Master's thesis

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

David Kabichev Kvale

Deep Learning in Construction Safety: Quality Assessment, Hazard Identification, and Preventive Measure Proposals in Job Safety Analysis

Master's thesis in Engineering and ICT
Supervisor: Nils Olsson

December 2023

NTNU
Norwegian University of
Science and Technology

David Kabichev Kvale

Deep Learning in Construction Safety: Quality Assessment, Hazard Identification, and Preventive Measure Proposals in Job Safety Analysis

Master's thesis in Engineering and ICT
Supervisor: Nils Olsson
December 2023

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, a part of the subject TPK4920 - Project and Quality Management (Master's Thesis), is written as a master's thesis at the Norwegian University of Science and Technology, counting for 30 credits. The work is conducted during the autumn semester of 2023 at NTNU's Faculty of Engineering and marks the end of my Master's program in Engineering and ICT. Last semester I wrote the thesis *Utilizing Machine Learning and Text Data for Predictive Safety Identification of High-Risk Construction Projects* as a part of my specialization project. This thesis therefore shares structural similarities to the preliminary report.

Over the past years, I have combined subjects within production management and artificial intelligence. In the summer of 2021, I was involved in working within the data field for a company, which sparked an interest in the field. The following year on exchange in Lisbon, I enrolled in all classes possible within Artificial Intelligence and Data Science. Returning to NTNU, I continued taking courses within machine learning, leading to the selection of the project thesis called *Artificial Intelligence in Projects.*

Within the project thesis, I chose to write for *Sustainable value creation by digital predictions of safety performance in the construction industry (DiSCo).* This choice was made because construction safety seemed like an interesting application of artificial intelligence, and the availability of datasets that presented interesting opportunities. Initially, my knowledge of Artificial Intelligence greatly surpassed my knowledge of construction safety, but through writing this thesis I have tried to expand my knowledge of construction safety. I hope this thesis inspires the construction industry to explore new technologies, hopefully enhancing safety and reducing injuries within the industry.

While writing this thesis in English, there have been certain challenges due to the textual data used being Norwegian. Because this thesis is written in English the textual data has mostly been translated into English. Exceptions to this are in Figures 16 and 22, which are in Norwegian due to the content of the most common words and featuring a word cloud. The translation process has involved the use of many different online tools, followed by a manual review. Translating construction-specific words has been especially challenging. Therefore, the original Norwegian text is included in the Appendix.

Acknowledgement

I would like to thank my supervisor, Nils Olsson. His guidance and valuable feedback throughout the semester have been extremely helpful, and I am very grateful for having him as my supervisor.

Furthermore, I would like to thank the DiSCo project team. Their work has intrigued me, and given me a great opportunity to learn and develop my skills. I am particularly grateful for granting me access to the dataset utilized in this research.

There are many people I would like to thank for helping me on my five-year journey at NTNU. In middle and high school, some teachers inspired me and gave me the confidence to pursue a degree in STEM. Friends who spent evenings teaching me math while in the military, helping me to fulfill NTNU's math prerequisites. People in the corporate world giving me a chance and opportunity to learn. I am very grateful for the help I have gotten along this journey, I would not be here today without it.

Lastly, I would like to thank my family, and especially my mother, who has been a source of support and a great conversational partner.

David Kabichev Kvale

David Kabichev Kvale, December 2023

Abstract

This research explores the application of machine learning and deep learning for enhancing job safety analysis in the construction industry, known for its high accident rates. The research conducted utilizes job safety analysis data spanning from 2019 to 2023, gathered from a major European construction company's Norwegian division. The study proposes an artificial intelligence-driven approach to improve safety measures.

For this purpose, three distinct machine learning algorithms are developed. The first algorithm is designed to assess the quality of job safety analysis, using a dataset of previously evaluated safety analysis for training. The second algorithm utilizes multi-label classification techniques to identify potential hazards based on textual descriptions of activities and their activity types. The third algorithm uses a large language model, trained on a dataset with activity descriptions, their associated hazards, and identified measures, to generate new preventive measures.

The results of this study are promising, and indicate potential for artificial intelligence in construction safety. The quality assessment algorithm shows that machine learning can evaluate the quality of job safety analysis, even though the performance is heavily dependent on the quality of the training data. The hazard identification algorithm displays a good capability of classifying various hazards, potentially detecting hazards missed by human analysis. Finally, the generative artificial intelligence model for preventive measures suggests relevant and practical measures, though these tend to be less detailed compared to human-written measures. Future work involves improving the machine learning models employed and exploring how machine learning tools can be integrated into the job safety analysis workflow.

Sammendrag

Denne studien utforsker hvordan maskinlæring og dyplæring kan brukes til å forbedre sikker jobbanalyse i byggebransjen, som er kjent for høye ulykkesrater. Forskningen er basert på data fra 2019 til 2023, samlet inn fra den norske avdelingen av et stort europeisk bygg- og anleggsselskap. Studien foreslår en kunstig intelligensdrevet tilnærming for å forbedre sikkerhetstiltak.

For dette formålet er tre forskjellige maskinlæringsalgoritmer utviklet. Den første algoritmen er utviklet for å vurdere kvaliteten på sikker jobbanalyser ved å bruke et datasett av tidligere evaluerte sikkerhetsanalyser for opplæring. Den andre algoritmen benytter en klassifiseringsteknikk med flere etiketter for å identifisere potensielle farer basert på tekstbeskrivelser av aktiviteter og deres aktivitetstyper. Den tredje algoritmen bruker en stor språkmodell, som er trent på et datasett med aktivitetsbeskrivelser, deres tilhørende farer og identifiserte tiltak, for å generere nye forebyggende tiltak.

Resultatene fra denne studien er lovende og viser at kunstig intelligens har et potensial innen sikkerhet i bygg- og anleggsbransjen. Kvalitetsvurderingsalgoritmen viser at maskinlæring kan evaluere kvaliteten på sikker jobbanalyser, selv om resultatene er sterkt avhengig av kvaliteten på treningsdataene. Fareidentifiseringsalgoritmen viser god evne til å klassifisere ulike farer, og kan potensielt detektere farer som menneskelige analyser ikke har fanget opp. Til slutt foreslår den generative modellen for forebyggende tiltak relevante og praktiske tiltak, selv om disse har en tendens til å være mindre detaljerte sammenlignet med tiltak skrevet av mennesker. Fremtidig arbeid innebærer å forbedre maskinlæringsmodellene som brukes, og utforske hvordan maskinlæringsverktøy kan integreres i sikkerjobbanalyseprosessen.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence.

AUC Area Under the Curve.

DL Deep Learning.

EDA Exploratory Data Analysis.

FK Foreign Key.

FN False Negative.

FP False Positive.

FPR False Positive Rate.

GB Gradient Boosting.

GDPR General Data Protection Regulation.

HSE Health, Safety, and Environment.

IC Inclusion Criteria.

IDF Inverse Document Frequency.

JHA Job Hazard Analysis.

JSA Job Safety Analysis.

LLM Large Language Model.

LSTM Long short-term memory.

ML Machine Learning.

NaN Not a Number.

NLP Natural Language Processing.

NTNU Norwegian University of Science and Technology.

PK Primary Key.

PR Precision-Recall.

RF Random Forest.

RNN Recurrent Neural Network.

ROC Receiver Operator Characteristic.

RQ Research Question.

SMOTE Synthetic Minority Over-sampling.

SOTA State-of-the-art.

TF Term Frequency.

TN True Negative.

TP True Positive.

TPR True Positive Rate.

1 Introduction

This thesis looks into how utilization of Machine Learning (ML) and Deep Learning (DL) techniques can be used in the process of performing a Job Safety Analysis, referred to as a JSA. More specifically, the research looks into how ML can be used to measure the quality of a JSA, identify hazards, and generate preventive measures.

1.1 Background and Motivation

Globally, the construction industry is widely recognized for being among the most dangerous industries (Pinto et al., 2011). The nature of construction work, often working at heights, handling heavy materials, and operating complex machinery, contributes to these risks (Bhole, 2016). This holds true for Norway as well, where the construction industry is one of the most accident-prone industries (Mostue et al., 2022). In the last decade, there has been an average of eight fatalities each year within the construction industry in Norway (Isachsen Berntsen, n.d.(a)). Therefore, there is a need for innovative approaches to enhance the safety in construction projects.

Recently, Artificial Intelligence (AI) has been rapidly developing, reshaping various industries, and changing daily life for many (Nawaz, 2023). These rapid advancements are contributed by improvements in ML and DL techniques, and increased computational capabilities (Saha et al., 2021). AI's ability to understand patterns in complex data has made it a tool that can be utilized for innovative problem-solving across various industries (Huang et al., 2020, Huntingford et al., 2019).

The construction industry is one of the least digitized industries in the world (Abioye et al., 2021). Abioye et al., 2021 argues that the lack of digitization has caused delays, bad performance in terms of quality and productivity, poor decision-making, and issues related to health and safety. AI presents a unique opportunity to change this industry. Within safety in construction, AI can be used for predictive analysis, automated hazard identification, and performing risk-assessments in real time (Abioye et al., 2021). These methods can proactively address safety issues, reducing accident rates. NTNU created DiSCo, Sustainable value creation by digital predictions of safety performance in the construction industries, to address safety issues in the construction industry (NTNU, 2023). The research project aims to do this by using AI in early phases of construction projects to predict future safety performance.

1.2 Research Questions

The primary long-term aim of this research is to enhance safety performance in the construction industry. This objective is explored through three specific Research Questions (RQs), each exploring the application of ML using data from JSA. The research questions are as follows:

RQ1: How can machine learning methodologies assess the quality of a job safety analysis?

RQ2: How can machine learning detect potential hazards during activities in construction projects?

RQ3: How can generative AI propose preventive measures for identified hazards?

To address these questions, three unique ML algorithms have been developed. The first RQ is addressed by utilizing a variable within the JSA dataset named "Good Example", which indicates whether a JSA is exemplar. The second RQ involves the use of a multi-label classification of activity descriptions to identify possible hazards. The last RQ is explored by fine-tuning a generative AI model using activity descriptions with their associated hazards, and the preventive measures taken to address them.

1.3 Project Scope

The data for this research is collected from a large European construction company, specifically its Norwegian division. The dataset consists of JSA records collected from 2019 to 2023. The focus of the research is to explore how JSA can be utilized with modern AI methods to enhance their effectiveness and improve safety performance. Even though the data is from the construction industry in Norway, similar approaches could be applied universally and in other industries concerned with safety and injury prevention.

1.4 Structure of Thesis

Section 1 offers an introduction to the research, explaining the background and motivation of the research, followed by a presentation of the RQs and the scope of the project. Section 2 delves into the theoretical background of the research. The initial subsection discusses health and safety in the construction industry, before examining how ML can be utilized to address safety concerns in the construction industry. Further subsections explore more technical aspects, Natural Language Processing (NLP) is covered in subsection 2.3, ML techniques are explored in subsection 2.4, and DL is examined in subsection 2.5. The methodology employed in the research is described in section 3. This includes a literature review, Exploratory Data Analysis (EDA), data cleaning, and the ML algorithms developed in the research. In section 4, the results of the three ML algorithms developed in the thesis are presented, showing the empirical findings of the study. In section 5 the research is discussed. The results are discussed in addition to the limitations, and practical applications of the study. Lastly, in section 6 the research is concluded, stating the findings and suggesting further work.

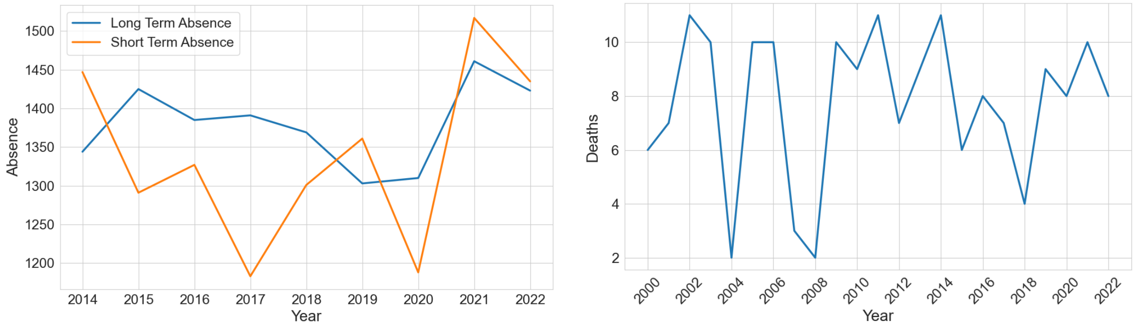
2 Theory

This chapter delves into the theory that forms the foundation of this thesis. It explores concepts and frameworks that are important for this study, such as safety in the construction industry, the use of ML in the construction industry, and principles of NLP. Furthermore, it examines the field of ML and DL. This chapter serves as the theoretical basis for the research and development of this thesis.

