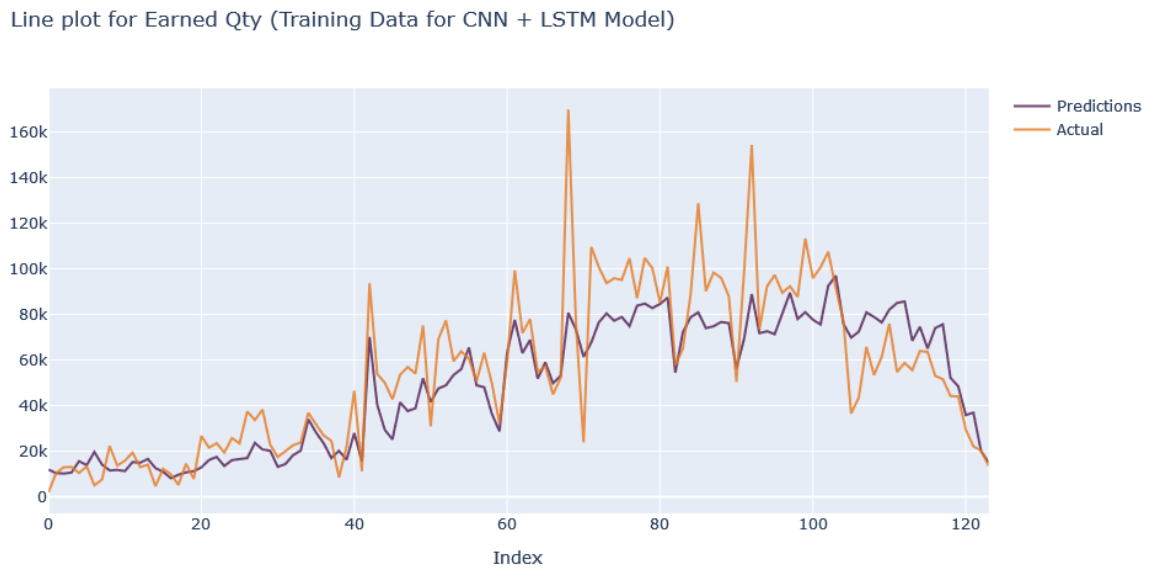


Figure 26: Earned quantity prediction using CNN and LSTM model on dataset 2 (Training data)

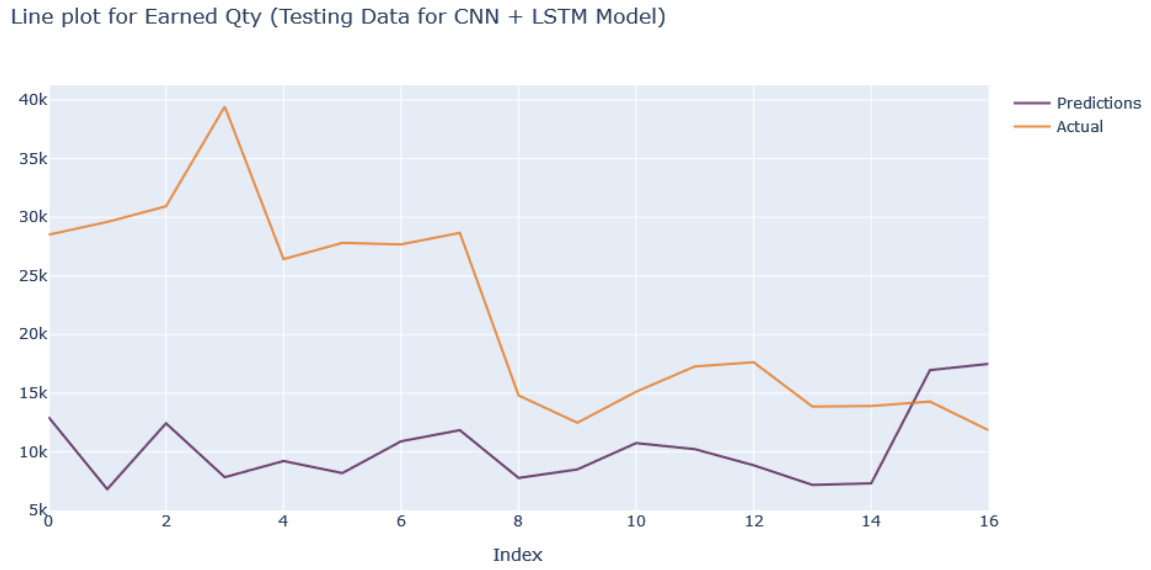


Figure 27: Earned quantity prediction using CNN and LSTM model on dataset 2 (Testing data)

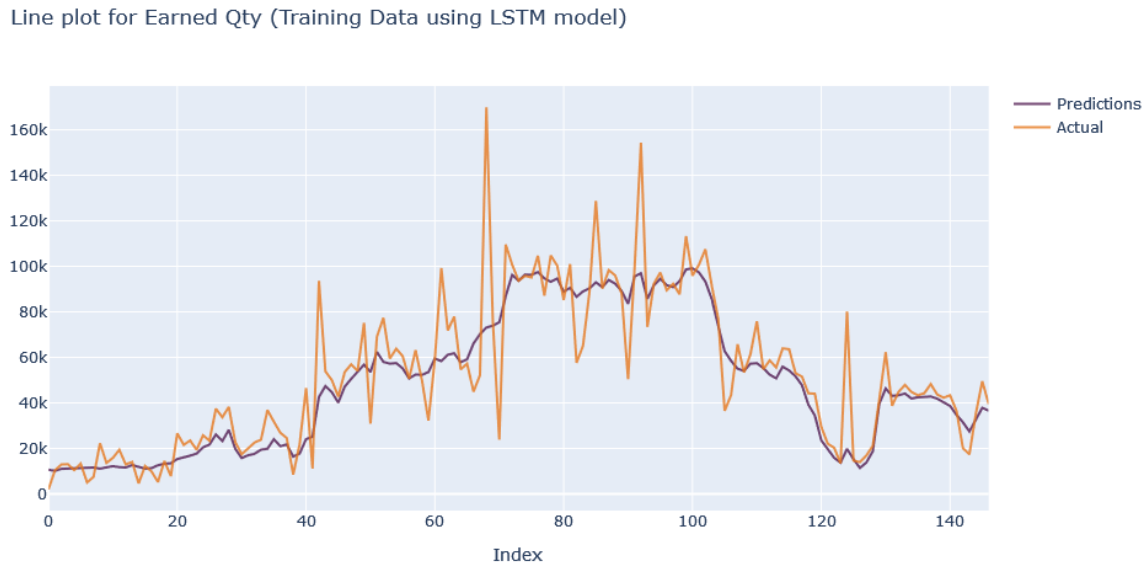


Figure 28: Earned quantity prediction using LSTM model on dataset 1 (Training data)

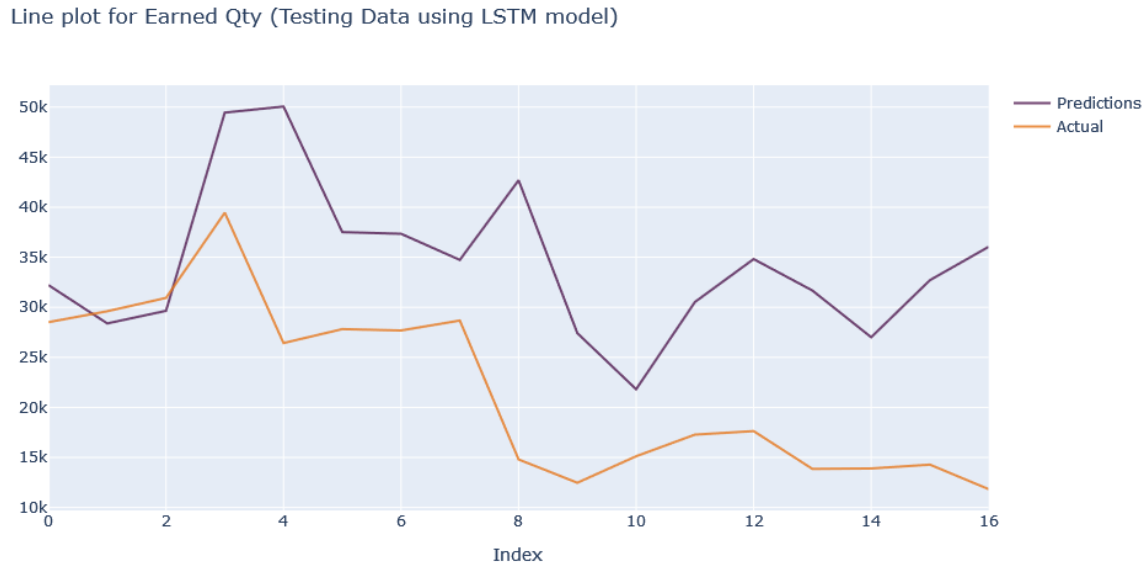


Figure 29: Earned quantity prediction using LSTM model on dataset 1 (Testing data)

4.4 Deep learning explainability

Deep learning model integrated gradient attributions were calculated using "Alibi" (Klaise et al., 2021) library for python. These attributions were divided into 4 distinct parts of project namely, start of project, middle of the project, near end of project and end of project. The end of project is not a literally end of the project, it is named for the test data attributions which are still not end of project since last few weeks of data were removed from dataset due to assumption of no VOs at actual end of project. The start of project lasts for first 45 weeks of dataset, middle of the project is next 35 weeks and

near end of project is 33 weeks after that. The last segment lasts for 22 weeks and consists of test set used for testing of models. Linearity of the model layers are also calculated using the method described in section 2.9 for test set instances. These attributions are calculated using dataset 2 and best model found in results. Dataset 2 consisted of three dimensions with first dimension signifying time steps to capture any time dependencies, second dimension signifies activities sorted by their progress in a week and third dimension contains features for each activity and time step. The dataset is elaborated in section 3.6.1 and explained in table 1.

4.4.1 Attributions at start of project

Attributions at start of project were calculated for each feature of the 84 features and later for 3 time steps. The aggregation of attributions is made by computing sum along certain axis such that the required attributions are additive in nature. These attributions are shown in figure 30 and 31. In the figures showing attributions a color scale is used to demonstrate the values of calculated attributions in which gray scale maps the lowest values to white and highest values to black color. Figure 30 shows attributions for features where x-axis contains feature number starting from zero and y-axis shows week number starting from 0, the attribution values are mapped using a continuous color scaling. Figure 31 shows attributions for time steps on y-axis where 2 is the latest time step, y-axis shows number of weeks starting from 0 with values mapped using a continuous coloring scale to show attributions.

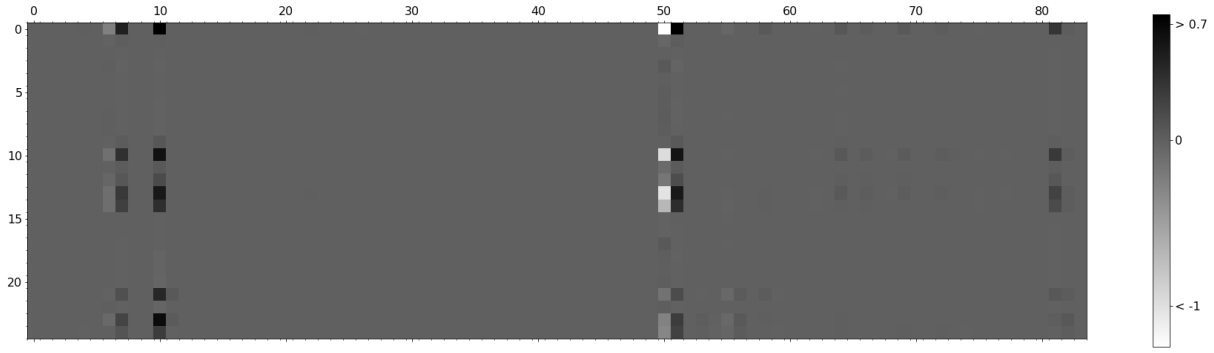


Figure 30: Feature attributions for start of project phase

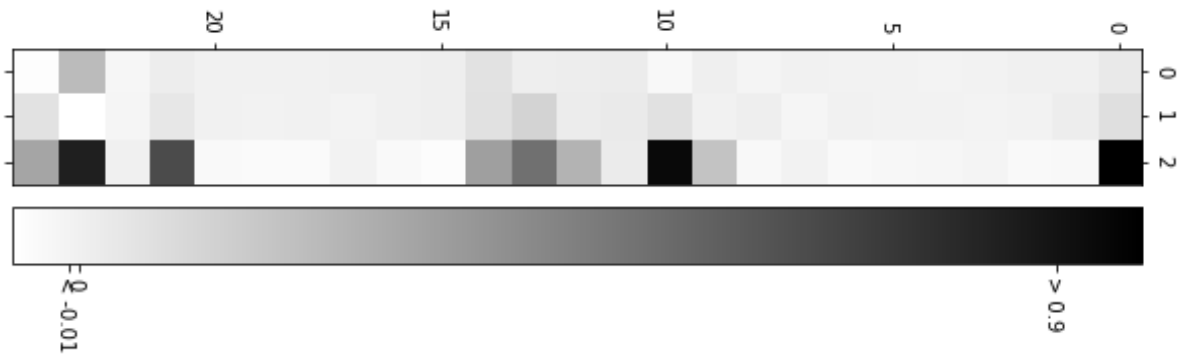


Figure 31: Time attributions for start of project phase

Attributions for the activities in a week are also plotted, which are shown in figure 32 and 33 for first 100 activities and last 120 activities in these weeks. In these figures color scales demonstrate the calculated attributions values where blackish colors signify greater than mean and white color signifies lowest value. In figure 32 first 100 activities are shown in x-axis and weeks are shown on y-axis with attribution values mapped using a continuous coloring scale. Similarly in figure 33 last 120 activities are shown on x-axis while y-axis shows number of weeks starting from zero, the attributions are shown using a continuous gray scale.