2.1 Health, Safety and Environment in the Construction Industry

Health Safety and Environment referred to as HSE, focuses on the well-being of workers. “Health” is concerned with the preservation of workers physical and mental well-being against illness. Meanwhile, “Safety” deals with the protection of physical harm, as defined by Hughes and Ferrett, 2012. The construction industry in Europe has reported the highest rate of fatal work-related accidents in 2014, and it is widely recognized as one of the most dangerous industries in the world (Winge, Albrechtsen and Mostue, 2019, Lingard and Rowlinson, 2004). As a result of this, there has been a notable rise in HSE-related publications since the early 2010s (Jin et al., 2019).

Figure 1 presents the accidents in Norway within the construction industry, distinguishing between fatal accidents, and accidents with Long-term and Short-term absence. Absence that has lasted for longer than three days is in this case defined as an injury that has caused long-term absence (Isachsen Berntsen, n.d.(a), Isachsen Berntsen, n.d.(b)).



(a) Annual Accidents Causing Long/Short-Term Absence (Isachsen Berntsen, n.d.(b))

(b) Construction Industry Fatalities per Year (Isachsen Berntsen, n.d.(a))

Figure 1: Injury and Fatality Trends in the Norwegian Construction Industry

The data presented in Table 1 are from the same dataset as Figure 1. The data reveals that annually more than one percent of workers in the construction industry in Norway experience either a short-term or long-term injury. These numbers are solely based on reported incidents. Research indicates that the actual numbers might be higher, as some accidents are unreported, representing “dark figures” (Albrechtsen, 2012).

Type of absence	Accidents per 1000 workers
Short-Term	5.3
Long-Term	5.4

Table 1: Annual Accident Rates per 1000 Workers Categorized by Absence Type (2014-2022) (Isachsen Berntsen, n.d.(b))

2.1.1 Hazards in Construction Projects

Hazard recognition is fundamental to effective safety management in the construction industry (Holt, 2008). When hazards go unnoticed, they remain unaddressed, posing a threat to worker safety (Carter and Smith, 2006). Research suggests that due to the dynamic nature of construction sites workers may fail to recognize up to 57% of safety hazards (Albert et al., 2017). Furthermore, the ability of workers to identify some hazards may be higher than for others. For instance, workers were found to be better at identifying "being struck by" accidents compared to "chemical exposure" accidents (S. Uddin et al., 2020).

Table 2 shows accident types within the construction industry in Norway from 2015 to 2022. It reveals that "falls" and "struck-by" accidents are the most common types of accidents. A study by the United States Department of Labor reports similar findings, with "fall", "struck-by" and "caught-between" accidents being the most common (Safety and Administration, 2011). This indicates a global pattern in the most common site hazards within the construction industry.

In construction safety management, identifying hazards is an important part of preventing accidents and enhancing safety performance. A study by Winge and Albrechtsen, 2018 identified frequent accident types and found the barriers and consequences associated with these incidents in the Norwegian construction industry. This study shows an essential first step in accident prevention, which is recognizing the various hazards that could potentially occur on construction sites.

The dynamic nature of construction projects adds another layer to the complexity of hazard identification. Factors such as weather conditions, frequent work team rotations, and changing construction site topography over time, are factors that add to construction site complexity (Bobick, 2004). As construction sites are changing all the time, there is a need to use strategies to identify hazards that are flexible, to identify and mitigate risk (Rozenfeld et al., 2010).

2.1.2 Safety Indicators

Safety performance indicators measure an organization's capability to manage and mitigate the risk of accidents (Kjellen and Albrechtsen, 2017). These indicators provide a quantitative method of assessing the safety performance over time, enabling the creation of targets for continuous improvement (Herrera, 2012). When evaluating safety performance, it is common to distinguish between two types of indicators; lagging and leading indicators.

Type of Accident	Number of Accidents	% of Accidents
Fall	4662	21.51
Struck by object	4330	19.98
Puncture/cut by sharp/pointed object	2806	12.95
Electric shock	1387	6.40
Crushed/trapped	1300	5.99
Collision/impact	533	2.45
Overturn	402	1.86
Chemicals	257	1.19
Threats of violence	150	0.69
High/low temperature	100	0.46
Explosion, blast, fire	90	0.41
Inflicted injury by violence	37	0.17
Other	2264	10.45
Unknown	3352	15.47

Table 2: Categorization of Construction Accident Types in Norway (2015-2022) (Isachsen Berntsen, n.d.(c))

Lagging indicators look at changes that have already occurred, and are considered reactive. In economics, a lagging indicator is a measure that changes after the economy has changed (Manuele, 2009). Within the construction industry, examples of lagging indicators include incident and fatality rates (Hinze et al., 2013). These rates are often calculated per hour worked, to normalize the metric relative to the size of the project and workforce.

Leading indicators, on the other hand, are used to predict future trends, and are therefore deemed as proactive (Stock and Watson, 2008). In construction, the indicator is forward-looking, expressing the future safety performance. Examples of leading indicators can be the quality and amount of training provided to workers, the thoroughness of hazard, and measures analysis (Hinze et al., 2013, Alruqi and Hallowell, 2019).

There are many ways of measuring safety performance in a construction project, relying on a single metric can be insufficient. Therefore, Kjellen and Albrechtsen, 2017, suggests combining several indicators to measure and understand safety performance within construction projects. This approach creates a more nuanced understanding of workplace safety, and helps in decision-making.

2.1.3 Job Safety Analysis

SIBA (Safety Management in Construction) defines JSA, also known as Job Hazard Analysis (JHA), as a review and assessment of possible hazards before performing an activity where dangerous situations can arise. The aim is to assess if the safety is addressed well enough with the current procedures and plans, or if there is a need to implement additional measures, to eliminate and control the hazards identified (Tinmannsvik et al., 2016).

JSA focuses on the relationship between four elements: the worker, the task at hand, the tools and the equipment being used, and the overall work environment (Chao and Henshaw, 2002). Performing a JSA involves looking into every step of an activity to identify potential hazards. When a hazard is identified the process continues by suggesting and implementing appropriate safety rules and procedures to address the hazards. This reduces the risk of accidents and injuries during the execution of the activity.

Hazards are defined as conditions or activities that can lead to undesirable events, leading to injuries to individuals, harm to the environment, or damage to materials or property. Examples of hazards can be found in Table 2. Table 2 provides an overview of hazards and their corresponding frequency in the Norwegian construction industry. Understanding the hazards is a crucial step for developing effective safety measures to both reduce the frequency of accidents and minimize their impact when occurring.

Ideally, most risks should be identified and mitigated during the planning stage of a project. However, due to the dynamic and unpredictable nature of construction projects, there is often a need for re-evaluating safety in response to unexpected developments or changes in plans. In this situation, JSA offers a proactive tool in hazard management, that can be a valuable tool for re-evaluating safety measures (Tinmannsvik et al., 2016). Figure 2 shows at which stage of the project JSA can be used, and the amount of risk it potentially can reduced. As seen in Figure JSA is used right before the execution stage.

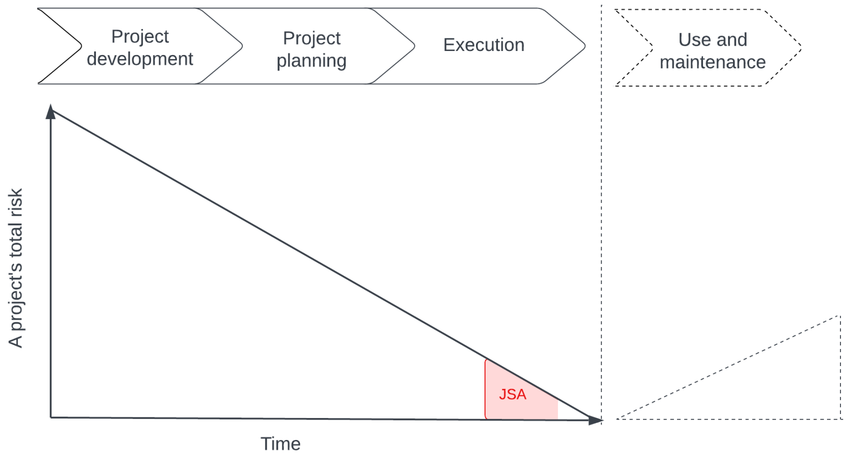


Figure 2: Impact of JSA on Project Risk Over Time - Author's own illustration - based on (Svensli and Solberg, 2016)

Various papers suggest slightly different approaches to conduction a JSA. Despite some variations, the fundamental essence of the JSA remains consistent throughout the literature, centering on identifying and mitigating hazards associated with a specific task. The JSA that is presented in this thesis is created by SIBA, and includes the following steps (Tinmannsvik et al., 2016):

1. Assess the need for a JSA: According to Kjellen and Albrechtsen, 2017, a JSA should be performed in scenarios such as; activities involving uncontrolled hazards, new tasks, deviations from standard procedures, unfamiliar teams or equipment, and changing conditions such as weather or operational changes.

2. Preparation and Planning: Tinmannsvik et al., 2016 suggests a designated JSA manager that oversees the gathering of relevant data such as work procedures, manuals, previous JSAs, and forming the JSA team. The JSA team should include all activity participants, a safety representative, a team leader, and experts in the specific area (Rausand, 2013; Roughton and Crutchfield, 2013). The JSA manager should be responsible for scheduling and documenting the JSA meeting, and ensuring that all preventative measures are performed.

3. Conduct the JSA: Conducted right before the activity, the JSA should be conducted and everyone involved or effected by the activity should have the possibility to attend the JSA (Tinmannsvik et al., 2016). When performing the JSA the following steps should be carried out (Albrechtsen et al., 2019):

1. Decomposition of job: breaking down the job into functions, tasks, and steps. The steps are listed and described in order.
2. Hazard identification: Potential events and conditions that can lead to dangerous situations are identified for each sub-task identified in 1).
3. Potential consequences of the hazards identified in 2) are assessed.
4. Expected frequency of occurrence for the hazards identified in 2) are assessed.
5. An assessment of the risk for each sub-task is performed based on the assessed frequency and consequence, which is, in turn, assessed in relation to a risk matrix.
6. Risk reduction measures that can help to improve safety for performing the work is identified for those sub-tasks that have an intolerable risk.

4. Implementation of measures and execution of work After the JSA, the identified safety measures are implemented, and the planned activity is executed safely.

5. Summary and Learning Points: The JSA manager is responsible for documenting the insights and learning's from the JSA process, contributing to continuous safety improvements.

JSA is a process that relies heavily on the input and expertise of the team involved in performing the JSA. Every measure should have a responsible person to make sure that the measure is acted upon. Researchers have looked into the effectiveness of using JSA as a safety tool. Two studies, Van Derlyke et al., 2022 and Halim et al., 2018, found that properly implementing JSA contributes to reducing accidents. In addition to these findings, Albrechtsen et al., 2019 identified six benefits from the use of JSA in safety management:

1. Formalization of Work
2. Accountability
3. Participation of employees
4. Organizational learning
5. Hazard identification and situation awareness
6. Loss Prevention

Regulations

In the Norwegian construction industry, JSA, is a widely used tool, however it is not mandated by law, nor mentioned in Norwegian legislation (Svensli and Solberg, 2016). Nonetheless, JSA can still help to satisfy certain legal requirements.

Paragraph § 3-2 (3) of the Working Environment Act requires a written instruction for tasks that pose an increased risk to the safety and health of workers. This law emphasized the need for guidelines on how to safely execute tasks with a high risk of injury, with the demand for necessary safety measures being implemented (Arbeidstilsynet, 2006).

Paragraph § 10-4 of the Regulation on the Performance of Work requires the workers to receive adequate training for using specific equipment. The training should ensure that the workers can use the equipment safely. This law requires the details of the training to be documented in writing (Arbeidstilsynet, 2016). Both laws are translated and written below, the original laws in Norwegian text can be found in Appendix A.

§ 3-2. Specific Precautions to Ensure Safety

> **(3)** If work is to be carried out that may pose a particular risk to life or health, a written instruction must be prepared on how the work is to be performed and what safety measures should be implemented. (Arbeidstilsynet, 2006).

§ 10-4. Requirements for Equipment-Specific Training

> The employer must ensure that the employee receives the necessary training on the specific work equipment they will use. The training must be adapted to the nature of the work equipment and ensure that the employee can use the work equipment in a safe manner. It must be documented in writing which work equipment training has been provided for, who provided the training, and who has received the training (Arbeidstilsynet, 2016).

2.1.4 Using data within HSE in Constructions Projects

Within HSE in construction projects, effective utilization of data is important. Research conducted by Andreassen et al., 2020 mapped all the data collected from clients and contractors within the construction industry. The primary objective was to get a better understanding of how data is currently being utilized in the sector and to identify which data holds potential for value creation. For this purpose, 30 different data types were ranked by its potential to be a part of a leading safety indicator. The research highlighted the number of active JSAs as an effective metric due to its simplicity and insightful nature.

The study by Andreassen et al., 2020 also investigated the data storage practices of the companies. It discovered that a wide array of software solutions, 26 in total, were used for data collection and data storage. Despite the many different software solutions, the nature of the data stored was found to be similar across companies. These findings suggest a lack of standardization in software solutions and data structuring in the construction industry. This makes the data hard to use across companies and software solutions. The common goal of these software solutions is to better understand a company's HSE performance. The study found that the software solution lacked the capabilities to visualize and analyse the data in order to extract the desired information. This makes it hard to extract meaningful insights from the collected information.

The study concludes that a substantial amount of data is collected by companies, but the data is not utilized properly (Andreassen et al., 2020). In addition, the study finds that there are no industry standards on how to store data, which makes it hard to collect data from multiple sources and do a comprehensive analysis of data from multiple sources. These findings underline a need for a structured and standardized approach for data management in the construction industry. In addition to easier ways to display and analyse the data to make it easier to utilize.

2.2 Machine Learning in Construction Safety

This chapter focuses on incorporating ML into construction safety. Recently, there has been collected a substantial amount of data regarding project operations, injury records, and preventive measures. Such extensive data opens up for new and innovative approaches to enhance safety performance in construction projects.

Several studies have been conducted trying to utilize ML to forecast injury outcomes within construction projects. Research conducted by Alkaissy et al., 2023 and Tixier et al., 2016 have focused on predicting the consequences of an injury post-occurrence, utilizing data related to the accident. These studies aim to predict various aspects of the injury, including the type of injury, the affected body part, the injury's severity, and the type of energy involved in the accident. Furthermore, Marucci-Wellman et al., 2017 extends this approach by classifying injury narratives using text data.

Several studies have looked into the integration of JSA with ML to enhance safety in the construction industry. A paper, *Ontology-based semantic modeling of construc-*

tion safety knowledge: Towards automated safety planning for job hazard analysis (JHA), by S. Zhang et al., 2015, proposes to utilize advanced technologies such as Building Information Modeling (BIM), Virtual Design and construction technology, and Geographic Information Systems (GIS), to enhance hazard identification and safety planning. These technologies are suggested to detect, visualize, and mitigate safety hazards. The paper argues these tools should be used to assist human decision-making, and suggests involvement of safety experts in reviewing and audit the outcome produced by these automated systems.

Trying to use Information and Communication Technology (ICT) and prior experience to help identify safety hazards, Hadikusumo and Rowlinson, 2004 developed the DFSP-database. The database is a comprehensive collection of potential safety hazards and corresponding preventive measures, trying to facilitate hazard and preventive measure identification. Since the process of creating a JSA is complex and time-consuming (S. Zhang et al., 2015), using previously identified hazards and measures can facilitate identifying hazards and measures in upcoming activities.

A particularly interesting paper, by Chi et al., 2014, delves into how text classification potentially can be utilized to enhance JSA. The research used text data, including activity descriptions and hazards, and classified them. The study utilized ontology representation to connect the hazards with the corresponding measures. The research, conducted in 2014, showed limitations in performance measured with precision and recall. The ML methods employed in this paper are nearly a decade later considered as simple and less advanced compared to state-of-the-art (SOTA) techniques.

Several studies have been conducted trying to use visualizations and animations to identify potential hazardous zones in construction sites. Bansal, 2011 implemented a GIS-based 3D animation in safety planning, identifying areas and activities with an increased risk of accidents. On the other hand, Kiviniemi et al., 2011 developed a method for conducting a JSA using a virtual construction site model, using a simulated environment to enhance the understanding of potential hazards. Lin et al., 2011 developed a 3D video game where players, acting as safety inspectors, pass through a virtual construction site identifying potential hazards, enhancing their hazard recognition skills.

Two papers by Poh et al., 2018 and Jafari et al., 2019 explored how ML can be utilized as a leading safety indicator in construction projects. Poh et al., 2018, trained ML models to predict the likelihood and severity of accidents using data from construction sites. The researchers found that the models, particularly Random Forest (RF), showed promising results in predicting accidents. A limitation of using ML to predict accidents highlighted by the researchers, is the model's "black-box" nature. This makes it hard for humans to interpret the results and makes it difficult to get a deeper understanding of how specific factors contribute to the occurrence of accidents. Jafari et al., 2019 used ML with construction data to develop a leading safety indicator. The research has promising results and found that 10 of 23 data points effectively could indicate safety performance in a construction project. These studies show the potential of using ML to enhance safety management in construction projects.

Some studies have explored the potential generative AI has to be utilized within construction safety. A study by, Rane, 2023, highlights the potential Generative AI has in doing predictive analysis. According to the study, utilizing historical data, including past accidents and safety measures, an AI model can identify patterns and trends within construction projects. By doing so, the AI system can identify potential safety hazards. S. J. Uddin, 2023 argues the utilization of generative AI can enhance construction safety education and training. In the study, students aspiring to become construction engineers were assessed on their ability to recognize hazards. Following this evaluation, the students were trained using ChatGPT to assist in hazard identification. A final assessment of their hazard identification skills was conducted. The results showed an improvement, suggesting that ChatGPT can be utilized in safety education to enhance hazard recognition skills.

Despite there being some research on the application of generative AI in construction safety, the domain seems mostly unexplored. Every study found on the application of generative AI in construction safety was conducted in 2023, suggesting it is a new and emerging field of study. This presents opportunities for further research of Large Language Models (LLMs) used within construction safety.

2.3 Natural Language Processing

Natural Language Processing, commonly referred to as NLP, is the subfield at the intersection of computer science and linguistics. It seeks to enable computers to comprehend and interpret human language (Chowdhary, 2020). Essentially, NLP wishes to bridge the gap between human language and computer language by helping with efficient information exchange between the two. By using NLP, computers can process, analyse, and even generate human language in ways that are meaningful, this can be enabled in a wide range of applications such as text translations, and text-to-text generation.

This chapter looks into the NLP techniques employed in this thesis, mainly focusing on NLPs role in converting human language into formats that can be quantitatively used by computers. Through different tools in NLP, such as tokenization, normalization, and vectorization, it tries to extract meaning and patterns from the unstructured text data. Therefore NLP techniques offer important tools to help computers interpret and make sense of complexities within the human language. While this chapter explores various aspects of preprocessing in NLP, it is worth noting that the application of ML techniques such as Long Short-Term Memory (LSTM) networks and Transformers using text data will be covered separately in the subsection 2.5.

2.3.1 Tokenization

Tokenization is a fundamental step in NLP, involving the segmentation of text into smaller entities known as 'tokens'. Typically, a piece of text is split into individual words, using spaces and punctuation as the separators. These tokens work as the basic building blocks for NLP models (Hassler and Fliedl, 2006). Once the text is tokenized, it becomes more manageable for computers, and can more easily be processed using different NLP techniques (Jurafsky and Martin, 2007). In Table 3 an example of a sentence broken down into individual tokens can be seen, where each token represents a word.

Text	**Tokenization**
The builders are using bricks.	[“The”, “builders”, “are”, “using”, “bricks”]

Table 3: Example of Tokenization in NLP

2.3.2 Stopword Removal

Stopword removal is a technique within NLP that seeks to filter out words, commonly used, but typically add little to the overall meaning of a sentence. These frequently occurring words, known as stopwords, are often removed to help textual analysis, as they might not give semantic value (Jurafsky and Martin, 2007). The Natural Language Toolkit, a widely used tool in NLP, identified about 127 stopwords in the English language, including words like “I”, “should”, “is” and “of” (NLTK Contributors, n.d.).

The main purpose of using stopword removal is to simplify the textual data by reducing its dimensionality. This reduction can enhance computational performance (Yogish et al., 2019), and in some cases improve the precision of ML models by eliminating noise and unnecessary data (Haddi et al., 2013). By removing words with minimal semantic contribution, stopword removal can possibly make ML using text data both more efficient and precise.

However, some researchers including Sarica and Luo, 2021, have raised some concerns concerning stopword removal. Sarica and Luo, 2021, argues that stopword removal might remove valuable contextual information, which could be crucial for certain tasks. The trade-off lies in balancing the need to simplify textual data while keeping the context and meaning of the text.

There is primarily two ways of identifying stopwords:

1. Predefined List: This approach matches the tokenized text with a predefined list of words which is defined as stopwords. This method is straightforward, but it ignores the context of the text (J. Kaur and Buttar, 2018).

2. Frequency-based Removal: This method removes words that occur more frequently than a specified threshold. This method risks eliminating words with a semantic meaning that occur frequently (J. Kaur and Buttar, 2018).

Schofield et al., 2017 proposes combining the two methods by having a predefined stopword list but conditioning removal on a specified frequency threshold. Table 4 shows an example of a stopword with typical stopwords being removed from the text.

Text	Stopword Removal
[“The”, “builders”, “are”, “using”, “bricks”]	[“Builders”, “using”, “bricks”]

Table 4: Example of Stopword Removal in NLP

2.3.3 Normalization Techniques

The two most common normalization techniques in NLP are stemming and lemmatization. Both techniques aim to reduce words into a more standardized form. Stemming simplifies words by transforming them into their root form (Santosh and Hegadi, 2018, p.594). Lemmatization, on the other hand, reduces the words into their base or dictionary form, ensuring that the word is a valid word in the given language (Santosh and Hegadi, 2018, p.599). The main objective of these normalization techniques is to create a more compact and efficient representation of text while preserving its semantic meaning (C. Manning and Schutze, 1999, p.194). Having a more compact representation of text can help reduce complexity in subsequent ML applications.

Critics such as, (C. Manning and Schutze, 1999, p.194), have pointed out that there is a potential downside to these techniques, which is a loss of semantic meaning. Normalization techniques can sometimes cause two distinct words with different meanings to be reduced into the same lemma or stem. This can lead to a loss in information, and affect the performance of further NLP tasks. In a study assessing the effectiveness of normalization techniques, Hickman et al., 2022, found varied results. The study assessed the performance of ML algorithms after and before using normalization techniques. The results show that stemming improved the performance of 6 out of 12 studies looked at. Lemmatization, on the other hand, improved performance in 6 out of 8 studies looked at. It is important to note that the study was conducted using English text, and the tools for lemmatization and stemming in English may be more advanced and developed compared to other languages, such as Norwegian. Normalization techniques used on English text seem to have varied results on its effectiveness.

An example of stemming applies to a sentence can be seen in Table 5.

Text	Stemming
[“Builders”, “using”, “bricks”]	[“Builder”, “use”, “brick”]

Table 5: Example of Stemming in NLP

2.3.4 Vectorization

Vectorization is an important process, translating textual information into a numerical format, enabling computers to process the data. As written by Krzeszewska et al., 2022, the process involved mapping textual content to a multi-dimensional vector space, with each dimension corresponding to a distinct textual feature. This step is necessary since ML models require numerical input to function, as stated by

Russell and Norvig, 2010. Vectorization is therefore an essential tool, ensuring that text data can interact with computers and ML algorithms.

In this thesis, multiple vectorization techniques have been utilized. This subsection will present each of these methods. For clarification, a “corpus” is the collection of many “texts” or “documents”, while a “document” is an individual piece of writing within the corpus.

Term Frequency - Inverse Document Frequency

Term Frequency- Inverse Document Frequency (TF-IDF) is a vectorization method used to assess the importance of a word within a dataset (C. D. Manning, 2009). The method assigns a weight to each term in a document, determined by its frequency within that document. This weight is known as the term frequency (TF), denoted by $TF_{t,d}$, and it is mathematically shown in Equation 1 (C. D. Manning, 2009).

$$\text{TF}(t,d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d} \tag{1}$$

Term frequency represents the proportion of occurrences of a specific term in a document, with the value ranging between 0 and 1. The Inverse Document Frequency (IDF) measures the rarity of a term across the entire corpus. The total number of documents in the collection is denoted by N, and the IDF for a term t is given in Equation 2 (C. D. Manning, 2009, p.118) .

$$\text{IDF}(t) = \log\left(\frac{N}{\text{Number of documents containing term } t}\right) \tag{2}$$

By integrating both metrics, a weight for each term in the document can be calculated. The TF-IDF value, which is shown in Equation 3.