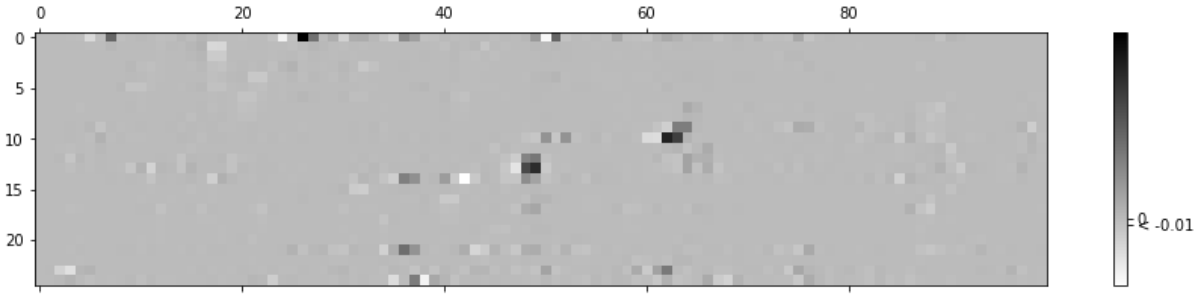


Figure 32: Activity attributions of first 100 activities for start of project phase

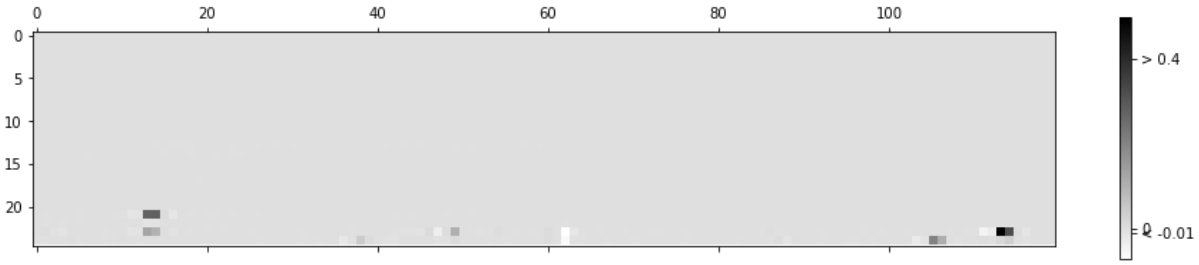


Figure 33: Activity attributions of last 120 activities for start of project phase

4.4.2 Attributions at middle of project

Middle of the project is supposed to be of 35 weeks of length after ending period of start of project. Attributions at middle of project was calculated similar to the above method but the data was for middle of project. These attributions are shown in figure 34 and 35. The color scale is the same as above figures where black color signifies greatest values while white colors mean lowest values in the figure. In figure 34 x-axis contains features and y-axis contains number of weeks in this part of project, the attributions are shown using a continuous gray scale. In figure 35 x-axis signifies number of weeks and y-axis signifies time steps where 2 is the latest time step, the attributions are shown using continuous gray scale.

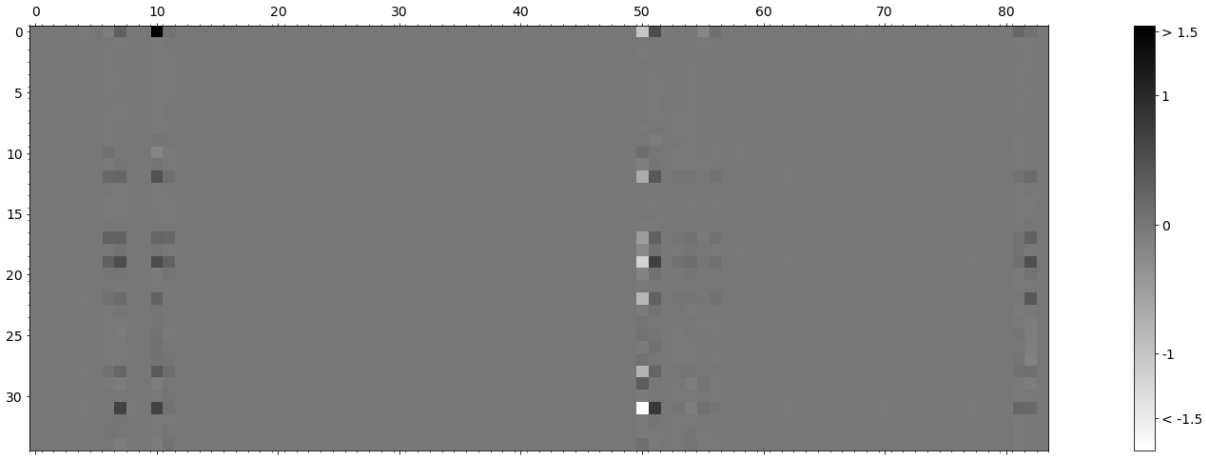


Figure 34: Feature attributions for middle of project phase

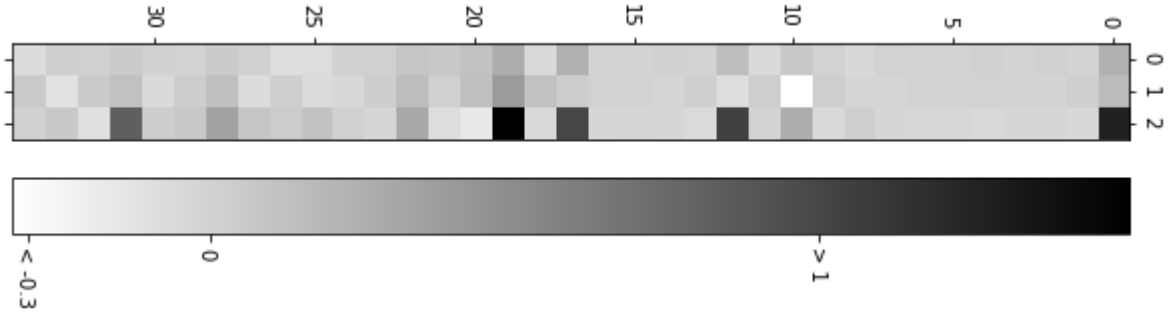


Figure 35: Time attributions for middle of project phase

Attributions for first and last 140 activities in these weeks are also plotted which are shown in figure 36 and 37. The color scheme in these figures are same as the figures above. In figure 36 x-axis shows first 140 activities and y-axis contains number of weeks in this part of project, the attributions are shown using a continuous gray scale.

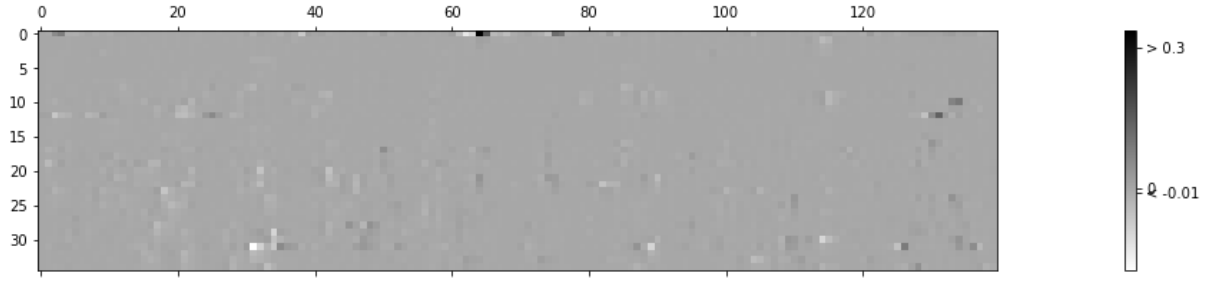


Figure 36: Activity attributions of first 140 activities for middle of project phase

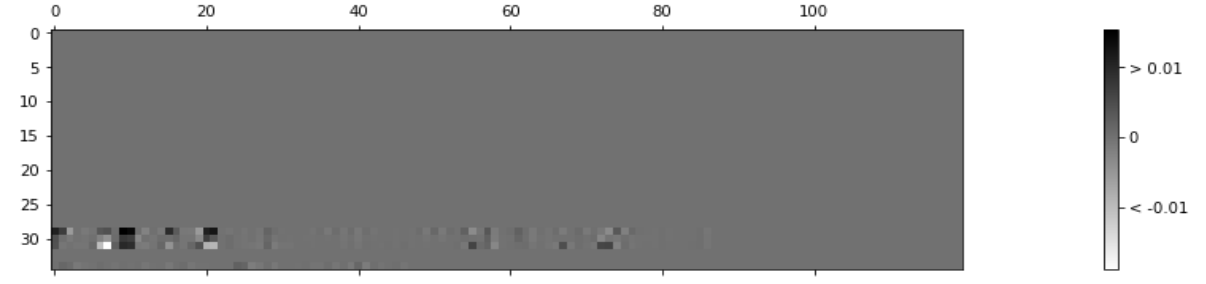


Figure 37: Activity attributions of last 140 activities for middle of project phase

4.4.3 Attributions at near end of project

Near end of the project is supposed to be of 33 weeks of length after ending period of middle of the project. The results for attributions are shown in Figure 38 and 39. In these figures the color scale is the same as figures above where black colors signifies greatest value in figure and white color shows lowest values. In figure 38, x-axis shows features and y-axis shows number of weeks in this part of project, the attributions are shown using a continuous gray scale. In figure 39 x-axis shows number of weeks while y-axis shows time steps where 2 is the latest time step, the attributions are again mapped using a continuous gray scale.

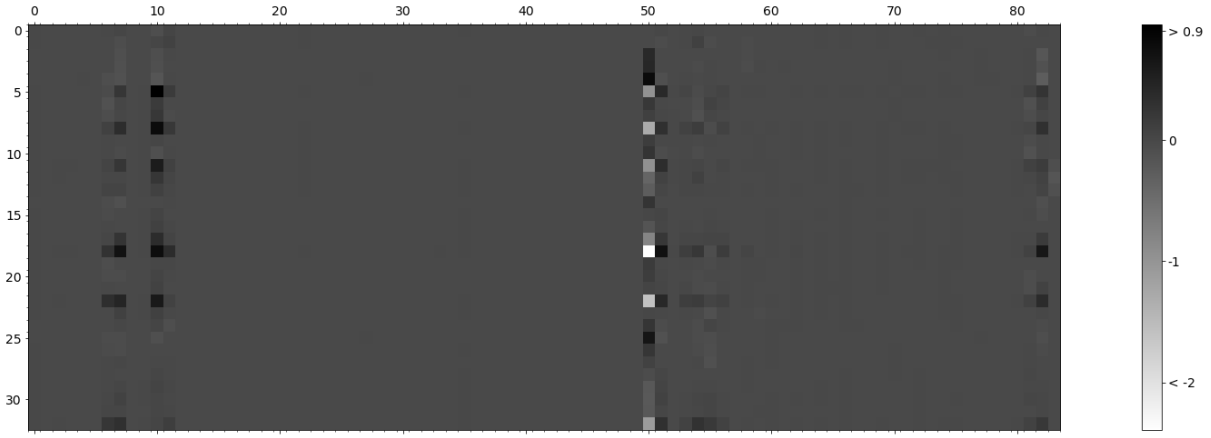


Figure 38: Feature attributions for near end of project phase

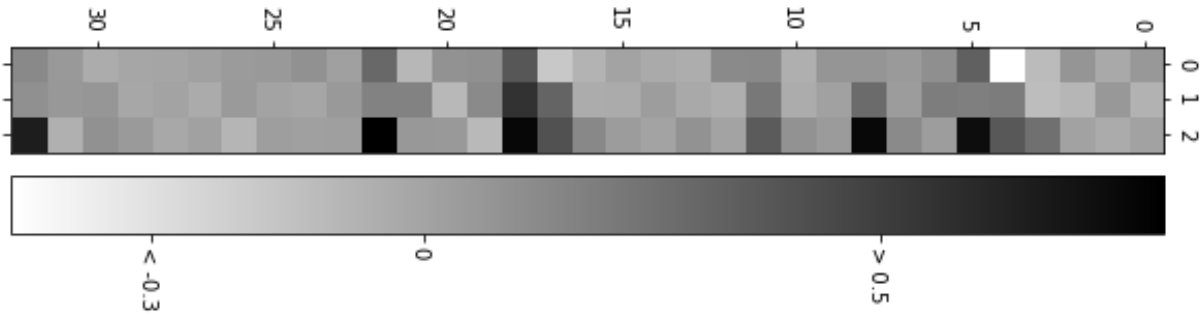


Figure 39: Time attributions for near end of project phase

Activities attributions for first 100 and last 140 activities during these weeks are shown in figure 40 and 41. In figure 40 and 41 x-axis contains activities and y-axis contains number of weeks while attributions are shown using a continuous gray scale.

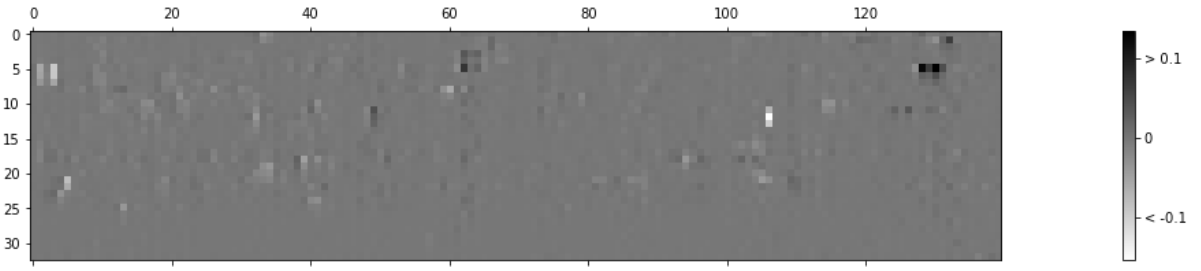


Figure 40: Activity attributions of first 140 activities for near end of project phase

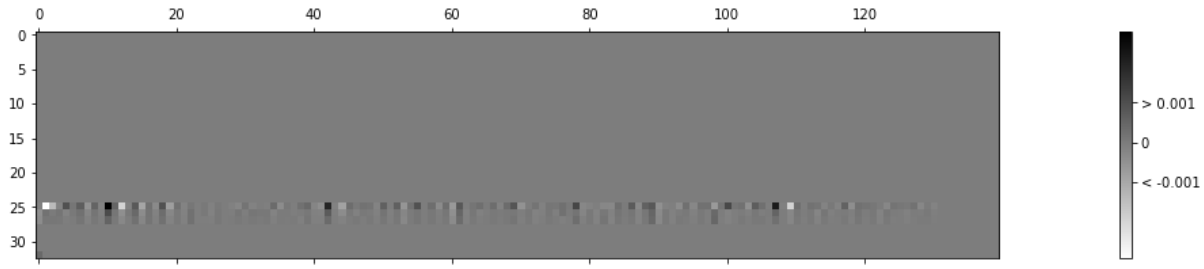


Figure 41: Activity attributions of last 140 activities for near end of project phase

4.4.4 Attributions at end of the project

End of the project is supposed to be of 22 weeks of length after ending period of near end of the project, but it is not literal end of the project. This is supposed to be end of the project since Vos are not issued after certain time as standard practice in PM. The attributions during this period are shown by figure 42 and 43. The color scales used in these and following figures is the same as employed in figures above. In figure 42 x-axis contains number of features and y-axis contains number of weeks while attributions are shown using a continuous gray scale. In figure 43 x-axis shown number of weeks and y-axis shows time steps while attributions are shown using a continuous gray color scheme.