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t) \tag{3}$$

The TF component explains the significance of a term within a specific document, while the IDF is the term’s uniqueness across the entire dataset. Combined, the TF-IDF score measures a term’s relative importance within a particular document, taking into account both the local frequency (TF) and its global rarity (IDF). A high TF-IDF score indicates that the term is both important and distinctive to the document it appears in, and a low score suggests that the term is common across many documents, and is less significant in the specific document.

TensorFlow and Keras Tokenization

The TensorFlow and Keras package uses a vectorization algorithm based on the TF-IDF model (Martín Abadi et al., 2015). The TF-IDF algorithm is described in detail in the previous subsection. TensorFlow and Keras have been used for vectorization of the text data in every LSTM model in this thesis.

Word2Vec

Another method employed in this thesis is Word2Vec, a technique that creates word embeddings. Developed by Mikolov et al., 2013, the algorithm uses neural networks to generate vector representations of words based on the textual context of the words. These embeddings capture semantic meaning between words, which means that words with a similar meaning will have similar vector representations.

Sentence Piece

SentencePiece is a vectorization algorithm used with the model Text-To-Text-Transfer-Transformer (T5). SentencePiece is an unsupervised text tokenizer developed by Google (SentencePiece Contributors, 2023). SentencePiece is mainly used for Neural Network-based text generation systems, where the vocabulary size is predetermined before the training of the neural model. Having a predetermined vocabulary size makes it effective at managing out of vocabulary words, making the T5 model good in different NLP tasks.

2.4 Machine Learning

This chapter explores the theory of various ML techniques. ML, a subfield of AI, focuses on developing systems that can learn or enhance performance from experience, often represented as data (Mitchell, 1997). Unlike traditional programming, in ML the system is not explicitly programmed to perform a certain task, the system is expected to learn from patterns from data, thereby making the system able to perform tasks or make predictions without directly being programmed to do so (Alpaydin, 2020, p.2).

ML is commonly divided into three main types: supervised learning, unsupervised learning, and reinforcement learning (Bishop, 2006, p.3). This thesis will mainly focus on supervised learning since it is the methodology used in this research.

In supervised learning, the objective is to make a model that can accurately predict or classify new instances based on a given set of labeled training data. The training data consists of samples like: $\boldsymbol{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)}$, where each $\boldsymbol{y_j}$ is the output of a unknown function $\boldsymbol{y = f(x)}$. The task is to find a function $\boldsymbol{h}$ that is the best approximation to the unknown function $\boldsymbol{f}$. The performance of the hypothesis function $\boldsymbol{h}$ is assessed using a test dataset, which is separate from the training dataset used to build the model (Russell and Norvig, 2010, p.695). The test dataset provides an unbiased evaluation of the model's performance, giving feedback on how well the model learns patterns in the data, and how good it is to generalize to unseen data.

In supervised learning, the output variable $\boldsymbol{y}$ typically falls into one of two main categories: Classification or Regression. In regression, the output variable is a continuous numerical value. This method is used when the aim of the model is to predict a quantifiable outcome. A classic example is predicting the temperature the next day, where the temperature is the target variable and a number. In classification, the output variable is viewed as a category. Using an example with weather, in a classification task the weather condition for tomorrow can be predicted. The options for weather status can be sunny, cloudy, or rainy, which functions as classes. Both examples are found in Russell and Norvig, 2010, p.696. This study will exclusively focus on classification methods, and the next subsections will explore different classification techniques and methods.

2.4.1 Training, Test, and Validation Datasets

In ML, training and evaluating a model requires a sequence of steps to ensure its effectiveness. The first step is the training phase, where the model is exposed to labeled data, consisting of input-output pairs. The model learns patterns in the data, and by altering its internal parameters it aligns its predictions with the actual output. Sometimes a validation set is used to fine-tune the model's parameters, to avoid overfitting to the training data (Müller and Guido, 2016, p.262).

After the model is trained and validated, the testing phase starts, where the algorithm is evaluated using a distinct set of data that is unseen, known as the test set. This phase evaluates the performance of the ML algorithm, looking at its ability to make accurate predictions of new data (Müller and Guido, 2016, p.17). A challenge called overfitting, is when the model learns patterns too specific to the training set and therefore performs better on the training set compared to the test set.

Splitting the data into a training, validation, and test set can be challenging if the dataset is small. It is important to have enough data for each phase to train and measure the performance of the model accurately (Xu and Goodacre, 2018). When there is limited data, cross-validation becomes an important and effective technique (Müller and Guido, 2016, 257).

2.4.2 Cross-Validation

Cross-validation is a technique used to assess how effectively a model can generalize to new, unseen data. This technique divides the dataset into multiple folds and iteratively uses different folds as training and test data (Müller and Guido, 2016, p.252).

The advantage of using cross-validation in predictive ML tasks is the ability to average out biases (Müller and Guido, 2016, p.254). This is achieved since there are multiple training and test sets, which makes the model robust. Cross-validation can mitigate the risk of overfitting, making sure that the model generalizes (Santos et al., 2018). Additionally, cross-validation provides a method for setting hyperparameters, optimizing the model's performance.

K-Fold

K-Fold Cross-Validation is a widely used method for cross-validation. This approach divides the dataset into $\boldsymbol{k}$ number of equal-sized folds. Each fold is used as the test set exactly once, while the remaining folds function as the training set (Müller and Guido, 2016, p.252). The model is trained using the training folds before it is tested using the test fold. This process is repeated $\boldsymbol{k}$ times. The results of every iteration are aggregated and averaged to provide an assessment of the model. Figure 3 shows how the dataset is split and utilized in K-Fold Cross-Validation with $\boldsymbol{k}$ equal to five.

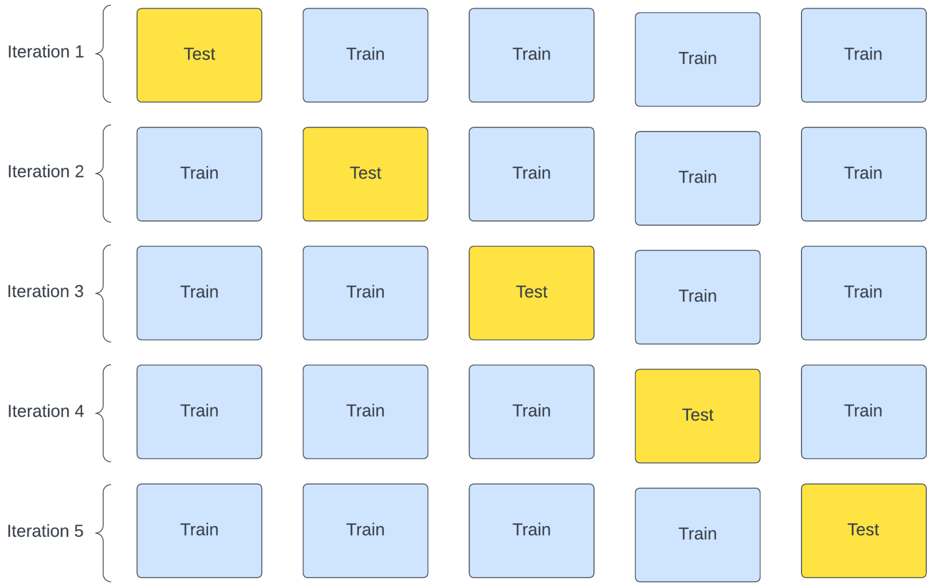


Figure 3: Illustration of K-fold Cross-Validation Process with Five Iterations - Author's own illustration - Adapted (Müller and Guido, 2016, p.252)

2.4.3 Performance Measures - Classification

In this subsection, performance measures of ML algorithms in classification tasks are explored. Given that the data in this thesis is categorical, the focus will be on classification metrics.

The aim in classification tasks is to assign an input vector $\mathbf{X}$, to one distinct class $\mathbf{k}$, among all classes $\boldsymbol{C_k}$, where $\mathbf{1, ..., \boldsymbol{K}}$ (Bishop, 2006, p.179). In standard classification these classes are mutually exclusive, meaning each input vector is categorized into exactly one class.

The primary focus of this thesis is binary classification problems, which is when instances can labeled as one out of two states (Russell and Norvig, 2010, p.696). Often these states are labeled as positive and negative, mathematically shown as $\boldsymbol{C_k} \in \mathbf{0, 1}$ (Bishop, 2006, p.180). When classification has more than two classes it is called multi-class classification. This occurs when $\boldsymbol{K} \geq \mathbf{2}$ and there is a finite number of classes (Shalev-Shwartz and Ben-David, 2014, p.47). Moving on, the performance measures discussed will be binary classification problems, since the research problems are simplified into binary classification problems.

Confusion Matrix

In binary classification, the evaluation of a model's predictions can be done using a confusion matrix, classifying all predictions into four categories. A True Positive (TP) occurs when the model correctly predicts a positive sample (Luque et al., 2019). A False Positive (FP), or a Type 1 error, is a positive prediction, when the instance actually is negative (Vujović et al., 2021). A True Negative (TN) is when the model accurately predicts a negative outcome. In contrast, a False Negative (FN), known as a Type 2 error, is a negatively labeled instance that is wrongly labeled (Vujović et al., 2021).

These four values together compose a confusion matrix. An example of a confusion matrix is illustrated in Figure 4.

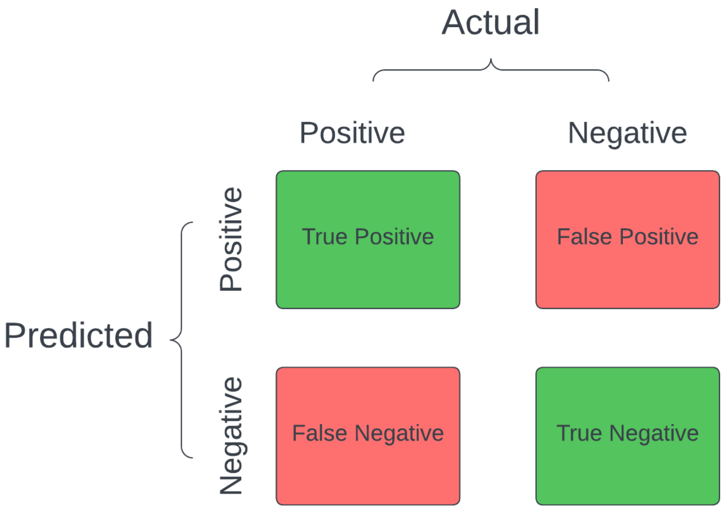


Figure 4: Example of a Confusion Matrix for Binary Classification - Author's own illustration - Adapted Müller and Guido, 2016, p.281

Accuracy

Accuracy is a metric that measures the ratio of correct predictions to the total number of predictions made (Baldi et al., 2000). Mathematically it is expressed in Equation 4.

$$\begin{aligned}\text{Accuracy} &= \frac{\#\text{ of correct predictions}}{\text{Total }\#\text{ of predictions}}\\ &= \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}\end{aligned} \tag{4}$$

Accuracy provides an intuitive metric of understanding how often a ML model is correct. However, accuracy can be misleading in cases where the dataset is imbalanced (Juba and Le, 2019). In a scenario involving a group of 1000 people, where one member of the group has cancer, a model that predicts that everyone is cancer-free will have 99.9 % accuracy. While the accuracy appears impressive, it does not give any insight into the models ability to correctly identify cancer (Bishop, 2006, p.45). Another weakness of using accuracy is that it does not differentiate between Type 1 and Type 2 errors, only providing information about the total amount of errors. In many scenarios, it is valuable to differentiate between types of errors (Jain and H. Kaur, 2017). Using cancer diagnoses as an example again, the cost of a FN might be significantly higher than a FP.

Precision

Precision is the proportion of the positive identifications that are correct. The definition of precision is given in Equation 5 (Luque et al., 2019).

$$\begin{aligned}\text{Precision} &= \frac{\text{\# of true positive predictions}}{\text{\# of true positive predictions} + \text{\# of false positive predictions}} \\ &= \frac{\text{TP}}{\text{TP} + \text{FP}}\end{aligned} \tag{5}$$

Precision is a metric that is particularly important when dealing with imbalanced datasets (Juba and Le, 2019). The reason for that is that correctly identifying positive samples, is especially important when there are few positive samples in the dataset. The precision only looks at the positive classified examples and assesses how many of them are correct.