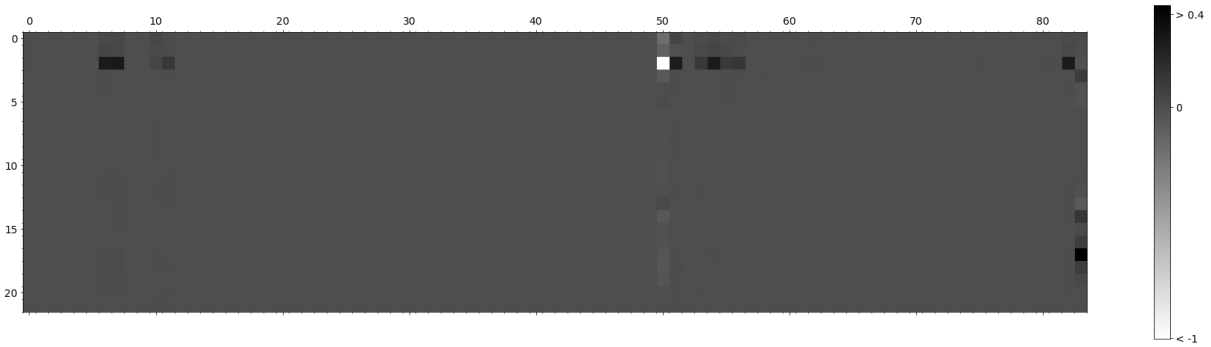


Figure 42: Feature attributions for end of project phase

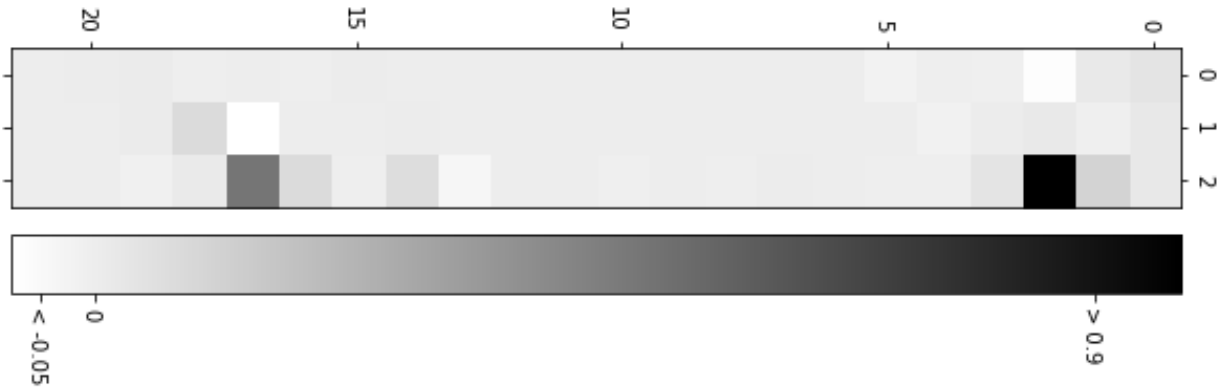


Figure 43: Time attributions for end of project phase

Activity attributions for first and last 100 activities during these weeks are shown in figure 44 and 45. In figure 44 and 45 x-axis shown activities in this part of project while y-axis shows number of weeks, the attributions are mapped using a gray scale.

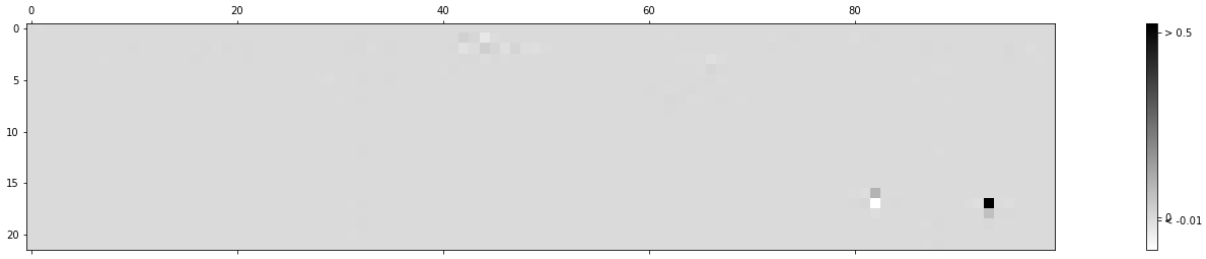


Figure 44: Activity attributions of first 100 activities for end of project phase

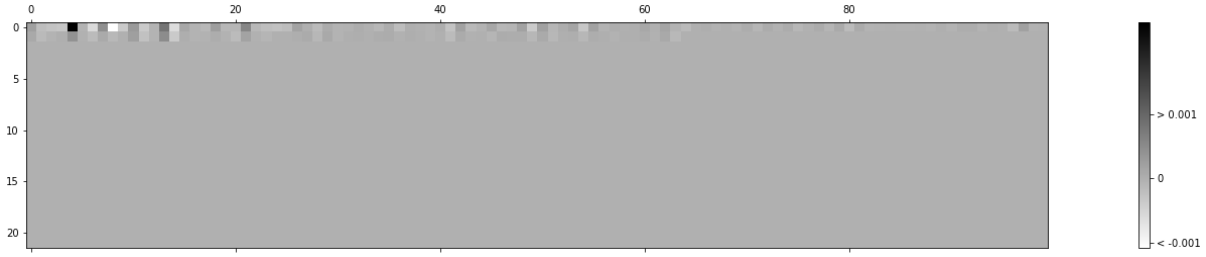


Figure 45: Activity attributions of last 100 activities for end of project phase

4.4.5 Measure of linearity for test set

Linearity is calculated for test set by the method described in section 2.9. The results are plotted graphically for every instance in the test set. These are shown in figure 46. the x-axis shows number of weeks in testing data while y-axis signifies the amount of linearity found a certain week, where values closer to 1 means more linearity.

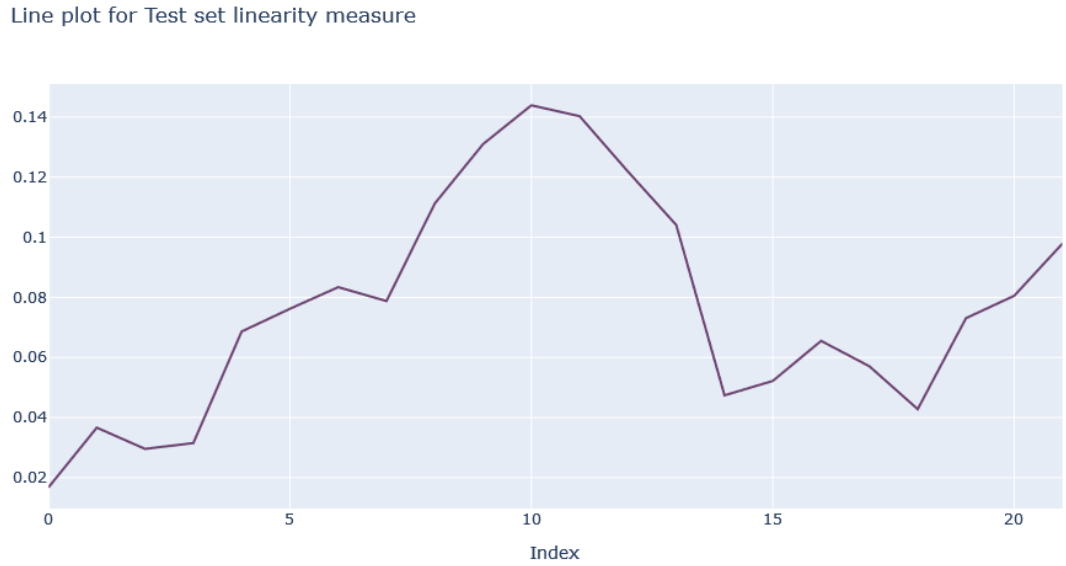


Figure 46: Linearity measure results for testing data in dataset 2

5 Discussion

5.1 Limitations

Due to the nature of data and ML methods incorporated for analysis there are limitations for the methods employed. The first limitation is the regarding data and then there are some model limitations for each method.

Data limitations exist because the project data used for analysis comes from a single project data with origins from a single company. This imposes the limitations on how the data handling was done and therefore, this method cannot be generalized for every project data. For example, the company logs several reference fields, flags and date fields which could be unique to this project data. It is therefore, not recommended to follow same approach for data handling as this thesis for every project data.

Another limitation from data is that it consists of a single project data and thus the data handling and target generation process was adapted for this use case. In case of application to a different project data these steps for data handling must be adopted to capture the dynamics of the data available. This data is also limited to the energy construction industry and therefore, transfer of these methods to a different domain requires handling procedures and adaptations. This study is thus confined to single project data and multiple project data could improve data quality and model performance.

Since the data consisted of a single project, the amount of data is small for dataset 1 and moderate for dataset 2 and 3. However, it is still a limiting factor for analysis since the test set contains distribution of data that the algorithm has never seen. ML models make a strong assumption of same distribution of data for training and test data (Sugiyama et al., 2007) (McGaughey et al., 2016) and therefore, the test set and train set distribution is a limiting factor for this analysis. However, subsampling of training set is not considered because of small size and statistically incorrect assumption made in subsampling technique of having seen the test data. The assumptions made about missing data is also a limiting factor for this analysis since the data structure used to store project data is not of high quality.

A more general limitation for data is the amount of data needed and the expertise required to handle heterogenous data. In a live data capturing the amount of data can be enormous and thus any extension of the capabilities will require consideration of time spent on data handling and procedures of data capture. The data was shaped to fit the RQ posed and there could be other data handling and pre-processing steps that can be applied for better granularity of data points.

A limitation for hyperparameters tuning is that the method employed is not good for scaled optimization (Falkner et al., 2018). If more data is available however, then this step can improve model performance. A strong assumption was made for generation of dataset 1 which could be unrealistic in real life scenarios. The assumption made was that in every week only one statistically significant number should exist for each feature, which means that every activity in that week was assumed to be of same importance for every feature. However, to capture any extreme or mean effects of these aggregations different measures were calculated such as mean, sum, minimum and maximum of a feature. Dataset 2 also has some limitations in data generation, it assumes that each

week has same number of activities which was supposed to be 3000 for each week since the largest number of activities in any week was less than this number. If an activity is missing, then a zero is inserted in that place. It also imposes a limitation on its feature extraction by CNN layers since the data is inserted at top left of the array therefore, the corner features might get less convolutions. This effect can eb reduced by using stride parameter of CNN however, this adds multiplicative complexity in model parameters and thus was not employed. A simple solution to this problem could be to insert padding in number of features and activities more which should be explored in future. Dataset 3 comes with limitations of its own, it does not assume any inherent structure for any week and activities in this dataset are not sorted by any column such as progress. However, a study conducted by (MacGregor, 1988) suggested finding an optimal sequence of examples in dataset for maximization of accuracy, modern algorithms however assumes no interaction between examples in time-independent classification task such as the task for dataset 3.

The data extracted through PM software imposes some limitations as well, due to higher complexity of the data structure in several tables and time constraints complex data extraction and transformation was not performed since the goal of the thesis is not extract only relevant features and not every feature possible from PM software. These complex data extraction process could further improve the results but were deemed unnecessary given the RQ of the thesis. These operations might include resource constraints, calendar information and interactions with activities, activity flagging for being on critical path and more.

5.2 RQ1

This subsection discusses the results and methods adopted for data quality assurance of project plan data. These results and methods adopted for extraction, processing and wrangling of dataset can be attributed to similar project data structures. Data maturity to support ML is attributed as one of the limiting factors for application of AI and ML methods in organizations (Akkiraju et al., 2020). Thus, an in-depth discussion of PM data will be carried out in this subsection which answers the first RQ on data quality and suitability of PM plan data for ML applications.

The data extraction from PM software used was done via SQL queries. The data model for the software was provided but the information contained was less explanatory than required for efficient data processing. Period status updates which is normally made at specific intervals during lifetime of a project was recorded in separate tables for activities and resources however, during the calculation of whole project update historic information of state of project is lost. An example of this is when a second period status update is issued the whole project state at first period status update is lost and thus cannot be recovered. This saves the storage and processing time for PM software however, for ML applications these historic project states are important and no direct way of retrieving these states inhibits the efficient ML workflow during the first step of CRISP-DM methodology (Wirth & Hipp, 2000). The first step in any data mining methodology is related to data understanding and therefore data maturity is important in applications for ML.