However, the precision has some limitations. A significant limitation is that the precision completely ignores the TNs (Flach and Kull, 2015). This can be problematic in scenarios where identifying TNs is as important as identifying TPs. Another problem that can occur if only precision is used as the performance metric is a conservative model that labels very few samples as positive. While this might lead to a model with high precision, it might still miss many TPs. This problem shows the need to use several performance metrics to get a complete understanding of how the model is performing.

Recall

Recall, also known as sensitivity, measures the proportion of actual positives that a model correctly identifies. Mathematically the recall is shown in Equation 6 (Chicco, 2017).

$$\begin{aligned}\text{Recall} &= \frac{\text{\# of true positive predictions}}{\text{\# of true positive predictions} + \text{\# of false negative predictions}} \\ &= \frac{\text{TP}}{\text{TP} + \text{FN}}\end{aligned} \tag{6}$$

Recall looks at the completeness of the positive predictions made by a model. The metric tries to minimize the number of FNs. One method of increasing the recall is to classify more instances as positive, but this will lead to a decrease in precision. If all samples are labeled as positive, the recall would be 1, but the precision would be very low, indicating many FPs (C. Manning and Schutze, 1999, p.156).

The importance of the recall compared to the precision is often dependent on the use-case. Using the medical diagnostics example with cancer screening, a FN might be more undesirable than a FP. Then recall might be considered more important than precision. This highlights the trade-off between precision and recall. Maximizing one of them often leads to the reduction of the other, therefore a balanced approach is often necessary (C. Manning and Schutze, 1999, p.22).

F-1 Score

The F1-score is the harmonic mean between precision and recall, and is made to create a performance metric that balances both precision and recall (C. Manning and Schutze, 1999, p.156). The formula for the F1-score is given in Equation 7.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \tag{7}$$

Using the F1-score is especially relevant in scenarios when using imbalanced datasets, since the accuracy might not be a precise measure to evaluate a model's performance (Juba and Le, 2019). By combining precision and recall, the F-1 score offers a more comprehensive evaluation of a model's performance than both metrics isolated. It balances the trade-off between precision and recall, creating a reliable measure in situations when it is desirable to minimize both FPs and FNs.

One of the limitations of using the F1-score is that it treats both precision and recall equally. This means that the result is an aggregate between them, and it is impossible to see if the limitations of the model are because of the recall or precision by only looking at this metric. It means that the metric does not distinguish between Type 1 and Type 2 errors. Therefore, it is often beneficial to look at the F1-score in combination with other performance metrics to fully understand the model's performance and behavior.

The Precision-Recall Curve

When evaluating a model's performance, especially using an imbalanced dataset, relying on a single performance metric might not reflect the model's performance (Hasanin et al., 2019, Branco et al., 2016). Therefore the Precision-Recall (PR) curve emerges as a viable tool. It offers a visual representation showing the trade-off between precision and recall for different thresholds.

The PR curve is a graph with recall on the x-axis and precision on the y-axis. It shows the model's behavior by illustrating the precision and recall at different threshold levels. In classification, a threshold is the level of certainty a model has to reach to label an instance as positive (Saito and Rehmsmeier, 2015). Traditional performance metrics, such as the four already discussed: Accuracy, Precision, Recall, and F1, are single-threshold measures, where the scores are calculated based on a specific threshold. The PR curve, on the other hand, evaluates the model's performance across all thresholds.

The curve is valuable when fine-tuning the model's behavior to balance precision and recall in a manner that is preferable for the task's use-case. Some tasks demand minimizing FPs, while other tasks wish to minimize FNs. The PR curve offers a tool to understand how thresholds can affect the results, enabling informed decision-making in setting the thresholds.

An additional metric that can be derived from the curve is the area under the PR curve (AUC-PR), which is the integral under the PR curve. It gives a single number describing the model's performance over all thresholds (Sofaer et al., 2019). A perfect classifier will have AUC-PR equal to 1. For random guessing the AUC-PR will be the same as the proportion of positive samples in the dataset. Therefore, the AUC-PR offers a single metric that can describe the performance of the model over all thresholds.

ROC curve

The Receiver Operating Characteristic (ROC) curve is a graph that depicts the True Positive Rate (TPR) against the False Positive Rate (FPR) at different thresholds (C. Manning and Schutze, 1999, p.162). The mathematical Equation of the TPR and the FPR is shown in Equation 8 and 9 (Davis and Goadrich, 2006).

$$\mathrm{TPR} = \mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{8}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}} \tag{9}$$

Similarly to the PR curve, the ROC curve assesses the model's performance over different thresholds. The curve works in exactly the same way as the PR curve but shows the TPR and the FPR, instead of the Precision and Recall (Hoo et al., 2017). The curve is a tool that can be used to select optimal thresholds to balance the number of FPs and the number of .

An ROC curve will typically go from the bottom left corner to the top right corner. A high performing classifier rises steeply towards the upper left corner of the plot, indicating a high TPR and a low FPR (C. Manning and Schutze, 1999, p.162). An example ROC curve is illustrated in Figure 5, with different colors representing classifiers of different performance levels.

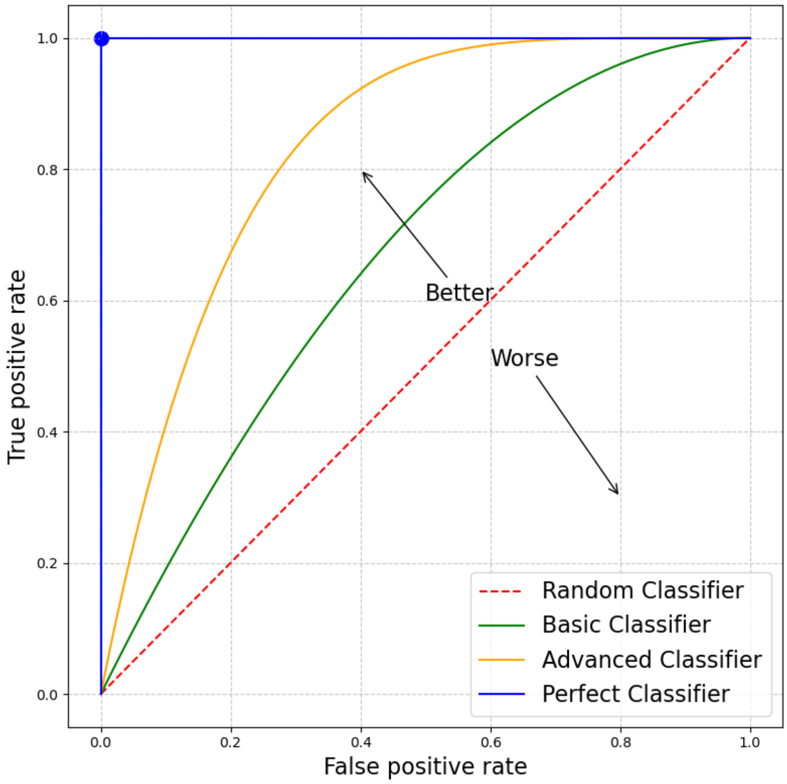


Figure 5: Example of a ROC-Curve - Author's own illustration - adapted Thoma, 2018

The AUC is also a metric used when looking at ROC curves. The ROC-AUC will range from 0.5, indicating a random classifier, to 1.0, indicating a perfect classifier (Hoo et al., 2017). Previous studies have shown that classifiers that perform well according to ROC curve, often also perform well when looking at the PR curve (Davis and Goadrich, 2006), since both curves are related.

2.4.4 Multi-Label Classification

Multi-label classification refers to the problem where an instance can be associated with multiple classes (Tsoumakas and Katakis, 2007). A movie can be categorised into multiple genres. An example of this can be seen in Table 6, where "Inception" can be categorized as "Romance", "Action", and "Fantasy".

Movie Title	**Romance**	**Action**	**Fantasy**
The shape of Water	✓		✓
Titanic	✓	✓	
Inception	✓	✓	✓

Table 6: Example Multi-Label Classification

To solve a multi-label classification problem a strategy is needed to simplify the problem. A common approach is Binary Relevance, where each label is treated as an independent binary classification problem (M.-L. Zhang et al., 2018). This is a straightforward solution, but it ignores potential correlations between labels. Classifier Chains is a similar method to Binary Relevance, with the only difference being, that it uses previous classifications as additional input to the classifier (M.-L. Zhang et al., 2018).

Label Powerset is another method used in multi-label classification where each unique combination of labels is treated as a distinct class (Read et al., 2014). This method is effective at capturing correlations between labels. The issue of using this method is that many classes will make the total number of combinations very high.

Choosing the best approach for multi-label classification depends on the dataset, and label correlation. Each method has strengths and weaknesses, and the choice of method should align with the requirements of the specific problem.

Micro and Macro-Averaging Techniques

In multi-label classification, the performance metrics such as precision, recall, and F1-score differ compared to single-label classification. In Multi-label classification an instance can be in two classes, and therefore there is a need for a method to calculate the performance metrics differently. Therefore micro and macro averaging techniques are used (Pereira et al., 2018).

Micro-averaging is the method of aggregating all predictions and computing the average metric. Looking at precision, micro-averaging calculates the TP and FP across all classes, and computes the precision. The formula for micro precision is given in Equation 10 (Sokolova and Lapalme, 2009).

$$\text{Micro Precision} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i} \tag{10}$$

Macro-averaging computes the performance metric independently for each class and then calculates the average (Pereira et al., 2018). This means that every class is given the same importance, regardless of the class size. Using Precision again, the macro-average can be calculated by calculating the precision for each class and then averaging the results. The formula for macro precision can be seen in Equation 11 (El Kafrawy et al., 2015).

$$\text{Macro Precision} = \frac{1}{n} \sum_{i=1}^{n} \text{Precision}_i \tag{11}$$

Both methods can be useful to look at. Micro-averaging takes into account the performance of frequent labels, while macro-averaging treats every class equally. The same procedure would be used to calculate the micro and macro recall and F1-score, as Equation 10 and 11 calculates precision.

2.4.5 Class Imbalance

Dealing with class imbalance is a common challenge in ML, particularly classification tasks. The problem arises when one class significantly outnumbers another class in a dataset, leading to biased ML models that tend to predict the majority class (Müller and Guido, 2016, p.277). This chapter explores various methods used to address the issue of class imbalance.

Under and Over-sampling

Under-sampling and over-sampling are two common techniques used to mitigate the problems of class imbalance (Han et al., 2005). Both methods aim to equalize the class imbalance, providing a balanced dataset for training a ML model.

Under-sampling involves removing a number of instances in the majority class to make the frequency of every class the same (Mohammed et al., 2020). The most common approach to under-sampling is using random under-sampling, where random instances of the majority class are removed. This is an easy to implement method. The main drawback is data loss, which is particularly problematic in cases when there are very few samples of the minority class since much valuable information will be discarded.

Over-sampling, on the other hand, increases the number of instances of the minority class (Mohammed et al., 2020). The simplest method is called random over-sampling and randomly chooses instances of the minority class to duplicate. The problem of this is the possibility of overfitting since the same data point can be duplicated many times. Examples of both undersampling and oversampling are shown in Figure 6.

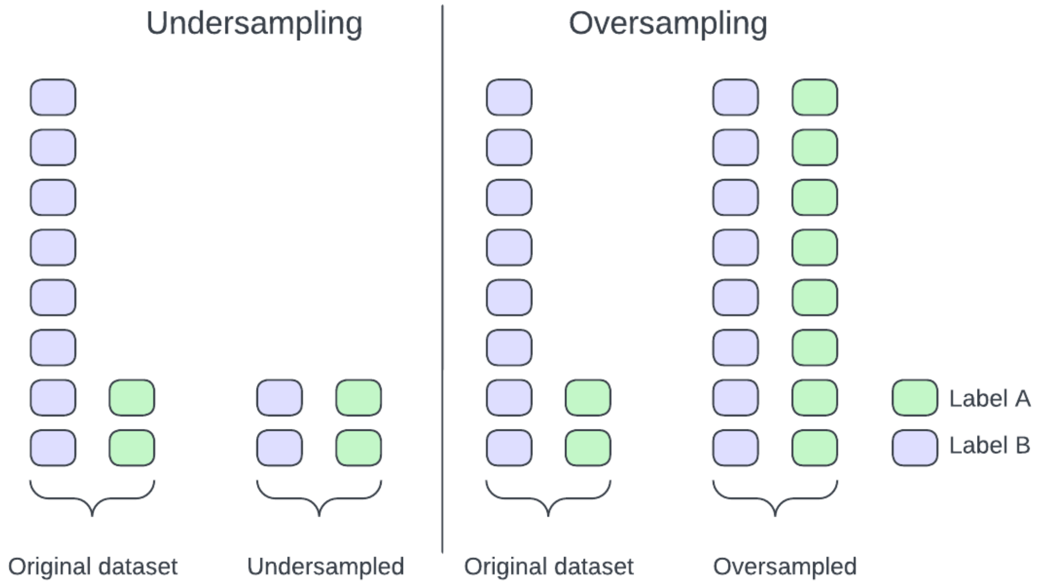


Figure 6: Illustration of Undersampling and Oversampling techniques - Author's own illustration - Adapted Xia et al., 2019

The Synthetic Minority Over-sampling Technique (SMOTE) offers a different over-sampling approach. SMOTE generates new, synthetic instances of the minority class (Han et al., 2005). To achieve this the method chooses samples from the minority class, and slightly alters them, creating a new sample. SMOTE offers a more diverse set of data points for the minority class, removing some of the risk of overfitting. A problem with SMOTE is that the random new samples might introduce "noise" to the model, as some of the artificially created samples might overlap with the majority class (Han et al., 2005). There are many different SMOTE algorithms designed for different purposes, such as Borderline-SMOTE and Adaptive Synthetic Sampling (Han et al., 2005, He et al., 2008).