Number of recording variables in PM has increased and therefore complex data models are needed. The PM software is required to have some flexibility for different company needs as well since each company has different policies for planning and might adapt their needs for a particular project. Due to uniqueness of every project and diverse

structures of PM methodologies adopted PM software are designed to accommodate this requirement which inhibits its effective use in data mining applications. Manual tasks are required by means of data engineering for effective extract transform and load (ETL) operations. This consumes large amount of time during the lifecycle of ML model development. Effective ML is iterative process (reference) employing constant loop that requires feedbacks and continual improvements of data understanding and preparation (Wirth & Hipp, 2000). The data extracted from PM software had to be extracted and pre-processed several times during the lifecycle of experimentation. The generation of three different datasets is an example of how this loop was carried out continuously. PM software include the functions of reporting for different project metrics and statistics however, they are not data enrich and requires formatting along with extraction of manual data which is not directly available through the reporting functions. An example of this is the extraction of weekly number of variation orders and weekly project overview.

The manual extraction of data produces redundancies since in joining of tables extreme care must be taken and domain knowledge is necessary to be able to join these tables and generate ML compatible data format. The data extracted were more than 10 tables and through continual loop of testing ML models' data was made compatible with the requirements of the RQs. For example, activities table contain a snapshot of whole project status with respect to activities at certain time while period status of activities table contains historic information of activities and while joining these tables some columns are of same name which requires domain knowledge and PM software knowledge to capture and handle.

The data quality of project plans can be assessed for use in ML using framework provided by (Merino et al., 2016). According to this framework two type of adequacies are required for data to fulfil the need to be used in a particular context, contextual adequacy, and temporal adequacy. Adequacy here means the ability of PM plan data to be used in ML satisfactorily. Contextual adequacy means for data to have adequate capability to be used in a context without consideration of data formats (Merino et al., 2016). The requirements of data for this adequacy is to have relevancy, completeness, semantically interoperability, semantic accuracy, credibility, confidentiality and compliance. The data relevancy is sufficient for PM software as the data has capability to be used in ML applications. The data is also not complete since some project characteristics such as cost information is not reliable and complete and the data also contain some inconsistencies as there are duplications when generating datasets. Sematic accuracy is adequate as data signifies a real event. Data is also not entirely credible as data comes from logging of different events by several people and it is prone to errors such as missing a calendar attached to an event or activity, this in turn requires more effort to be analyzed and filtered for increased credibility. The PM software however provides sufficient data confidentiality and compliance by having reporting and exclusive access options for users.

Another dimension of data adequacy is temporal adequacy which means to have timely adequate data. It has several dimensions including data concurrency, currency of data, timeliness, frequency, and time consistency (Merino et al., 2016). The data from PM software is concurrent since the software provides functionality to extract information as needed for a time slot. Currency of data is ambiguous since some tables have different time dimensions than others for example, baseline information storage table and activities table might not be compatible for merging. The data however is timely as it is

updated as required. The frequency of data is also appropriate since in PM data future desired states are known and for ML applications this is important because we can forecast or predict for known time period. Time consistency of data is also ambiguous since some tables have incoherent data for different events for example, in status updates table all status updates are saved including baseline updates.

PM software are not yet completely equipped with compatibility with ML lifecycles. These software, however, are data enrich and can be utilized in ML applications with certain domain and Pm software knowledge. The readiness for the software could be improved by proving a functionality of data extraction through direct connections with the development environment and the access to reporting functionalities. Another improvement could be inclusion of project snapshots tables which contains all the project data at every status update in compressed format and allows the capture of all fields as required such as reference fields, flag information, resource information, calculated fields and date fields etc. the data tables can also be semantically structured to give a ML practitioner the required information on technicalities of data models and structure of data.

5.3 RQ2

This subsection discusses the validity of data given the developed datasets for improving the reliability of results as pointed out in RQ2. The validity of data is important for insights from the models and data reliability. Data quality and predictive power of datasets generated for purpose of the VO count prediction is directly related to reliability of data according to quality in use concept (Merino et al., 2016). The data quality is assured by continual feedback on assumptions made while development of datasets from PM software representatives and company representatives. The predictive power of model is validated by two means, one by analysing the inter-dependence of generated dataset 1 using VIF. The process of removing redundant data as explained in section 3.5 also provides insight into the validity of data. The process of removing redundant features reduced the feature space from 324 to 57 features, this implied that 267 features could have been directly evaluated using a linear model and one variable. An example is the feature current_progress_mean which is the mean progress for every activity in a week for dataset 1, this feature was found to have large VIF factor which means that the feature is directly predictable from one of the features in the dataset.

A more rigorous validation is done by developing a model for prediction of earned quantity for next week and compared to previous research done by (Iranmanesh & Zarezadeh, 2008) the results of models developed are shown in section 4.3. The MAPE performance metric was calculated and lowest for any dataset model is 0.27 percent on training, 0.16 percent on validation and 0.52 percent on test sets which is much lower than reported lowest error of 5.72 percent in the research by (Iranmanesh & Zarezadeh, 2008). The validation results also show the validation of dataset generation process and importance of correct data pre-processing steps in application of ML. The validation is important since RQ put forward in the thesis has not yet been studied in a single project context and thus validation increases the authenticity and reliability of models. It also provides extensive support in correct interpretation of models and builds trust in data.

5.4 RQ3

This subsection discusses the crux of this thesis where a discussion will be done on the possibility of predicting variation orders and the results. Prediction of changes in projects

is considered a difficult task in PM literature (Chen, 2015). However, these predictions could improve PM scope management and thus is deemed important. Previous research was done for prediction of changes in a project using data from different projects and using aggregated project features and domain expert knowledge. This research aims to find prediction capability of a single project data that can help decision makers in intra project planning. Scope changes are causes of disruptions in a project, it can increase the cost and timeline of a project. It is recommended to make controlled changes in a project (Madhuri et al., 2018). However, a limitation for change prediction by domain experts stems from the fact that project managers have limited knowledge about the project scope especially during the front-end phase of project (Andersen et al., 2011). As pointed out by (Olsson, 2006) changes in a project should be controlled to limit negative effects of the changes.

As directed by the CRISP-DM methodology (Wirth & Hipp, 2000) RQ3 was handled by using iterative method. Using three datasets presents a unique overview of effect of data structure on model performance as well. Dataset 1 was aggregated values of activities and resources in a week and thus variance is lost in the data. This is evident from the results in section 4.1.1 where the results are worse than results from dataset 2 in which all the variances of features were kept and no aggregation was performed. The MSE on training, validation and test set are too distant from one another which points that the model is overfitted and have high variance in its predictions. When these models predictions are compared graphically it is more evident LSTM model on dataset 1 performed better than traditional ML algorithms such as extra trees, LGBM and gradient boosting methods. The table 16 summarizes the MSE errors using different algorithms for dataset 1. Although all these methods are better than a baseline method of predicting mean, LSTM model had the best result on dataset 1.

	Data partition used		
Model	Training set	Validation set	Test set
Baseline	1.0224	0.7857	0.4690
LSTM	0.1351	0.7832	0.7399
LGBM	0.0371	0.7603	0.7592
Extra Trees	0	0.7178	0.6143
GBR	0.0933	0.7836	1.3289
ADA boost	0.7508	0.6850	0.4028
CAT boost	0.0035	0.8721	0.4003

Table 16: MSE of different models on dataset 1

Although LSTM generalizes to the data more than rest of these models, CAT boost provides easy models for model interpretation and therefore feature importance were analysed to understand which factors influenced the model more. The most important feature was the maximum difference of original and current early start date for the week. This gives us insight to monitor the differences between original plan dates and current plan dates for an early warning sign of VOs. The second feature of importance is week of per start which highlights the fact that VOs are cyclic in nature and only appear at certain times in a year. This is intuitive for a ML model since this data was not provided to ML model, but it was able to detect this behaviour. The other factors that influence issue of VO are change in schedule variance, change in early start dates, cost variance, increase in total duration of activities, minimum total float in a week, free float increases etc. This provides a better understanding of when a VO is likely to be issued. If in a certain week the total activity duration has increased too much along with these factors, then next

week have a higher chance of VO issuance. However, these attributes are not to be understood as standalone influencers on VO issuance since a model is non-linear and the interactions between these features are not completely explained.

Dataset 2 was specially shaped to support use of CNN architectures. It was hypothesized that these CNN models would be better at extracting spatial features between different activities in a week than simple aggregation of features over a week such as in dataset 1. Several architectures were tested and developed following Crisp-DM approach however, two of the most promising models are included in this thesis. The results for these models are shown in section 4.1.2 and summarized in table 17. First model is combination of CNN and LSTM where the output of CNN is fed into LSTM layer to capture the time dependencies in the data. The results suggest very good performance on task of VO count prediction. The second model had same architecture with inclusion of self-attention mechanism for LSTM layer. This layer acts as a filter for LSTM and provides the capability of providing more attention to important points in time. The results of this model suggest a pattern of underfitting the data since this model has much more convolutions, filters, and attention mechanism therefore it requires larger amount of data and time for training. Due to limitations of data this model's performance was worse than simple CNN and LSTM model but still better than simple ML methods like LGBM and others.

Model	Training set	Validation set	Test set
Baseline	1.0224	0.7857	0.4690
CNN+LSTM	0.1865	0.9309	0.2823
CNN+ LSTM+ self-attention	0.5178	0.9619	0.4583

Table 17: MSE of different models on dataset 2

Classification methods were also tried for best performing models in regression task. Two models were developed for this purpose, one with CNN and LSTM layers and other with addition of attention method. The accuracy metric used is F1 score due to imbalance in target classes. Class 1 means that next week will have a VO and class 0 means no VO for next week. For comparison of these models a baseline was considered with a naive method which predicts 1 for every week. The results for the classification on F1 metric are shown in table 18. The results suggest that classification is an easier task than prediction of exact number of VOs for next week however, by classification project managers cannot prepare completely for the variation orders since the quantity of VO will be unknown. A more sophisticated approach could be stacking of regression models on top of classification models with highest accuracy. This will lead to better results since the weeks which have no VO will be predicted with greater accuracy using classifier. MIL models did not show good results since the models developed for MIL made strong assumptions about the positive and negative bags. The models assumed that every week must have certain number of activities that affect the labels for a week which is not a realistic assumption. In any week VOs are not affected by just certain activities alone and number of VOs should be a function of interactions between activities that is the reason for MIL worse results in this case of VO prediction. However, MIL is a great method for data types such as project data and in case of a different objective that can make this assumption of having effect on a week by only certain number of activities this method could provide good results.

Model	Training set	Validation set	Test set
Baseline	0.3137	0.2400	0.1999
CNN+LSTM	0.8235	1.0	0.8889
CNN+LSTM+ self-attention	0.8571	1.0	0.8889
MIL - STK	0.4571	0.3333	0.7272
MIL - MICA	0	0	0

Table 18: F1 score results of different models on classification task

As discussed in section 2.9 model explainability is important and with the current use case it becomes much more important since project management team could have insight into planning strategies and how to structure the project to avoid VO as much as possible. Due to hybrid structure of models two methods were employed to interpret the model input effects on output. Attributions were calculated for only one model that contained CNN and LSTM layers. Attributions suggest the impact of inputs on output and linearity measure helps us understand if a model is acting as a linear model around some instances.

The attributions are calculated for effect of features, time stamps and activities for 4 parts of the data. At the start of the project which is supposed to last for first 45 weeks of data and the attributions are shown in section 4.4.1. By looking at the feature attributions for this part of data the most important features seem to be total planned scope, current scope, original planned quantity, resource type, current quantity, days till current late finish, days till revised plan early start, remaining hours and earned hours. The most impactful features are current scope, total scope, original planned quantity, current quantity, and remaining hours. By looking at time attributions it suggests that number of VOs are highly dependent on last week and time dependency decreases as time passes. The activities attributions give insight into the impact of activities completion effect on the model. In first 100 activities which are the least completed activities have less impact on the model than last 120 activities which are most completed. If we combine the effects of these attributions intuitively then we can see that it makes sense for the model to make decision based on most completed activities on the features that it identifies as important such as current quantity and remaining hours etc. This is a key insight from the model. Any project in the start phase will more likely to have a VO if an activity is near completion but we cannot directly extract linear relationship between planned quantity, current quantity and percentage completion of activity due to non-linear nature of model and complex interaction and feature extraction by intermediate CNN layers.

In the middle of the data that lasts 35 weeks after end of start part of data, the effects of the features decrease and now the model attributes nearly even importance to the features and attributes more importance to earned hours feature. However, the impact of current quantity is reduced. This is natural in a project lifecycle as mentioned by (Olsson, 2006) uncertainty in projects decrease gradually throughout the lifecycle of project. The model similarly attributes less weight to current week and tries to capture the time dependencies more by weighting each time steps as required through complex transformations. By the activity attributions in this part of data the model imitates the same behaviour by giving more importance to the activities that are near completion. In the next phase of the project, that lasts for last 33 weeks of training data the model gives more importance to expended quantities, earned quantities, current scope, total scope and earned hours. During the first phase of project data the model did not gave

too much weight to earned hours but now with the inclusion of expended quantity and more weight to earned quantity the model is keeping track of the planned and expended quantities while nonlinearly mapping other features. In this part of data, the model behaviour changes for activities now the model is giving more importance to the activities that are least completed while keeping track of activities that are nearly completed. This gives an important insight into the project data, if an activity is least completed during the last part of the project, then it is more likely to have a VO issued. However, similar to the other cases the exact working of model is not accessible with these attributions.

In the test data the model attributes more weight to a more important feature seems to be performance factor, expended quantity, and current quantity. The time importance is given more to the current week while activities that are least completed are given more importance than activities which are nearly completed. Linearity measure for test data is also important to check if model behaviour is linear for any week of data. the results are presented in section 4.4.5 while taking similar instances from train set into account. The results suggest that model behaviour is not linear and thus the attributions cannot be directly interpreted as having a linear effect on the model.

All of the models developed and tested suggest that the number of variational order prediction for next week is a predictable target and this research should be extended to include other factors into modelling. The conclusion is based on comparison of models with the baseline models developed. All the models performed better than the baseline of predicting mean in case of regression and class 1 in case of classification. The impact of classification modelling is also huge in this special case of VO count prediction since the target values contain large number of zeros a special model can be utilized that first classifies a week as probable VO and then uses regression models to predict number of variational orders.

5.5 RQ4

This subsection discussed the implications of successful ML method implementation for change management using single project data as pointed out in the research question. Prediction of changes in a project can have great planning significance for project managers (Chen, 2015). According to (Høylandskjær, 2018) change management can reduce the negative effects of scope changes. By implementing a good performing ML model for VO prediction one of the biggest impact could be to improve labour productivity. (Moselhi et al., 2005) studied the impact of variation orders on labour productivity and used variation order data for prediction. Since by using the model and extending it to predict total hour of changes we can forecast anticipated changes in future therefore the impacts of these VOs can be reduced by successful change management. Similarly (Chen, 2015) recommended use of proactive change management by using forecasting tools such as ML models for better change management. He proposes a strict change control process that can reduce the change costs. The models developed in the thesis can help planners in reduction of costs by providing essential information such as number of change orders and help them plan efficiently. (Mir & Pinnington, 2014) found 44.9% of variance in project success can be explained by different performance factors. The study included PM KPIs which includes earned value, prediction of change orders can help improve planned value of project and thus can indirectly improve several KPIs. (Gunduz & Mohammad, 2020) suggested limitations of impacts of change orders for reduction in reworks, demolitions, and late

payments. The proposed model can be used for this purpose by reducing uncertainty in number of anticipated VOs. The same model architecture of CNN and LSTM can be extended for long term forecasts such as 2 weeks, 4 weeks, or 6 weeks ahead forecasts. The initial research is to show that VOs are a predictable target using a single project data.

By successful implementation of this ML model project planning phase can also be improved. A baseline is set at every variation order acceptance and project plan is generated, implementing ML models such as the proposed model provides insight into the VO prone weeks in the proposed plan at a baseline. This phenomenon can also be run at start of a project in planning phase if a project plan is passed through a ML model trained on the plan data using same CNN and LSTM architecture. The data structure however, must be adapted to the use case. An example is using this current plan data in three-dimensional format where number of rows represent activities, columns represent features planned and third dimension represent simulation model results. These simulation models are already available for project forecasting and thus can be employed as input to ML models for prediction of VOs in a project before project start. Planning can thus be improved while project is ongoing or in early phases of project before project start.

The predictive power of a single project data is less than predictive power of multi project data, this is due to known concept of covariate shift in modelling literature. This effect is when training and testing data contains different distributions of input data any maximum likely estimator such as used in traditional ML models loses its efficiency (Sugiyama et al., 2007). This effect is present in a single project data since the data consist of a single project and division of data into training, validation and test set breaks the whole distribution of data. This effect can be visualized by plotting two principal components (Jolliffe, 2002) of data for two sets. First set is data contained in training and validation sets and second set is data contained in test set. Visually we can see this effect by plotting both components of principal component analysis (PCA) (Jolliffe, 2002). The resulting density plots are shown in figure 47 and 48. We can clearly see the difference in distribution of training and test set for both components, this thus limits the predictive power of a single dataset. By having multiple projects data this covariate shift effect can be minimized and better generalizations can be made from the results of ML models. By the performance of models developed it is evident however, that a single project data can be used to predict future states of a project with carefully modelling data and selecting appropriate ML models. The use of multiple project data, however, might also introduce noise in the data since every project is unique and characteristics of portfolio of projects might not give valuable information for a specific task.

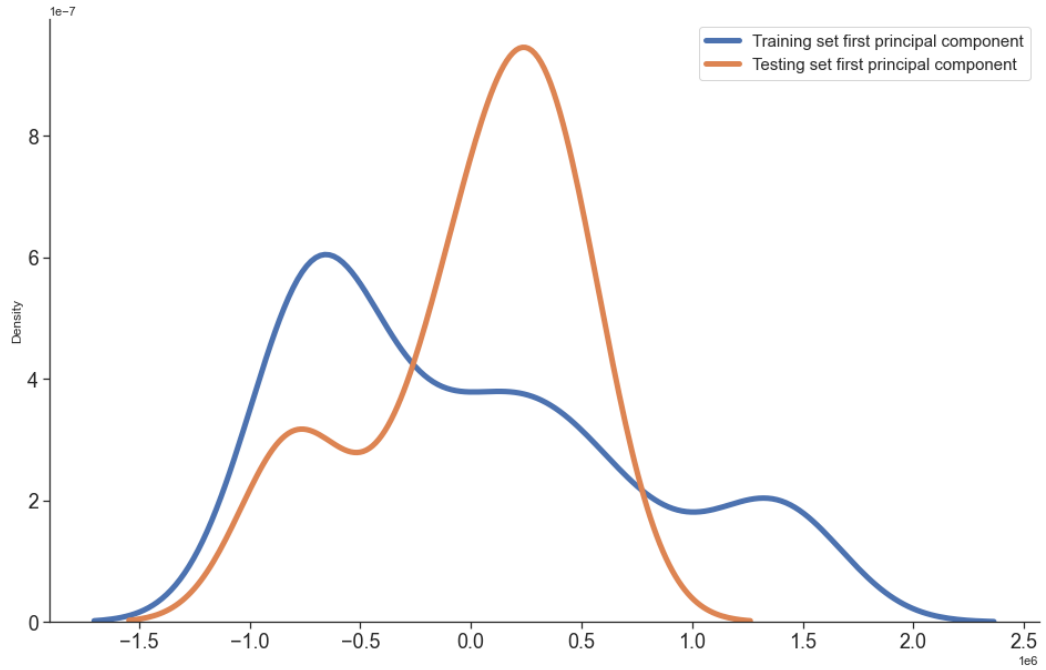


Figure 47: Density plot for training and testing data first principal component

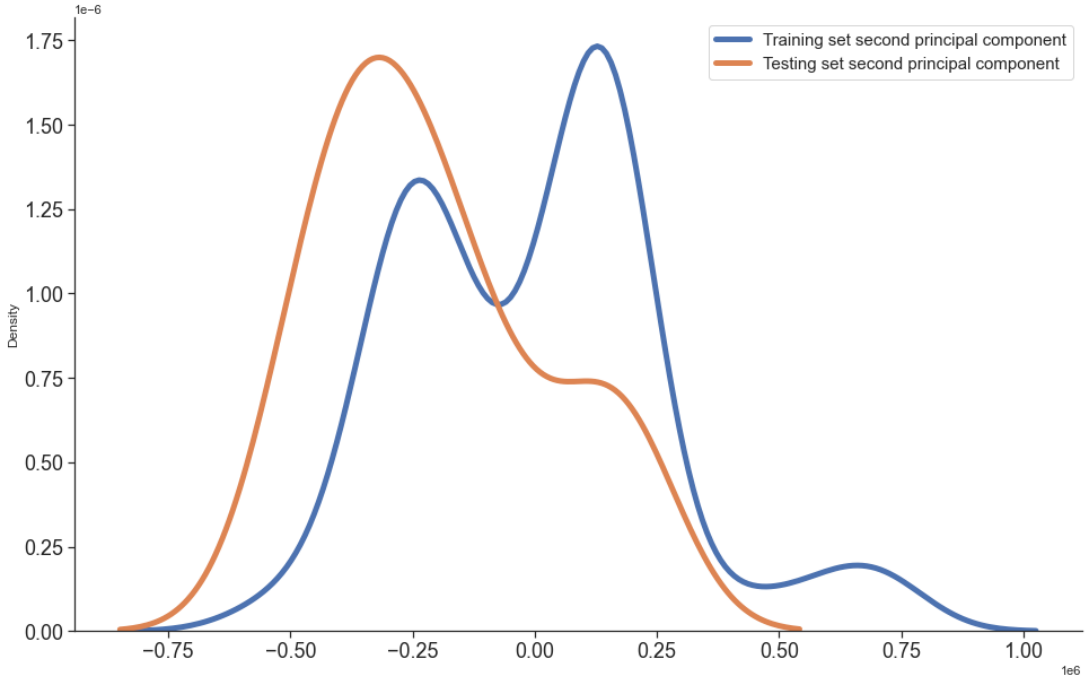


Figure 48: Density plot for training and testing data second principal component

6 Conclusion

Research questions put forward in this thesis have practical implications for contractors and PM software developers. The first insight from the research questions is that PM software data storage and semantics should be improved for compatibility with ML automation and easier data processing. Current state of these software packages has severe limitations in applicability of ML directly. Due to complexity of these software packages huge amount of time is spent on data extraction, loading, pre-processing, and targets generation. Contractors using these software packages are also required to be vigilant in their inputs to these PM software packages for data validity and reliability to be of highest quality possible. After data generation validity of the data generation process is important for reliability of ML algorithms and interpretation of these results. Garbage in garbage out principle should always be kept in mind while working with project data due to higher complexity of data and concepts related to PM. In the validation of data another insight from the data is that earned quantity for next week is a predictable target with much greater accuracy and this in turn can reduce uncertainty in the project.

Variation order prediction is a historic topic of discussion for PM professionals (Shafaat et al., 2016). Its importance lies in the fact that scope creep and scope drift are one of the project success limitation factors. By using a ML model and structured data process VO can be predicted with more than 80% accuracy however, the model structure and data generation process plays a vital role in development of accurate systems. The difference between results of three generated datasets from same project data signifies the fact that customized data and model structures are important for successful implementation of ML and AI methods in project data. Important implications for planning and control management are to limit the uncertainty in project lifecycle by reduction of scope creep and scope changes. The methods employed also provides understanding of factors that influence issuance of VO such as current scope, total scope, days remaining in current late finish and revised plan early start, remaining hours and earned hours. The importance for these features however changes during the lifecycle of a project and it was found that the decision making by ML model is not linear in nature. At the end of project some other features are included by model for decision making such as expended quantity and performance factor. It is natural to consider scope creep is not only dependent on current time but might also be affected by past states of project, this behaviour is captured by the model in middle phase of project while at the start and end model gives more importance to recent past which is plausible because at both extremes of a project lifecycle recent events affect the issuance of VO more. Model also emphasises the activities differently during different phases of a project, in the start it emphasizes more on activities that were almost complete while near the end of completion of project activities that are least completed are of most importance for the issuance of VO. This understanding of model behaviour for a certain decision making enhances our understanding of how the project progress have different effects on changes implemented and issued during a project and suggest a dynamic approach for change management during different phases of a project.

VO predictions for a single project plan data however suffers from covariate shift and special care during data handling must be taken, inclusion of several projects data should

be tested but by the nature of projects a different project data might induce more noise than insight for the model. By utilizing a ML model for VO prediction PM can successfully control scope creep in a project and thus this applicability of ML methods on project data has high impact on project success. Project control methods can be improved because of low uncertainty in project for weeks ahead. A single project data is however, sufficient for project metrics forecasting and variation order forecasts given some time has passed. This time limitation is due to use of single project data however, if portfolio of projects exists and can be used to train the model then project planning team can use these ML models for better planning of projects and reduce uncertainty and scope changes before the project start.

This research is the first step in understanding high level impact and dependencies in variation order prediction problem. Number of available modelling techniques are numerous and special care must be taken in use of ML algorithms. Methods such as MIL and its variants assume strong characteristics about the data and might not be applicable to every use case in project data however, the structure of this method fits perfectly with the project data and other use cases must be explored where the assumption holds.

6.1 Further research

Further research must be carried out to understand deeper insights from the models and ML techniques for knowledge discovery. The use case of number of VO prediction could be modelled with count-based models and zero-inflated models which should be explored. The presence of reference fields in data provides unique opportunity to understand how certain disciplines, organizational structure, construction phase, category of work, and systems used in an activity would impact the issuance of VO. This step could be key in deeper understanding of scope changes and provide better recommendations for planning and change management strategies. The option to include several projects data in training might also be explored with special consideration on grouping of projects into similar groups and training different models for each group to reduce noise effects on models by different type of project. The application of new and efficient structures for Spatio-temporal data can also be explored. The use of loss function might also be adapted to capture complex relations in the data, for example in the dataset 2 arcface loss function could be employed to capture latent angle feature interactions between activities, timesteps and features. Exploration of network in network architectures might also be studied where instead of maximum pooling layers between convolution layers a network is present to extract features and reduce dimensions of data. Attention mechanisms are gaining research interest in capture of complex dependencies and should be explored as well. The amount of knowledge discovery in project plan data is huge and it should be explored to continue development of better PM methodologies using ML and AI methods.

References

Adadi, A. (2021). A survey on data-efficient algorithms in big data era. *Journal of Big Data*, *8*(1), 1-54.

Agarwal, N., & Das, S. (2020). Interpretable machine learning tools: A survey. 2020 IEEE Symposium Series on Computational Intelligence (SSCI),

Ahuja, H. N., Dozzi, S., & Abourizk, S. (1994). *Project management: techniques in planning and controlling construction projects*. John Wiley & Sons.

Akalu, M. M. (2001). Re-examining project appraisal and control: developing a focus on wealth creation. *International journal of project management*, *19*(7), 375-383.

Akkiraju, R., Sinha, V., Xu, A., Mahmud, J., Gundecha, P., Liu, Z., Liu, X., & Schumacher, J. (2020). Characterizing machine learning processes: A maturity framework. International Conference on Business Process Management,

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. 2017 international conference on engineering and technology (ICET),

Aldana, A., Hay, K., LeGrand, C., Schmidt, P., & Bereni, M. (2021). *Exploring the use of artificial intelligence (AI) solutions to improve the accuracy of project delivery forecasts*.

Aljuaid, T., & Sasi, S. (2016). Proper imputation techniques for missing values in data sets. 2016 international conference on data science and engineering (ICDSE),

Alotaibi, A. B., & Mafimisebi, O. P. (2016). Project management practice: redefining theoretical challenges in the 21st century. *Project Management*, *7*(1), 93-99.

Amoatey, C. T., & Anson, B. A. (2017). Investigating the major causes of scope creep in real estate construction projects in Ghana. *Journal of Facilities Management*.

Anastasopoulos, P. C., Labi, S., Bhargava, A., Bordat, C., & Mannering, F. L. (2010). Frequency of change orders in highway construction using alternate count-data modeling methods. *Journal of construction engineering and management*, *136*(8), 886-893.