Weights

Adjusting weights within a ML algorithm is another method to deal with class imbalance, focusing on making the model more sensitive to the minority class (Zhu et al., 2018). This approach tries to penalize the misclassification of the minority class more heavily than errors predicting the majority class. This method is simple to implement for some ML algorithms and has the advantage of not having to alter the dataset. This means that all the information in the data is preserved.

A limitation of this technique is that internal weights are not supported for every ML model. This will restrict the use of weights depending on which models and frameworks are used. In classification tasks in ML, class imbalance is a complex issue. There is no universal solution that can be applied to every model and dataset. The goal is to end up with a model with low bias, that can accurately predict both the minority and majority classes.

2.5 Deep Learning

DL focuses on algorithms inspired by the structure and function of the human brain, particularly artificial neural networks. DL has been especially important in driving advancements in fields such as computer vision and NLP. DL is a subfield of ML, which again is a subfield of AI, their relationship is depicted in Figure 7.

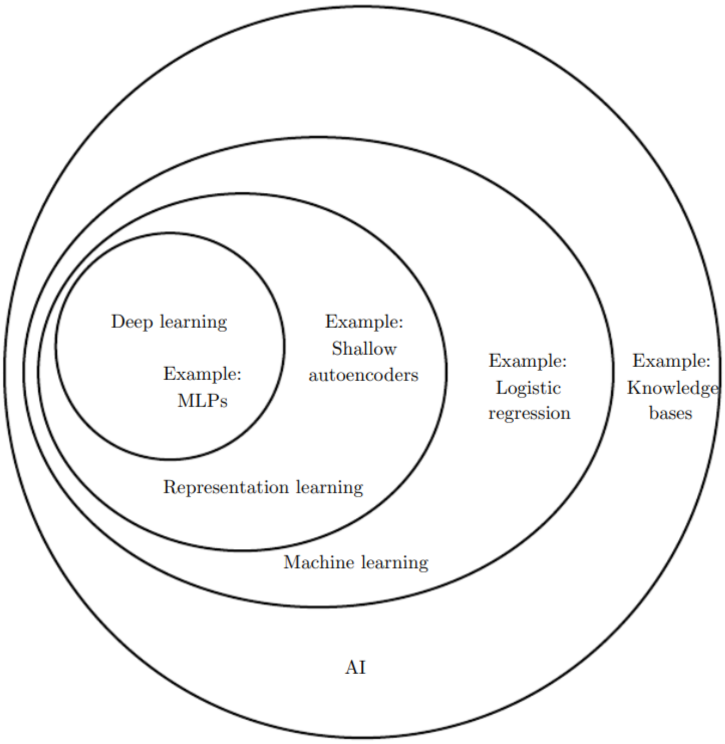


Figure 7: Venn Diagram of AI, ML, and DL (Goodfellow et al., 2016, p.24)

In this thesis, the data source is primarily text data, and therefore NLP techniques within DL will be explored. Techniques that will be explored are Recurrent Neural Networks (RNNs), LSTMs, and transformers since these techniques are especially effective in handling sequential and linguistic data.

2.5.1 Artificial Neural Networks

The most basic unit within an artificial neural network is the artificial neuron, often referred to as a perceptron. Each neuron receives multiple input signals, denoted as x, and processes these input signals to produce a single output signal (Aggarwal, 2018, p.5). Every input the neuron receives is associated with a weight, w, which determines the importance and influence of that particular input. In addition, each neuron has a bias, b, adjusting the output independently of the input (Goodfellow et al., 2016, p.15).

The process of calculating the output of a neuron, using both the input weights and bias, can be represented mathematically as shown in Equation 12. During the training of a neural network, these weights and biases are iteratively adjusted (Nielsen, 2015, p.16). The adjustments aim to improve the network's performance and enable the network to learn from complex patterns in data.

$$z = \sum_{i=1}^{n} w_i x_i + b \tag{12}$$

In a neural network, each neuron uses input from other nodes with its own weights and biases, and passes the combined input, z, into an activation function. The activation function decides the output of a specific neuron (Goodfellow et al., 2016, p.171). The activation function takes the input, z, and produces an output using a predefined function.

Historically, the Sigmoid function has been a popular activation function. However, in newer neural networks, the Rectified Linear Unit (ReLU) function has gained significant popularity in DL methods (Glorot et al., 2011). Both the sigmoid function and the ReLU function are mathematically expressed in Equation 13 and 14.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{13}$$

$$\text{ReLU}(z) = \max(0, z) \tag{14}$$

The entire process in neural networks, of how a neuron receives multiple inputs and generates an output, is illustrated in figure 8.

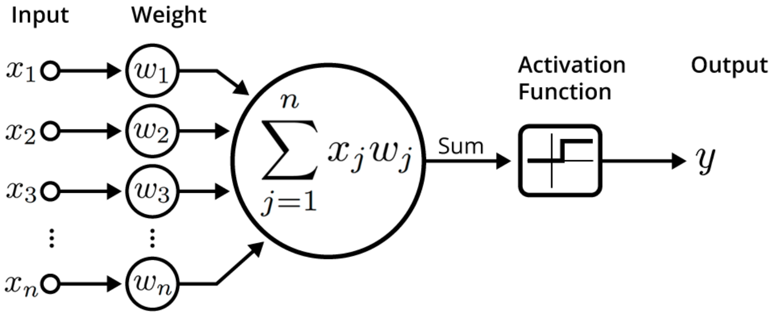


Figure 8: Neural Network Node Activation Process (McCullum, 2020, p.24)

In multi-layer neural networks, the neurons are arranged in a layered fashion, where the input and output layers are separated by a number of hidden layers (Aggarwal, 2018, p.6). Feed-forward neural networks are a type of artificial neural network where the connections between neurons do not form any cycles (Goodfellow et al., 2016, 168). The design ensures a one-way "flow" of data, starting in the input layer, moving through all hidden layers, before ending in the output layer (Nielsen, 2015, p.12). Figure 9 provides a visual representation of how feed-forward neural networks look, each circle representing a neuron.

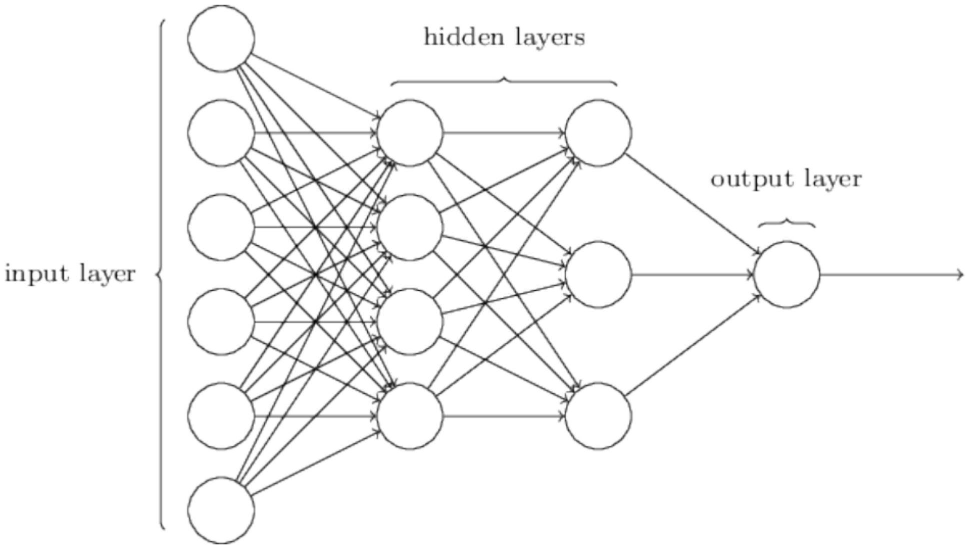


Figure 9: Feed-forward Neural Network Architecture (Nielsen, 2015, p.11)

When training artificial neural networks the backpropagation algorithm is utilized. The backpropagation algorithm works by adjusting weights and biases based on the errors in the predictions (Goodfellow et al., 2016, p.204). This adjustment allows the neural network to learn and enhance its performance.

The backpropagation process begins with calculating the network's prediction error, typically using the squared difference between the predicted values and the target values. This error calculation serves as the basis for adjusting weights and biases moving on (Alzubaidi et al., 2021). After the error is calculated, the gradient descent algorithm is used. Gradient descent is an optimization algorithm that finds the local minimum of the error function. The algorithm works by updating the network's weights in response to the calculated error, with the goal of minimizing the error over time (Alzubaidi et al., 2021). The gradient of the error function is calculated using the network's weight and is shown in Equation 15.

$$\nabla_w E = \frac{\partial E}{\partial w} \tag{15}$$

In Equation 15, $\nabla_w E$ is the gradient of the error function E with respect to the weights w of the network. Furthermore, the weights are updated using the formula expressed in Equation 16 (Goodfellow et al., 2016, p.231).

$$w_{\text{new}} = w_{\text{old}} - \eta \cdot \nabla_w E \tag{16}$$

In this formula w_{new} represent the updated weights, while w_{old} is the previous weights, η is the learning rate, $\nabla_w E$ is the gradient of the error function E. The learning rate, η, is a small positive number that controls the step size of the weight update, ensuring stable adjustments.

A key element in backpropagation is the chain rule, which is used for computing the gradient of the loss function with respect to each weight for networks with multiple layers. Each layer in the network is viewed as a unique function, and the chain rule is applied to multiply the derivatives of all functions (Goodfellow et al., 2016, p.205). This allows the error gradient to be propagated backward through the network, from the output layer to the input layer, ensuring all the weights in all layers are updated in response to the error.

To summarize, backpropagation distributes blame back through the network, adjusting weights to enhance learning. While feed-forward neural networks can be used for certain tasks, they struggle with sequential data, such as time series or text, due to their inability to remember previous inputs. To address this limitation, RNNs were designed.

2.5.2 Recurrent Neural Networks

RNN are a type of neural network, specialized for processing sequential data and natural language. Unlike traditional feed-forward neural networks, RNNs use a feedback loop, enabling it to keep memory of previous input (Fei and Lu, 2017). This feature makes RNNs able to capture sequential dependencies.

Structurally, RNNs integrates a form of "memory" by incorporating its input from previous time steps. In addition to the current input value, denoted as x_t, RNNs also consider a hidden state from the previous time step, labeled as h_{t-1}. A neuron's hidden state is calculated using Equation 17, where w_x, and w_h represents the weights of the current input and the previous hidden state, respectively, b is the bias, and f is the activation function (Goodfellow et al., 2016, p.379).

$$h_t = f(w_x \cdot x_t + w_h \cdot h_{t-1} + b) \tag{17}$$

This mechanism enables neurons to keep some state information across time steps. As a consequence, the output at any given time step is influenced by all preceding inputs. The presence of the hidden state enables RNNs to detect patterns in sequences, a capability that standard feed-forward neural networks lack. This unique characteristic of RNNs is depicted in Figure 10, which shows an RNN unrolled across time steps. In the Figure, the same network A is applied at each time step, with the hidden state h_t being passed along to the next step. This visualization shows how RNNs can handle different lengths of sequences as input, and maintain information across these sequences.