Anbari, F. T. (2003). Earned value project management method and extensions. *Project Management Journal*, *34*(4), 12-23.

Andersen, B., Olsson, N. O., Onsøyen, L. E., & Spjelkavik, I. (2011). Post-project changes: occurrence, causes, and countermeasures. *International Journal of Managing Projects in Business*.

Angelopoulos, M., Kontakou, C., & Pollalis, Y. (2019). Digital Transformation and Lean Management. Challenges in the Energy Industry of Utilities. A Review.

Assaf, S. A., & Al-Hejji, S. (2006). Causes of delay in large construction projects. *International journal of project management*, *24*(4), 349-357.

Atkins, D. C., & Gallop, R. J. (2007). Rethinking how family researchers model infrequent outcomes: a tutorial on count regression and zero-inflated models. *Journal of Family Psychology*, *21*(4), 726.

Atluri, G., Karpatne, A., & Kumar, V. (2018). Spatio-temporal data mining: A survey of problems and methods. *ACM computing surveys (CSUR)*, *51*(4), 1-41.

Avison, D., Baskerville, R., & Myers, M. (2001). Controlling action research projects. *Information technology & people*.

Aydilek, I. B., & Arslan, A. (2013). A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm. *Information Sciences*, *233*, 25-35.

Badiru, A. B., & Osisanya, S. O. (2016). *Project management for the oil and gas industry: a world system approach*. CRC Press.

Badiru, A. B., & Sieger, D. B. (1998). Neural network as a simulation metamodel in economic analysis of risky projects. *European Journal of Operational Research*, *105*(1), 130-142.
Banker, S. (2019). Demand Planning Solutions Improve Forecasting By Consuming More And More Data. *Forbes*. https://www.forbes.com/sites/stevebanker/2019/04/01/demand-planning-solutions-improve-forecasting-by-consuming-more-and-more-data/?sh=98b3e183ee77
Bansal, S. K. (2014). Towards a semantic extract-transform-load (ETL) framework for big data integration. 2014 IEEE International Congress on Big Data,
Bean, R. (2018). The state of machine learning in business today. *Preuzeto od https://www. forbes. com/sites/ciocentral/2018/09/17/the-state-of-machine-learning-inbusiness-today*.
Beck, P. J., & Kovacs, D. (2018). Earned Schedule and the Use of Schedule Execution Reporting Metrics. International Pipeline Conference,
Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing* (pp. 1-4). Springer.
Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798-1828.
Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157-166.
Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, *24*.
Berry, W. D. (1993). *Understanding regression assumptions* (Vol. 92). Sage.
Besner, C., & Hobbs, B. (2013). Contextualized project management practice: A cluster analysis of practices and best practices. *Project Management Journal*, *44*(1), 17-34.
Bhanja, S., & Das, A. (2018). Impact of data normalization on deep neural network for time series forecasting. *arXiv preprint arXiv:1812.05519*.
Blanchard, B. S., Fabrycky, W. J., & Fabrycky, W. J. (1990). *Systems engineering and analysis* (Vol. 4). Prentice hall Englewood Cliffs, NJ.
Block, H. D., Knight Jr, B., & Rosenblatt, F. (1962). Analysis of a four-layer series-coupled perceptron. II. *Reviews of Modern Physics*, *34*(1), 135.
Bosu, M. F., & MacDonell, S. G. (2013). A taxonomy of data quality challenges in empirical software engineering. 2013 22nd Australian Software Engineering Conference,
Botchkarev, A. (2018). Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology. *arXiv preprint arXiv:1809.03006*.
Bower, D. C., & Finegan, A. D. (2009). New approaches in project performance evaluation techniques. *International Journal of Managing Projects in Business*.
Brownlee, J. (2022). Data preparation for machine learning. In.
Bubeck, S. (2014). Theory of convex optimization for machine learning. *arXiv preprint arXiv:1405.4980*, *15*.
Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 1-58.
Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, *40*(1), 16-28.
Chao, L.-C., & Chien, C.-F. (2010). A model for updating project S-curve by using neural networks and matching progress. *Automation in Construction*, *19*(1), 84-91.
Charoenngam, C., Coquinco, S., & Hadikusumo, B. (2003). Web-based application for managing change orders in construction projects. *Construction Innovation*.
Chen, C. (2015). *A proactive approach for change management and control on construction projects* UC Berkeley].

Chen, H. L., Chen, W. T., & Lin, Y. L. (2016). Earned value project management: Improving the predictive power of planned value. *International journal of project management*, *34*(1), 22-29.
Cheng, J., Fowler, J., Kempf, K., & Mason, S. (2015). Multi-mode resource-constrained project scheduling problems with non-preemptive activity splitting. *Computers & Operations Research*, *53*, 275-287.
Cheng, M.-Y., Wibowo, D. K., Prayogo, D., & Roy, A. F. (2015). Predicting productivity loss caused by change orders using the evolutionary fuzzy support vector machine inference model. *Journal of Civil Engineering and Management*, *21*(7), 881-892.
Cheng, M.-Y., Wu, Y.-W., & Wu, C.-F. (2010). Project success prediction using an evolutionary support vector machine inference model. *Automation in Construction*, *19*(3), 302-307.
Choi, S.-W., Lee, E.-B., & Kim, J.-H. (2021). The Engineering Machine-Learning Automation Platform (EMAP): A Big-Data-Driven AI Tool for Contractors' Sustainable Management Solutions for Plant Projects. *Sustainability*, *13*(18), 10384.
Chou, J.-S., Lin, C.-W., Pham, A.-D., & Shao, J.-Y. (2015). Optimized artificial intelligence models for predicting project award price. *Automation in Construction*, *54*, 106-115.
Choudhary, R., & Gianey, H. K. (2017). Comprehensive review on supervised machine learning algorithms.
Cioffi, D. F. (2006). Completing projects according to plans: an earned-value improvement index. *Journal of the Operational Research Society*, *57*(3), 290-295.
Colenso, K. (2000). Creating the work breakdown structure. *Artemis Management Systems*.
Copeland, B. J. (2000). The turing test. *Minds and Machines*, *10*(4), 519-539.
Corrales, D. C., Corrales, J. C., & Ledezma, A. (2018). How to address the data quality issues in regression models: a guided process for data cleaning. *Symmetry*, *10*(4), 99.
Cottrell, W. D. (1999). Simplified program evaluation and review technique (PERT). *Journal of construction engineering and management*, *125*(1), 16-22.
Cui, Y., & Olsson, N. O. (2009). Project flexibility in practice: An empirical study of reduction lists in large governmental projects. *International journal of project management*, *27*(5), 447-455.
Daniel, K. (2017). Thinking, fast and slow. In.
de Carvalho, M. M., Patah, L. A., & de Souza Bido, D. (2015). Project management and its effects on project success: Cross-country and cross-industry comparisons. *International journal of project management*, *33*(7), 1509-1522.
Demeulemeester, E. L., & Herroelen, W. S. (2006). *Project scheduling: a research handbook* (Vol. 49). Springer Science & Business Media.
Devi, T. R., & Reddy, V. S. (2012). Work breakdown structure of the project. *Int J Eng Res Appl*, *2*(2), 683-686.
Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, *89*(1-2), 31-71.
Dogan, N., & Tanrikulu, Z. (2013). A comparative analysis of classification algorithms in data mining for accuracy, speed and robustness. *Information Technology and Management*, *14*(2), 105-124.
Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78-87.
Dong, X. L., & Srivastava, D. (2013). Big data integration. 2013 IEEE 29th international conference on data engineering (ICDE),
Douglas III, E. E. (2000). Project trends and change control. *AACE International Transactions*, C10A.
Drezet, L.-E., & Billaut, J.-C. (2008). A project scheduling problem with labour constraints and time-dependent activities requirements. *International Journal of Production Economics*, *112*(1), 217-225.
Drucker, H. (1997). Improving regressors using boosting techniques. ICML,

Dumoulin, V., & Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.
Dupláková, D., Radchenko, S., Knapčíková, L., Hatala, M., & Duplák, J. (2017). Application of simulation tool for scheduling in engineering. *Acta Simulation*, *3*(1), 5-10.
Edum-Fotwe, F. T., & McCaffer, R. (2000). Developing project management competency: perspectives from the construction industry. *International journal of project management*, *18*(2), 111-124.
El Naqa, I., & Murphy, M. J. (2015). What is machine learning? In *machine learning in radiation oncology* (pp. 3-11). Springer.
Enshassi, A., Arain, F., & Al-Raee, S. (2010). Causes of variation orders in construction projects in the Gaza Strip. *Journal of Civil Engineering and Management*, *16*(4), 540-551.
Eyob, E. (2009). *Social Implications of Data Mining and Information Privacy: Interdisciplinary Frameworks and Solutions: Interdisciplinary Frameworks and Solutions*. IGI Global.
Falkner, S., Klein, A., & Hutter, F. (2018). BOHB: Robust and efficient hyperparameter optimization at scale. International Conference on Machine Learning,
Field, A. (2013). *Discovering statistics using IBM SPSS statistics*. sage.
Fleming, Q. W., & Koppelman, J. M. (2016). Earned value project management.
Foulds, J., & Frank, E. (2010). A review of multi-instance learning assumptions. *The knowledge engineering review*, *25*(1), 1-25.
Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, *55*(1), 119-139.
Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.
Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis*, *38*(4), 367-378.
García, C., García, J., López Martín, M., & Salmerón, R. (2015). Collinearity: revisiting the variance inflation factor in ridge regression. *Journal of Applied Statistics*, *42*(3), 648-661.
García, J. A. L., Peña, A. B., & Pérez, P. Y. P. (2017). Project control and computational intelligence: Trends and challenges. *International Journal of Computational Intelligence Systems*, *10*(1), 320.
Gärtner, T., Flach, P. A., Kowalczyk, A., & Smola, A. J. (2002). Multi-instance kernels. ICML,
Gers, F. A., & Schmidhuber, E. (2001). LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE transactions on neural networks*, *12*(6), 1333-1340.
Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, *63*(1), 3-42.
Ghenbasha, M. a. W. O., Wan Mohd Sabki and Omar, Wan and Ayob, Afizah. (2018). ASSESSING THE POTENTIAL IMPACTS OF VARIATION ORDERS ON JKR ROADWAY CONSTRUCTION PROJECTS-A CASE STUDY NORTHERN REGION OF MALAYSIA.
Globerson, S., & Zwikael, O. (2002). The impact of the project manager on project management planning processes. *Project Management Journal*, *33*(3), 58-64.
Gondia, A., Siam, A., El-Dakhakhni, W., & Nassar, A. H. (2020). Machine learning algorithms for construction projects delay risk prediction. *Journal of construction engineering and management*, *146*(1), 04019085.
Granger, C. W. (1999). Outline of forecast theory using generalized cost functions. *Spanish Economic Review*, *1*(2), 161-173.
Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks*, *18*(5-6), 602-610.
Green, C. (1969). Theorem proving by resolution as a basis for question-answering systems. *Machine intelligence*, *4*(3).

Grossberg, S. (2013). Recurrent neural networks. *Scholarpedia*, *8*(2), 1888.
Grzymala-Busse, J. W., & Hu, M. (2000). A comparison of several approaches to missing attribute values in data mining. International Conference on Rough Sets and Current Trends in Computing,
Gunduz, M., & Mohammad, K. O. (2020). Assessment of change order impact factors on construction project performance using analytic hierarchy process (AHP). *Technological and Economic Development of Economy*, *26*(1), 71-85.
Gupta, N., Mujumdar, S., Patel, H., Masuda, S., Panwar, N., Bandyopadhyay, S., Mehta, S., Guttula, S., Afzal, S., & Sharma Mittal, R. (2021). Data quality for machine learning tasks. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining,
Guyon, I., Sun-Hosoya, L., Boullé, M., Escalante, H. J., Escalera, S., Liu, Z., Jajetic, D., Ray, B., Saeed, M., & Sebag, M. (2019). Analysis of the AutoML challenge series. *Automated Machine Learning*, 177.
Hajizadeh, Y. (2019). Machine learning in oil and gas; a SWOT analysis approach. *Journal of Petroleum Science and Engineering*, *176*, 661-663.
Han, J.-H., Choi, D.-J., Park, S.-U., & Hong, S.-K. (2020). Hyperparameter optimization using a genetic algorithm considering verification time in a convolutional neural network. *Journal of Electrical Engineering & Technology*, *15*(2), 721-726.
Hancock, J. T., & Khoshgoftaar, T. M. (2020). CatBoost for big data: an interdisciplinary review. *Journal of Big Data*, *7*(1), 1-45.
Hanga, K. M., & Kovalchuk, Y. (2019). Machine learning and multi-agent systems in oil and gas industry applications: A survey. *Computer Science Review*, *34*, 100191.
Hanna, A. S., Camlic, R., Peterson, P. A., & Lee, M.-J. (2004). Cumulative effect of project changes for electrical and mechanical construction. *Journal of construction engineering and management*, *130*(6), 762-771.
Hartmann, S. (2013). Project scheduling with resource capacities and requests varying with time: a case study. *Flexible Services and Manufacturing Journal*, *25*(1), 74-93.
Hartmann, S., & Briskorn, D. (2022). An updated survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, *297*(1), 1-14.
Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2). Springer.
Hawkins, D. M. (1980). *Identification of outliers* (Vol. 11). Springer.
He, Z., Shu, F., Yang, Y., Li, M., & Wang, Q. (2012). An investigation on the feasibility of cross-project defect prediction. *Automated Software Engineering*, *19*(2), 167-199.
Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, *18*(7), 1527-1554.
Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504-507.
Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.
Hong, H. (1998). An efficient point estimate method for probabilistic analysis. *Reliability Engineering & System Safety*, *59*(3), 261-267.
Hong, H. G., Christiani, D. C., & Li, Y. (2019). Quantile regression for survival data in modern cancer research: expanding statistical tools for precision medicine. *Precision clinical medicine*, *2*(2), 90-99.
Høylandskjær, M. (2018). Managerial perceptions of scope creep in projects: A multiple-case study. In.
Hsieh, T.-y., Lu, S.-t., & Wu, C.-h. (2004). Statistical analysis of causes for change orders in metropolitan public works. *International journal of project management*, *22*(8), 679-686.
Hsu, M.-W., Dacre, N., & Senyo, P. (2021). Identifying Inter-Project Relationships with Recurrent Neural Networks: Towards an AI Framework of Project Success Prediction. *British Academy of Management*.

Huang, F., & Yates, A. (2012). Biased representation learning for domain adaptation. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning,
Huang, J.-C., Ko, K.-M., Shu, M.-H., & Hsu, B.-M. (2020). Application and comparison of several machine learning algorithms and their integration models in regression problems. *Neural Computing and Applications*, *32*(10), 5461-5469.
Huang, L., Jin, H., Yuan, P., & Chu, F. (2008). Duplicate records cleansing with length filtering and dynamic weighting. 2008 Fourth International Conference on Semantics, Knowledge and Grid,
Huang, P., Wen, C., Fu, L., Peng, Q., & Tang, Y. (2020). A deep learning approach for multi-attribute data: A study of train delay prediction in railway systems. *Information Sciences*, *516*, 234-253.
Huang, Y. (2018). Research on container throughput forecast of Qingdao port based on combined forecasting model. Proceedings of the 4th International Conference on Industrial and Business Engineering,
Ibbs, C. W. (1997). Quantitative impacts of project change: Size issues. *Journal of construction engineering and management*, *123*(3), 308-311.
Ibbs, C. W., Kwak, Y. H., Ng, T., & Odabasi, A. M. (2003). Project delivery systems and project change: Quantitative analysis. *Journal of construction engineering and management*, *129*(4), 382-387.
Ibbs, C. W., Wong, C. K., & Kwak, Y. H. (2001). Project change management system. *Journal of management in engineering*, *17*(3), 159-165.
Iranmanesh, S. H., & Zarezadeh, M. (2008). Application of artificial neural network to forecast actual cost of a project to improve earned value management system. World congress on science, engineering and technology,
Johnson, R. A., & Wichern, D. W. (2014). *Applied multivariate statistical analysis* (Vol. 6). Pearson London, UK:.
Jolliffe, I. T. (2002). *Principal component analysis for special types of data*. Springer.
Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, *21*(4), 345-383.
Jones, R. M. (2001). Lost productivity: Claims for the cumulative impact of multiple change orders. *Pub. Cont. LJ*, *31*, 1.
Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255-260.
Kartam, N. A. (1996). Making effective use of construction lessons learned in project life cycle. *Journal of construction engineering and management*, *122*(1), 14-21.
Kazemi, S. M., Goel, R., Eghbali, S., Ramanan, J., Sahota, J., Thakur, S., Wu, S., Smyth, C., Poupart, P., & Brubaker, M. (2019). Time2vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*.
Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, *30*.
Kerzner, H. (2002). *Strategic planning for project management using a project management maturity model*. John Wiley & Sons.
Kerzner, H. (2009). Project management: A systems approach to planning. *Scheduling, and Controlling*, *7*.
Kerzner, H. (2022). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. John Wiley & Sons.
Keshavarzian, S., & Silvius, G. (2022). The Perceived Relationship Between Sustainability in Project Management and Project Success. *The Journal of Modern Project Management*, *9*(3).
Khalid, S., Khalil, T., & Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. 2014 science and information conference,
Khamooshi, H., & Golafshani, H. (2014). EDM: Earned Duration Management, a new approach to schedule performance management and measurement. *International journal of project management*, *32*(6), 1019-1041.

Kim, Y., Huang, J., & Emery, S. (2016). Garbage in, garbage out: data collection, quality assessment and reporting standards for social media data use in health research, infodemiology and digital disease detection. *Journal of medical Internet research*, *18*(2), e4738.
Klaise, J., Van Looveren, A., Vacanti, G., & Coca, A. (2021). Alibi Explain: algorithms for explaining machine learning models. *Journal of Machine Learning Research*, *22*(181), 1-7.
Ko, C.-H., & Cheng, M.-Y. (2007). Dynamic prediction of project success using artificial intelligence. *Journal of construction engineering and management*, *133*(4), 316-324.
Kohonen, T., Barna, G., & Chrisley, R. L. (1988). Statistical pattern recognition with neural networks: benchmarking studies. ICNN,
Kolloch, M., & Dellermann, D. (2018). Digital innovation in the energy industry: The impact of controversies on the evolution of innovation ecosystems. *Technological Forecasting and Social Change*, *136*, 254-264.
Kou, G., Peng, Y., Shi, Y., & Xu, W. (2003). A set of data mining models to classify credit cardholder behavior. International Conference on Computational Science,
Kuhl, M. E., & Graciano, M. K. P. (2014). Project planning and predictive earned value analysis via simulation. Proceedings of the Winter Simulation Conference 2014,
Ladha, L., & Deepa, T. (2011). Feature selection methods and algorithms. *International journal on computer science and engineering*, *3*(5), 1787-1797.
Le Roux, W. H. (2008). Modelling and simulation-based support for interoperability exercises in preparation of 2010 FIFA World Cup South Africa.
LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.
Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. Proceedings of the 26th annual international conference on machine learning,
Lee, S. (2007). *Understanding and quantifying the impact of changes on construction labor productivity: Integration of productivity factors and quantification methods*. University of California, Berkeley.
Lee, S. H., Peña-Mora, F., & Park, M. (2006). Dynamic planning and control methodology for strategic and operational construction project management. *Automation in Construction*, *15*(1), 84-97.
LightGBM. *Advanced Topics* Retrieved 12.05.2022 from https://lightgbm.readthedocs.io/en/latest/Advanced-Topics.html
Ling, F. Y., Low, S. P., Wang, S., & Egbelakin, T. (2008). Models for predicting project performance in China using project management practices adopted by foreign AEC firms. *Journal of construction engineering and management*, *134*(12), 983-990.
Ling, F. Y. Y., & Liu, M. (2004). Using neural network to predict performance of design-build projects in Singapore. *Building and Environment*, *39*(10), 1263-1274.
Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.
Lipton, Z. C., Elkan, C., & Narayanaswamy, B. (2014). Thresholding classifiers to maximize F1 score. *arXiv preprint arXiv:1402.1892*.
Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, *234*, 11-26.
Liu, X., Chang, C., & Duyn, J. H. (2013). Decomposition of spontaneous brain activity into distinct fMRI co-activation patterns. *Frontiers in systems neuroscience*, *7*, 101.
Loh, W. Y. (2011). Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, *1*(1), 14-23.
Lou, Y., & Obukhov, M. (2017). Bdt: Gradient boosted decision tables for high accuracy and scoring efficiency. Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining,
Lu, M. (2002). Enhancing project evaluation and review technique simulation through artificial neural network-based input modeling. *Journal of construction engineering and management*, *128*(5), 438-445.

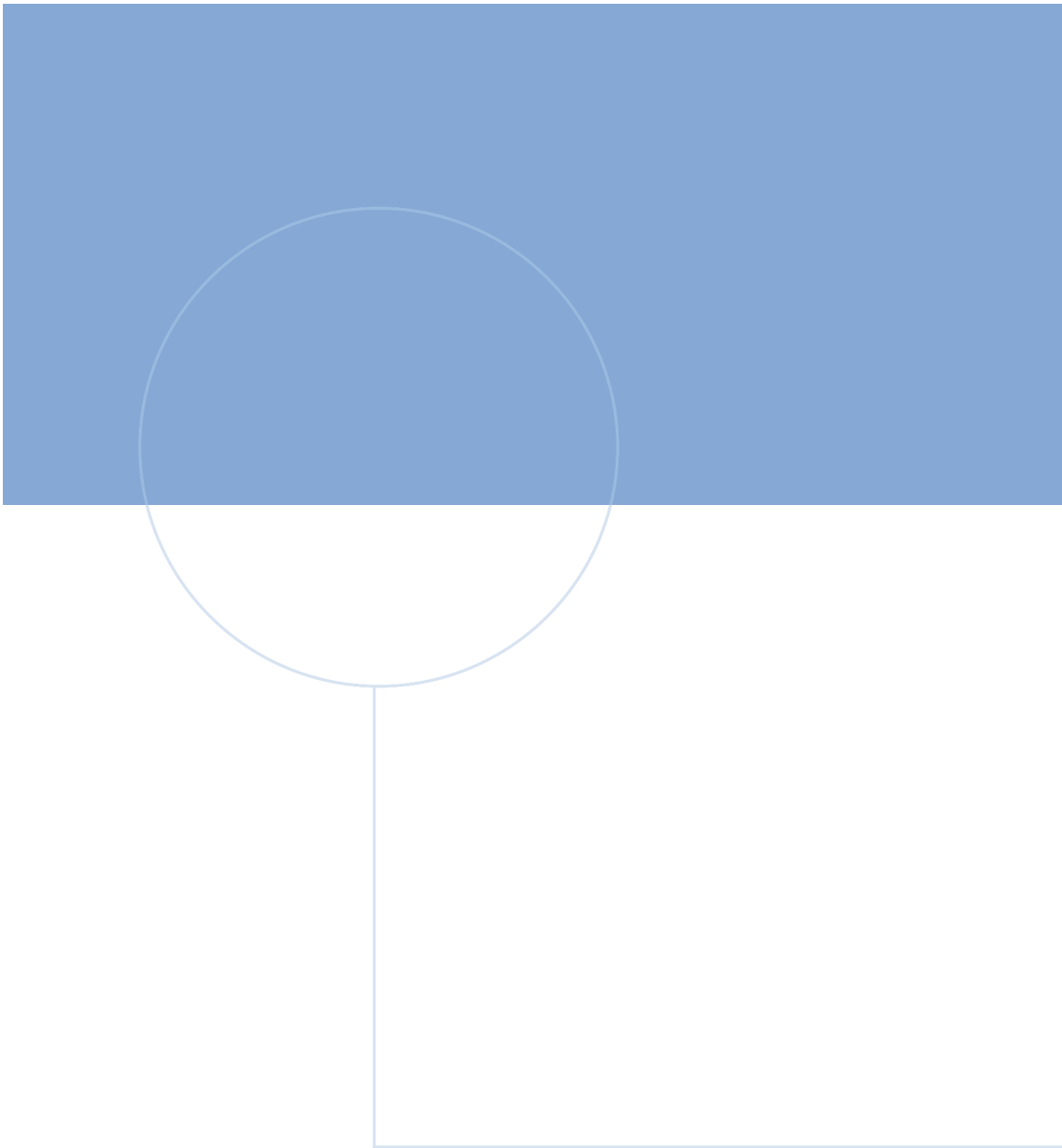
Ma, G., Wu, Z., Jia, J., & Shang, S. (2021). Safety risk factors comprehensive analysis for construction project: Combined cascading effect and machine learning approach. *Safety science*, *143*, 105410.
MacGregor, J. N. (1988). The effects of order on learning classifications by example: heuristics for finding the optimal order. *Artificial intelligence*, *34*(3), 361-370.
Madhuri, K. L., Suma, V., & Mokashi, U. M. (2018). A triangular perception of scope creep influencing the project success. *International Journal of Business Information Systems*, *27*(1), 69-85.
Magnani, M. (2004). Techniques for dealing with missing data in knowledge discovery tasks. *Obtido http://magnanim. web. cs. unibo. it/index. html*, *15*(01), 2007.
Mahdi, M. N., Mohamed Zabil, M. H., Ahmad, A. R., Ismail, R., Yusoff, Y., Cheng, L. K., Azmi, M. S. B. M., Natiq, H., & Happala Naidu, H. (2021). Software project management using machine learning technique—A Review. *Applied Sciences*, *11*(11), 5183.
Malhotra, R., & Bansal, A. J. (2014). Cross project change prediction using open source projects. 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI),
Mangasarian, O. L., & Wild, E. W. (2008). Multiple instance classification via successive linear programming. *Journal of optimization theory and applications*, *137*(3), 555-568.
Matheus, C. J., Chan, P. K., & Piatetsky-Shapiro, G. (1993). Systems for knowledge discovery in databases. *IEEE transactions on knowledge and data engineering*, *5*(6), 903-913.
McCarthy, J. (1960). *Programs with common sense*. RLE and MIT computation center Cambridge, MA, USA.
McGaughey, G., Walters, W. P., & Goldman, B. (2016). Understanding covariate shift in model performance. *F1000Research*, *5*.
McKinney, W. (2011). pandas: a foundational Python library for data analysis and statistics. *Python for high performance and scientific computing*, *14*(9), 1-9.
Memon, A. H., Rahman, I. A., Abdullah, M. R., & Azis, A. A. A. (2010). Factors affecting construction cost in Mara large construction project: perspective of project management consultant. *International Journal of Sustainable Construction Engineering and Technology*, *1*(2), 41-54.
Memon, A. H., Rahman, I. A., & Hasan, M. F. A. (2014). Significant causes and effects of variation orders in construction projects. *Research Journal of Applied Sciences, Engineering and Technology*, *7*(21), 4494-4502.
Mennis, E. A. (2006). The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations. *Business Economics*, *41*(4), 63-65.
Merino, J., Caballero, I., Rivas, B., Serrano, M., & Piattini, M. (2016). A data quality in use model for big data. *Future Generation Computer Systems*, *63*, 123-130.
Messenger, R., & Mandell, L. (1972). A modal search technique for predictive nominal scale multivariate analysis. *Journal of the American statistical association*, *67*(340), 768-772.
Michaelides, R., Bryde, D., & Ohaeri, U. (2014). Sustainability from a project management perspective: are oil and gas supply chains ready to embed sustainability in their projects?
Mir, F. A., & Pinnington, A. H. (2014). Exploring the value of project management: linking project management performance and project success. *International journal of project management*, *32*(2), 202-217.
Mochal, J. (2008). *Lessons in project management*. Apress.
Morbey, G. (2013). *Data Quality for Decision Makers: A dialog between a board member and a DQ expert*. Springer Science & Business Media.
Moselhi, O., Assem, I., & El-Rayes, K. (2005). Change orders impact on labor productivity. *Journal of construction engineering and management*, *131*(3), 354-359.