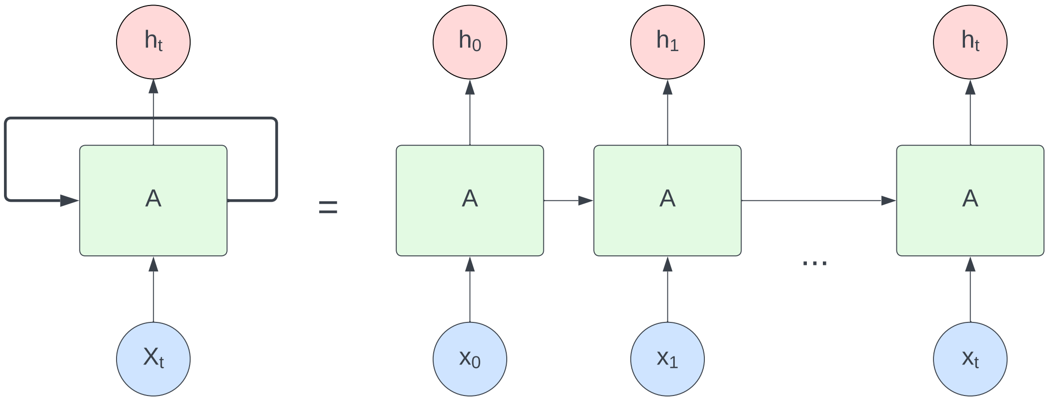


Figure 10: An unrolled RNN - Author's own illustration - adapted from (Olah, 2015)

However, RNNs encounter two challenges called the vanishing and exploding gradient problems. As RNNs iterate through sequences, the gradients, using backpropagation to adjust the networks weights, can either go to zero or increase exponentially (Sutskever, 2013). The vanishing gradient phenomenon happens due to the repeated multiplication of numbers smaller than one, during backpropagation through time (Alzubaidi et al., 2021). This makes the network unable to utilize information from earlier steps in the sequence. Exploding gradients, on the other hand, happen because of the multiplication of numbers larger than one during backpropagation (Alzubaidi et al., 2021). This leads to unstable network behavior, where the weight updates are so large that they destabilize the learning process. To address these issues, several advanced architectures have been proposed. One of the most popular of these architectures is called LSTM networks.

2.5.3 Long Short-Term Memory

The LSTM unit, introduced by Hochreiter and Schmidhuber, 1997, represents an advancement in the field of RNNs, specially designed to address the challenges of gradient instability. LSTMs are specially designed to avoid the long-term dependency problem, making them excellent at capturing information from sequences that have a long interval of relevance (Sutskever et al., 2014).

Figure 11 shows a single LSTM unit at time step t. It displays the flow of information through various gates, and how the cell state is updated. A LSTM cell contains the following three gates: a forget gate, an input gate, and an output gate (Goodfellow et al., 2016, 412). These gates manage both the hidden state and the cell state over time, regulating the flow of information, and deciding which data should be updated, stored, or discarded as the network process sequences.

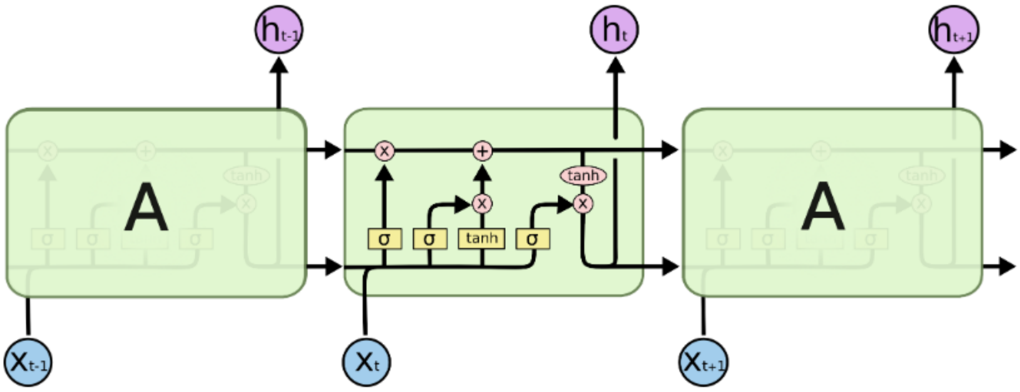


Figure 11: LSTM Module Structure (Olah, 2015)

The forget gate within a LSTM unit has the responsibility to determine which information should be removed from the cell state. It employs a sigmoid function (σ) to produce a number ranging from 0, indicating forget everything, to 1, indicating complete retention. The operation of the forget gate is mathematically expressed in Equation 18 (Graves, 2013).

$$\text{Forget gate:} \quad f_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f) \tag{18}$$

In Equation 18, x_t is the current input vector, h_{t-1} is the previous hidden state, W_f and b_f are the weight matrix and the bias term of the forget gate, respectively. The output f_t indicates which parts of the previous cell state, C_{t-1}, should be kept or discarded.

The input gate's function is to decide which values are to be updated in the cell state. It generates a vector of new candidate values, $\tilde{c}_t$, which could be added to the cell state. This process is represented in Equation 19 and 20 (Graves, 2013).

$$\text{Input gate:} \quad i_t = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i) \tag{19}$$

$$\text{Cell input:} \quad \tilde{c}_t = \tanh(W_c \cdot x_t + U_c \cdot h_{t-1} + b_c) \tag{20}$$

where, σ and tanh are the sigmoid and hyperbolic tangent functions, respectively. The cell state is then updated by using both the forget gate output with the new candidate values, expressed in Equation 21 (Graves, 2013).

$$\text{Update the cell state:} \quad c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{21}$$

Lastly, the output gate controls the flow of information from the cell state to the hidden state, which is either used for predictions or transferred to the next LSTM cell. The mathematical expressions of the output gate and the computation of the hidden state are shown in Equations 22 and 23 (Graves, 2013).

$$\text{Output gate:} \quad o_t = \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o) \tag{22}$$

$$\text{Compute the LSTM cell output:} \quad h_t = o_t \odot \tanh(c_t) \tag{23}$$

In these formulas, $\odot$ means the element-wise multiplication making sure each part of the cell state is updated based on the calculated outputs of the forget and input gates. The output gate utilizes the updated cell state to generate a new hidden state, h_t, hopefully only containing relevant information.

Through this structure, LSTM networks have revolutionized how sequential data is handled in neural networks. By addressing the limitations of traditional RNNs, LSTMs have enhanced the performance in solving complex tasks involving sequential data, such as language processing and time-series tasks.

2.5.4 Transformers and the Attention Mechanism

In 2017 the Google engineers, Vaswani et al., 2017, wrote the paper "Attention Is All You Need", which introduced the transformer architecture, and a paradigm shift in NLP. Vaswani et al., 2017's model beat traditional models, and set a new benchmark for text translation tasks, and has revolutionized other NLP tasks. This chapter starts with explaining the attention mechanism before describing the transformer architecture. Lastly, T5 model is explained, which is the transformer model utilized in this thesis.

In 2014, Bahdanau et al., 2014, introduced the attention mechanism, changing how neural networks process text. This mechanism enables models to dynamically focus on a particular segment of a sequence during text processing, similar to how humans read and comprehend text. This is achieved through learning alignments between the states of the decoder and the encoded representations of the input, allowing the model to assign different importance to different words in an input sequence.

Expanding this mechanism, the self-attention mechanism was developed, which is an integral part of the transformer model developed by Vaswani et al., 2017. Self-attention uses a single layer by aligning its processing to all positions within the sequence itself. This allows the model to assess the entire sequence's information when processing each word, capturing interdependencies across the entire sequence.

The potential of self-attention was realized by Vaswani et al., 2017 in the transformer architecture. This architecture uses scaled dot-product to calculate the attention using the formula in Equation 24.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{24}$$

In Equation 24 Q, K, and V represent queries, keys, and values, respectively. The Transformer calculates the dot products of the query with all keys and scales the results by the inverse square root of the dimensionality. Lastly, a softmax function is applied, which transforms the scaled dot product into a distribution. In this distribution, each weight indicates the attention, or importance, that each word in the input sequence is given. This allows the Transformer to focus on the most relevant part of information when constructing the output sequence.

The Transformer model enhances its attention capabilities using Multi-Head Attention, which allows multiple self-attention operations to be calculated in parallel. This allows the model to capture multiple parts of the input at the same time. As shown in Figure 12d, queries q, keys k, and values v are each projected across h layers. These are concatenated, and linearly transformed as seen in Equation 26

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \mathrm{head}_2, \ldots, \mathrm{head}_h)W^O \tag{25}$$

$$\text{where } \mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{26}$$

In Equation 26, QW_i^Q, KW_i^K, VW_i^V, is the linear projection matrices.

Figure 12a shows the transformer architecture as designed by Vaswani et al., 2017. It shows how the overall architecture looks, and how the encoders and decoders work. Firstly the input is embedded, which is a vectorization algorithm, the positional encoding is added to indicate where in the input a word is. This means that the same word in different places in the input will be embedded differently.

In Figure 12b the encoder architecture is depicted. The encoder within the transformer model uses a stack of identical layers, each consisting of two main sub-layers: the multi-head self-attention, and a feed-forward network. Multi-head attention makes it possible to handle inputs in parallel in the encoder, and it makes it possible to focus on different parts of the input independently. After this, the feed-forward network transforms the data received from the multi-head attention layer. The next step "Add & Norm" combines the output and normalizes the result, stabilizing the learning process. In the stack of encoders, the output of one encoder is used as the input for the next encoder, creating a complex representation of the original input.

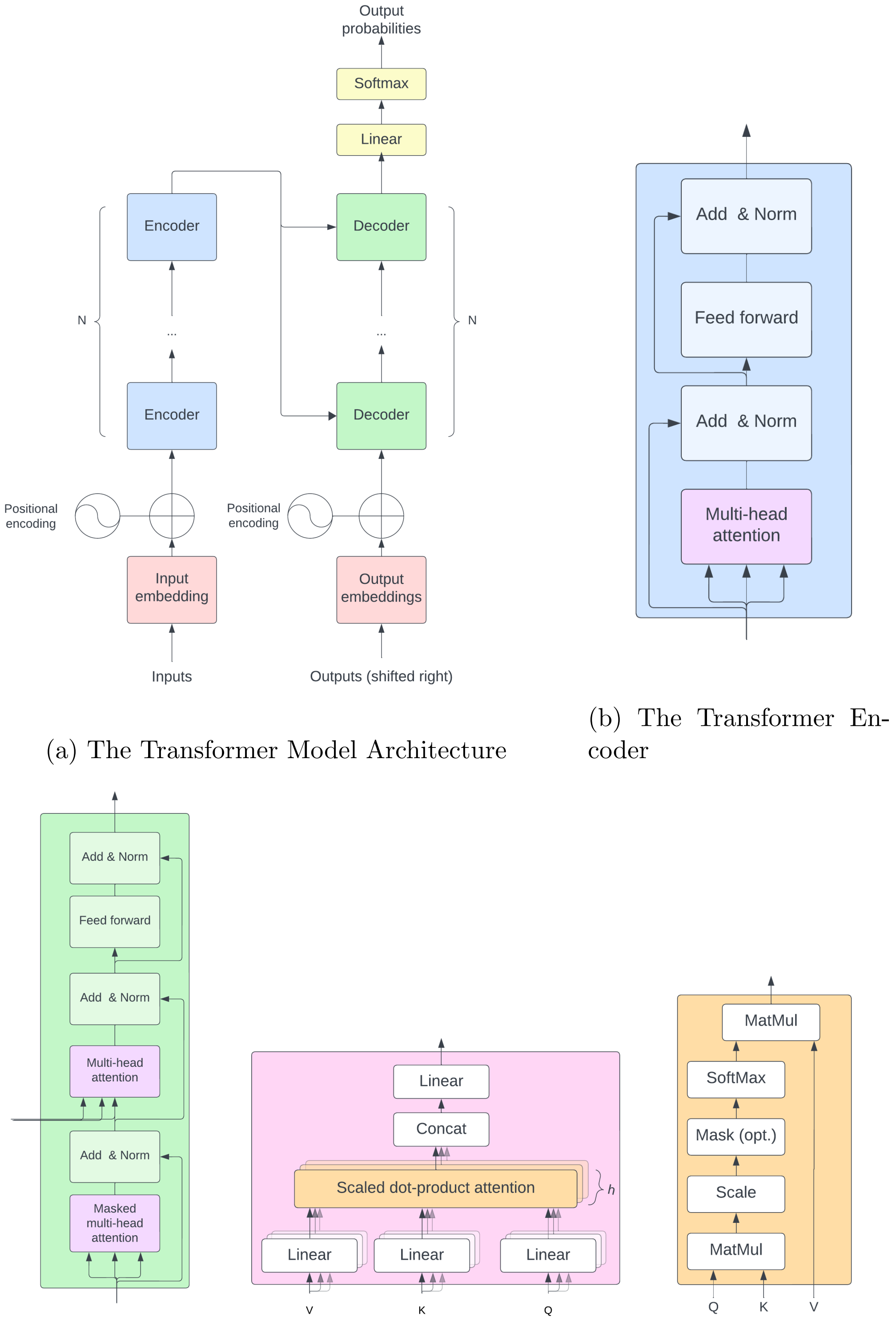
In Figure 12c the decoder architecture can be seen. The decoder is similar to the encoder in structure, but with one main difference to make sequence generation possible. The decoder starts by using masked multi-head self-attention, which makes sure that future words from the sequence are hidden from the decoder. The masking makes sure that predictions for a position only depend on already known outputs before it. After this, the multi-head attention layer is used to focus on relevant parts of the output of the encoder. The "Add & Norm" step normalizes in this case, and the feed-forward network enables the decoder to produce relevant sequences, one element at a time. The output sequences are generated using the full data of the encoder's input, and the part of the output sequence that is already generated.