Moselhi, O., Leonard, C., & Fazio, P. (1991). Impact of change orders on construction productivity. *Canadian journal of civil engineering*, *18*(3), 484-492.
Moser, T., Winkler, D., Heindl, M., & Biffl, S. (2011). Requirements management with semantic technology: An empirical study on automated requirements categorization and conflict analysis. International Conference on Advanced Information Systems Engineering,
Mullaly, M., & Thomas, J. L. (2009). Exploring the dynamics of value and fit: Insights from project management. *Project Management Journal*, *40*(1), 124-135.
Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.
Nady, A. E., Ibrahim, A. H., & Hosny, H. (2022). Factors affecting construction project complexity. *The Egyptian International Journal of Engineering Sciences and Technology*, *37*(1), 24-33.
Naeem, M., Jamal, T., Diaz-Martinez, J., Butt, S. A., Montesano, N., Tariq, M. I., De-la-Hoz-Franco, E., & De-La-Hoz-Valdiris, E. (2022). Trends and future perspective challenges in big data. In *Advances in Intelligent Data Analysis and Applications* (pp. 309-325). Springer.
Nilsson, N. J., & Nilsson, N. J. (1998). *Artificial intelligence: a new synthesis*. Morgan Kaufmann.
Olsson, N. O. (2006). Management of flexibility in projects. *International journal of project management*, *24*(1), 66-74.
Oyewobi, L. O., Jimoh, R., Ganiyu, B. O., & Shittu, A. A. (2016). Analysis of causes and impact of variation order on educational building projects. *Journal of Facilities Management*.
Park, J. E. (2021). Schedule delays of major projects: what should we do about it? *Transport Reviews*, *41*(6), 814-832.
Pavlus, J. (2016). 8. Sight-Reading Software. *Scientific American*, *315*(6), 39-39.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., & Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830.
Peña, A. B., Castro, G. F., Alvarez, D. M. L., Alcivar, I. A. M., Núñez, G. L., Cevallos, D. S., & Santa, J. L. Z. (2019). Method for project execution control based on soft computing and machine learning. 2019 XLV Latin American Computing Conference (CLEI),
Pierre, D. A. (1986). *Optimization theory with applications*. Courier Corporation.
Pint, E. M. (1992). Price-cap versus rate-of-return regulation in a stochastic-cost model. *The RAND Journal of Economics*, 564-578.
Pinto, J. (2010). Achieving competitive advantage. *New Jercy*.
Pospieszny, P., Czarnacka-Chrobot, B., & Kobylinski, A. (2018). An effective approach for software project effort and duration estimation with machine learning algorithms. *Journal of Systems and Software*, *137*, 184-196.
Project Management, I. (2017). *A guide to the project management body of knowledge : (PMBOK guide)* (6th edition. ed.). Project Management Institute.
Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems*, *31*.
Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.
Ramazani, J., & Jergeas, G. (2015). Project managers and the journey from good to great: The benefits of investment in project management training and education. *International journal of project management*, *33*(1), 41-52.
Rasnacis, A., & Berzisa, S. (2017). Method for adaptation and implementation of agile project management methodology. *Procedia Computer Science*, *104*, 43-50.
Rebala, G., Ravi, A., & Churiwala, S. (2019). Machine learning definition and basics. In *An Introduction to Machine Learning* (pp. 1-17). Springer.
Renewables, I. (2020). Technical report, IEA, Paris, 2020b. *URL https://www. iea. org/reports/renewables-2020*.
Revay, S. O. (2003). Coping with changes. *AACE International Transactions*, CD251.

Rolstadås, A., Olsson, N., Johansen, A., & Langlo, J. (2016). Praktisk prosjektledelse: fra idé til gevinst (2. utg.). *Bergen: Fagbokforlaget*.
Rozenes, S., Vitner, G., & Spraggett, S. (2004). MPCS: Multidimensional project control system. *International journal of project management*, *22*(2), 109-118.
Rozenes, S., Vitner, G., & Spraggett, S. (2006). Project control: literature review. *Project Management Journal*, *37*(4), 5-14.
Rui, Z., Li, C., Peng, F., Ling, K., Chen, G., Zhou, X., & Chang, H. (2017). Development of industry performance metrics for offshore oil and gas project. *Journal of natural gas science and engineering*, *39*, 44-53.
Russell, J. S., Jaselskis, E. J., & Lawrence, S. P. (1997). Continuous assessment of project performance. *Journal of construction engineering and management*, *123*(1), 64-71.
Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, *8*(4), e1249.
Saglam, S. O. (2017). Applied machine learning: project management performance prediction at information technology company project management office.
Samala, R. K., Chan, H.-P., Hadjiiski, L., & Koneru, S. (2020). Hazards of data leakage in machine learning: a study on classification of breast cancer using deep neural networks. Medical Imaging 2020: Computer-Aided Diagnosis,
Samek, W., Wiegand, T., & Müller, K.-R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*.
San Cristóbal, J. R., Diaz, E., Carral, L., Fraguela, J. A., & Iglesias, G. (2019). Complexity and project management: challenges, opportunities, and future research. In (Vol. 2019): Hindawi.
Sánchez, R. Á., Iraola, A. B., Unanue, G. E., & Carlin, P. (2019). TAQIH, a tool for tabular data quality assessment and improvement in the context of health data. *Computer Methods and Programs in Biomedicine*, *181*, 104824.
Sawadogo, P., & Darmont, J. (2021). On data lake architectures and metadata management. *Journal of Intelligent Information Systems*, *56*(1), 97-120.
Schatteman, D., Herroelen, W., Van de Vonder, S., & Boone, A. (2008). Methodology for integrated risk management and proactive scheduling of construction projects. *Journal of construction engineering and management*, *134*(11), 885-893.
Schelter, S., Biessmann, F., Januschowski, T., Salinas, D., Seufert, S., & Szarvas, G. (2018). On challenges in machine learning model management.
Schoper, Y.-G., Wald, A., Ingason, H. T., & Fridgeirsson, T. V. (2018). Projectification in Western economies: A comparative study of Germany, Norway and Iceland. *International journal of project management*, *36*(1), 71-82.
Schratz, P., Muenchow, J., Iturritxa, E., Richter, J., & Brenning, A. (2019). Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data. *Ecological Modelling*, *406*, 109-120.
Schreck, B., Mallapur, S., Damle, S., James, N. J., Vohra, S., Prasad, R., Accenture, B., & Veeramachaneni, I. K. (2018). The AI Project Manager. IEEE International Conference on Big Data,
Seely, M. A., & Duong, Q. P. (2001). The dynamic baseline model for project management. *Project Management Journal*, *32*(2), 25-36.
Shafaat, A., Alinizzi, M., Mahfouz, T., & Kandil, A. (2016). Can contractors predict change orders? Investigating a historical allegation. Construction Research Congress 2016,
Sharma, N. V., & Yadav, N. S. (2021). An optimal intrusion detection system using recursive feature elimination and ensemble of classifiers. *Microprocessors and Microsystems*, *85*, 104293.
Shirazi, F., Kazemipoor, H., & Tavakkoli-Moghaddam, R. (2017). Fuzzy decision analysis for project scope change management. *Decision Science Letters*, *6*(4), 395-406.
Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, 105524.

Soltaninejad, M., Yang, G., Lambrou, T., Allinson, N., Jones, T. L., Barrick, T. R., Howe, F. A., & Ye, X. (2017). Automated brain tumour detection and segmentation using superpixel-based extremely randomized trees in FLAIR MRI. *International journal of computer assisted radiology and surgery*, *12*(2), 183-203.
Sołtysik, M., Zakrzewska, M., Sagan, A., & Jarosz, S. (2020). Assessment of project manager's competence in the context of individual competence baseline. *Education Sciences*, *10*(5), 146.
Song, D., Hong, S., Seo, J., Lee, K., & Song, Y. (2022). Correlation Analysis of Noise, Vibration, and Harshness in a Vehicle Using Driving Data Based on Big Data Analysis Technique. *Sensors*, *22*(6), 2226.
Soofi, A. A., & Awan, A. (2017). Classification techniques in machine learning: applications and issues. *Journal of Basic and Applied Sciences*, *13*, 459-465.
Strike, K., El Emam, K., & Madhavji, N. (2001). Software cost estimation with incomplete data. *IEEE Transactions on Software Engineering*, *27*(10), 890-908.
Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P., & Kawanabe, M. (2007). Direct importance estimation with model selection and its application to covariate shift adaptation. *Advances in neural information processing systems*, *20*.
Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. International conference on machine learning,
Tanaka, H. (2014). Toward project and program management paradigm in the space of complexity: a case study of mega and complex oil and gas development and infrastructure projects. *Procedia-Social and Behavioral Sciences*, *119*, 65-74.
Tavares, L. V. (2002). A review of the contribution of operational research to project management. *European Journal of Operational Research*, *136*(1), 1-18.
Taylor, J. (2008). *Project scheduling and cost control: planning, monitoring and controlling the baseline*. J. Ross Publishing.
Tereso, A., Ribeiro, P., Fernandes, G., Loureiro, I., & Ferreira, M. (2019). Project management practices in private organizations. *Project Management Journal*, *50*(1), 6-22.
Tetlock, P. E. (2009). Expert political judgment. In *Expert Political Judgment*. Princeton University Press.
Thomas, H. R., & Završki, I. (1999). Construction baseline productivity: Theory and practice. *Journal of construction engineering and management*, *125*(5), 295-303.
Tjoa, E., & Guan, C. (2020). A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE transactions on neural networks and learning systems*, *32*(11), 4793-4813.
Tu, W. (2006). Zero-inflated data. *Encyclopedia of environmetrics*.
Tu, W., & Sun, S. (2012). Cross-domain representation-learning framework with combination of class-separate and domain-merge objectives. Proceedings of the 1st International Workshop on Cross Domain Knowledge Discovery in Web and Social Network Mining,
Tumin, V. M., Kharlanov, A. S., Zenkina, E. V., Kostromin, P. A., & Trifonov, V. A. (2022). Digital Transformation of Oil and Gas Sector during Transition to Renewable Energy Sources. International Scientific and Practical Conference Strategy of Development of Regional Ecosystems "Education-Science-Industry"(ISPCR 2021),
Vahdani, B., Mousavi, S. M., Hashemi, H., Mousakhani, M., & Ebrahimnejad, S. (2014). A new hybrid model based on least squares support vector machine for project selection problem in construction industry. *Arabian Journal for Science and Engineering*, *39*(5), 4301-4314.
van Niekerk, J., Wium, J., & de Koker, N. (2022). The value of data from construction project site meeting minutes in predicting project duration. *Built Environment Project and Asset Management*.
van Veen-Dirks, P., & Wijn, M. (2002). Strategic control: meshing critical success factors with the balanced scorecard. *Long range planning*, *35*(4), 407-427.
Walker, A. (2015). *Project management in construction*. John Wiley & Sons.

Wanasinghe, T. R., Wroblewski, L., Petersen, B. K., Gosine, R. G., James, L. A., De Silva, O., Mann, G. K., & Warrian, P. J. (2020). Digital twin for the oil and gas industry: Overview, research trends, opportunities, and challenges. *IEEE access*, *8*, 104175-104197.
Wang, P. (2006). *Rigid flexibility: the logic of intelligence* (Vol. 34). Springer Science & Business Media.
Wang, Q., Ma, Y., Zhao, K., & Tian, Y. (2022). A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, *9*(2), 187-212.
Wang, S., Cao, J., & Yu, P. (2020). Deep learning for spatio-temporal data mining: A survey. *IEEE transactions on knowledge and data engineering*.
Wang, T., Tang, W., Du, L., Duffield, C. F., & Wei, Y. (2016). Relationships among risk management, partnering, and contractor capability in international EPC project delivery. *Journal of management in engineering*, *32*(6), 04016017.
Wang, Z., & Bovik, A. C. (2009). Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE signal processing magazine*, *26*(1), 98-117.
Warburton, R., & Kanabar, V. (2008). The practical calculation of schedule variance in terms of schedule. PMI® Global Congress,
Wauters, M., & Vanhoucke, M. (2016). A comparative study of Artificial Intelligence methods for project duration forecasting. *Expert systems with applications*, *46*, 249-261.
Webster, F. M. (1994). *The WBS*. PM Network. Retrieved 22.03.2022 from https://www.pmi.org/learning/library/work-breakdown-structure-basic-principles-4883
Weinberger, D. (2019). *Everyday chaos: Technology, complexity, and how we're thriving in a new world of possibility*. Harvard Business Press.
Wen, J., Li, S., Lin, Z., Hu, Y., & Huang, C. (2012). Systematic literature review of machine learning based software development effort estimation models. *Information and Software Technology*, *54*(1), 41-59.
Wilson, H. J., Daugherty, P. R., & Davenport, C. (2019). The future of AI will be about less data, not more. *Harvard Business Review*, *14*.
Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a standard process model for data mining. Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining,
Yang, G.-Z., Andreu-Perez, J., Hu, X., & Thiemjarus, S. (2014). Multi-sensor fusion. In *Body sensor networks* (pp. 301-354). Springer.
Zaheer, R., & Shaziya, H. (2019). A study of the optimization algorithms in deep learning. 2019 Third International Conference on Inventive Systems and Control (ICISC),
Zang, H., Xu, R., Cheng, L., Ding, T., Liu, L., Wei, Z., & Sun, G. (2021). Residential load forecasting based on LSTM fusing self-attention mechanism with pooling. *Energy*, *229*, 120682.
Zhao, Z.-Y., Lv, Q.-L., & You, W.-Y. (2008). Applying dependency structure matrix and Monte Carlo simulation to predict change in construction project. 2008 International Conference on Machine Learning and Cybernetics,
Zhou, Z.-H. (2012). *Ensemble methods: foundations and algorithms*. CRC press.
Zou, Y., & Lee, S. H. (2008). The impacts of change management practices on project change cost performance. *Construction Management and Economics*, *26*(4), 387-393.



NTNU

Norwegian University of
Science and Technology