After the final decoder layer, the output is passed through a linear layer. The linear layer projects the decoder's high-dimensional output into a space that has the same dimensionality as the model's vocabulary. This process is needed because the decoder's output is high dimensional, but for humans to understand the output needs to be interpretable as words or tokens. The last layer used is a softmax layer, which converts the raw predictions made by the linear layer, to a probability distribution. Each element of the softmax vector represents the model's probability of that a corresponding token to be the next element in the output sequence, and the token with the highest probability is selected.

Text-to-Text Transfer Transformer (T5)

T5, also known as Text-to-Text- Transfer Transformer, extends the transformer model by framing NLP tasks as a text-to-text problem (Raffel et al., 2020). This means that it is specialized in taking text as input and producing text as output, examples of problems like this are summarizing and translation. T5 comes in five different sizes: small, base, large, XL, and XXL. The bigger models generally allow for better results but for the cost of computational complexity for fine-tuning.

T5 is already pre-trained using "unsupervised pre-training", where the model is trained on a large corpus without any specific objectives (Raffel et al., 2020). This makes T5 able to learn and understand language and context. T5 is then "fine-tuned", which is supervised training, where it is trained on a specific task. This allows the model to in the first stage understand language, and in the second stage enables this knowledge to be used to solve specific tasks.



(a) The Transformer Model Architecture

(b) The Transformer Encoder

(c) The Transformer Decoder

(d) Multi-Head Attention Block

(e) Scaled Dot-Product Attention Process

Figure 12: Overview of the Transformer Model Architecture and Components - Author's own illustration - based on Vaswani et al., 2017

3 Methodology

This section describes the methodologies employed in this research, exploring how ML can be utilized within construction safety. The section contains a literature review, looking into both construction safety and fields within AI. The data undergo a thorough EDA and cleaning. This study utilizes three distinct ML algorithms, each designed for a specific RQ related to JSA. The ML models aim to assess the quality of JSAs, identify potential hazards, and generate preventive measures.

3.1 Literature Review

The literature review in this paper is divided into three main categories. The first category focuses on safety within the construction industry, with an emphasis on the theory behind JSA. The second category looks into the applications of ML in improving construction industry safety. The final category examines more technical aspects, exploring methods in ML and in DL.

The literature review was conducted in two distinct steps. First, a search prompt was used to identify relevant papers within academic databases. The relevance of each paper was determined after reviewing its abstract. The two search engines used for this thesis were Google Scholar and NTNU's digital library, Oria. Both search engines provide the possibility to filter results based on whether it is peer-reviewed, ensuring papers are reviewed by scholars.

Search Query	No. of hits (Oria)
"Job Safety Analysis"	123
"Job Hazard Analysis"	61
"Job Safety Analysis" + "construction safety"	23
("Job Safety Analysis" OR "Job Hazard Analysis") AND "Machine learning"	1
"Construction safety" AND "machine learning"	117
"Safety Performance" AND "machine learning"	86
"Safety Performance" AND "Natural Language Processing"	3
"Construction safety" AND "Natural Language Processing"	26
"Long short term memory" AND "Text data"	136
"Long short term memory" AND "Classification"	5 818
"Transformers" AND "Text generation"	108

Table 7: Search Queries used in Literature Review

The selection of papers was based on the following inclusion criteria: (IC1): Availability on either Oria or Google Scholar; (IC2) publication date after 2010; (IC3) written in either English or Norwegian; (IC4) the papers had to be peer-reviewed.

Exceptions to these inclusion criteria were made on occasion. Certain older papers, heavily cited within their field, were included. In addition to this state-of-the-art

Id	Type	Criteria
IC1	Availability	✓
IC2	Time	2010-2023
IC3	Language	English OR Norwegian
IC4	Peer reviewed	✓

Table 8: Inclusion Criteria

technical papers, often published by major technology companies such as Google and Meta, were also considered. These papers are extremely relevant when looking into technical aspects, even though the papers are not always academically published or peer-reviewed.

3.2 DiSCo Project

DiSCo is short for, Sustainable value creation by digital predictions of safety performance in the construction industry, which is a research project funded by the Norwegian Research Council (NTNU, 2023). The project has four industry partners serving as important stakeholders in the research program. The project is managed by the Institute of Industrial Economics at NTNU, and the project's timeline spans from 2021 to 2025.

The aim of the DiSCo project is to gain insight into how AI can be used at the early stages of construction projects, to accurately forecast the safety standards of projects. The goal is that the adoption of new technology can help decision-making and lower the number of accidents in the industry.

3.3 Confidentiality and GDPR

The company from where the data was obtained is undisclosed to maintain confidentiality, as the dataset contains sensitive information like personal names. The reason for this confidentiality is to be inline with the General Data Protection Regulation (GDPR). To ensure privacy, the identifiable information in the data has been removed. The ML models used in this thesis do not use any sensitive data. They use data related to hazards and descriptive information within JSA.

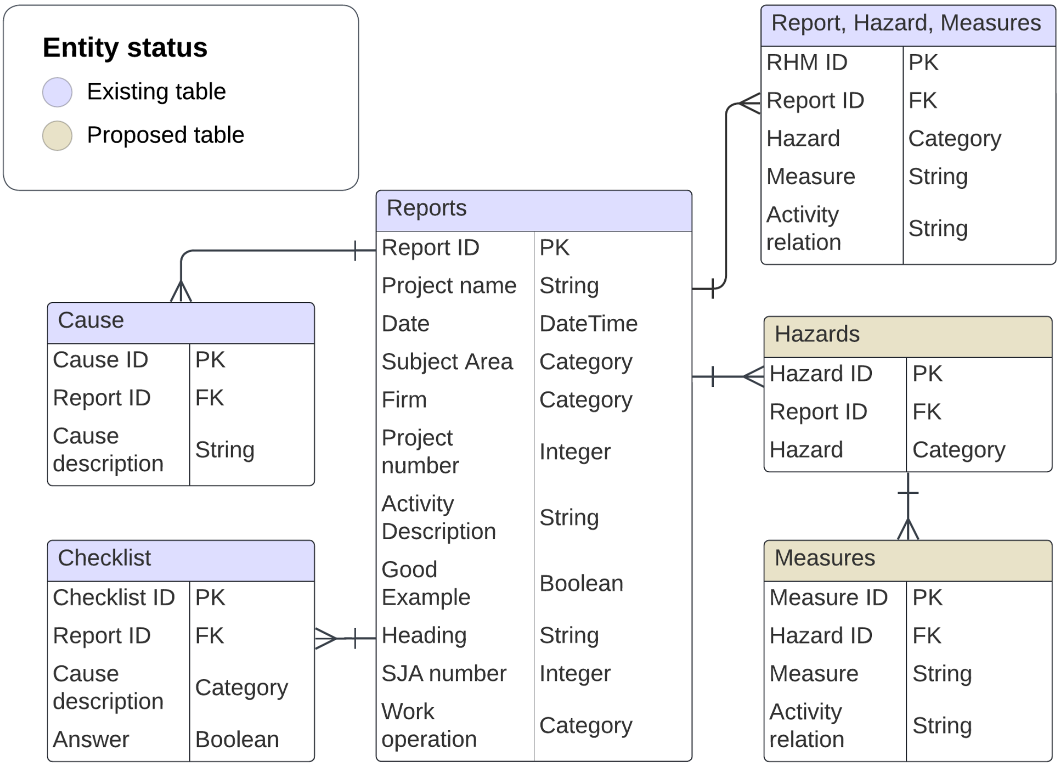
3.4 Exploratory Data Analysis

An EDA is a methodical approach for an in-depth examination of datasets, aiming to gain insights into the content of the data (Chatfield, 1986). This process can help identify patterns, challenges, and potential limitations within the dataset. The primary objective of doing an EDA is to get insights into the data that can enable well thought out decision-making in the future usage of the data.

3.4.1 Project Data

This paper uses several datasets gathered from a leading European contractor, specifically from its Norwegian branch, spanning from 2019 to 2023. These datasets include information related to JSA reports.

In this chapter, each data is systematically explored. In the first dataset, referred to as the “Reports”-dataset, each observation represents an individual JSA. Following, the “Reports, Hazards, and Measures” (RHM) dataset is delved into. A notable complexity within this dataset is that a JSA may include multiple hazards, and each hazard may have multiple identified preventative measures. To help clarify, hazards and measures have been divided into two separate datasets. Figure 13 illustrates a simplified data model. This model shows both the original dataset structure and the adjusted structure made for clarity. The “Cause” dataset looks into the causes of the creation of JSAs, while the “Checklist” dataset consists of questions asked about the JSA process to ensure a high quality of the JSA. Both the “Cause” and “Checklist” datasets are simplified in the figure for enhanced clarity.



Report, Hazard, Measures	
RHM ID	PK
Report ID	FK
Hazard	Category
Measure	String
Activity relation	String

Reports	
Report ID	PK
Project name	String
Date	DateTime
Subject Area	Category
Firm	Category
Project number	Integer
Activity Description	String
Good Example	Boolean
Heading	String
SJA number	Integer
Work operation	Category

Cause	
Cause ID	PK
Report ID	FK
Cause description	String

Hazards	
Hazard ID	PK
Report ID	FK
Hazard	Category

Checklist	
Checklist ID	PK
Report ID	FK
Cause description	Category
Answer	Boolean

Measures	
Measure ID	PK
Hazard ID	FK
Measure	String
Activity relation	String

Figure 13: Entity-Relationship Diagram of Job Safety Analysis Dataset

A comprehensive description of all columns in the datasets is presented in Table 9. This table includes the dataset in which the column is, the name of the column, a short description of the column, and lastly the data type of the column. The terms “Primary Key” and “Foreign Key” are respectively denoted by the abbreviations PK and FK.

Dataset	Column name	Description	Data type
Reports	Project name	Name of project associated with the JSA	Category
	Date	Date when the JSA was created	Date
	Subject area	Subject area of the JSA	Category
	Responsible firm	Firm responsible for the JSA	Category
	Work operation	Category describing the work operation performed	Category
	Project number	Project ID number	Integer
	Activity Description	Description of the planned activity	Text
	Good Example	Indicator of whether the JSA is considered a good example	Boolean
	Report ID	Unique identified for each JSA	PK
Report, Hazards, Measures	RHM ID	Unique identifier for each measure and hazard found within an activity	PK
	Hazards	Category showing the type of hazard associated with an activity	Category
	Preventative Measure	Preventative measure proposed to mitigate a hazard	Text
	Report ID	JSA report ID that links the hazard and measure to JSA	FK
	Activity relationship	The relation between the activity and the hazard described	Text
Cause	Cause ID	Unique identifier for each cause	PK
	Cause desc	Description of the cause	Text
	Report ID	Links the cause to the JSA report	FK
Checklist	Checklist ID	Unique identifier for each checklist item	PK
	Report ID	Links the checklist item to the JSA report	FK
	Question	The checklist question related to safety measures	Text
	Answer	The response to the checklist question	Boolean
Hazard	Hazard ID	Unique identifier for the hazard	PK
	Hazard	Description of the hazard	Text
	Report ID	Links the hazard to the JSA report	FK
Measures	Measure ID	Unique identifier for the measure	PK
	Measure	Description of the safety measure	Text
	Activity relation	Describes the relation of the measure to the activity	Text
	Hazard ID	Links the measure to the specific hazard	FK

Table 9: Description of Every Variable in the Dataset

3.4.2 JSA-Report Data

This subsection delves into the JSA-report data. Figure 14 displays the chronological distribution of JSAs, plotted by the month. The dataset spans from September 2019 to May 2023, and a notable trend is the increase of the number of JSAs over time.

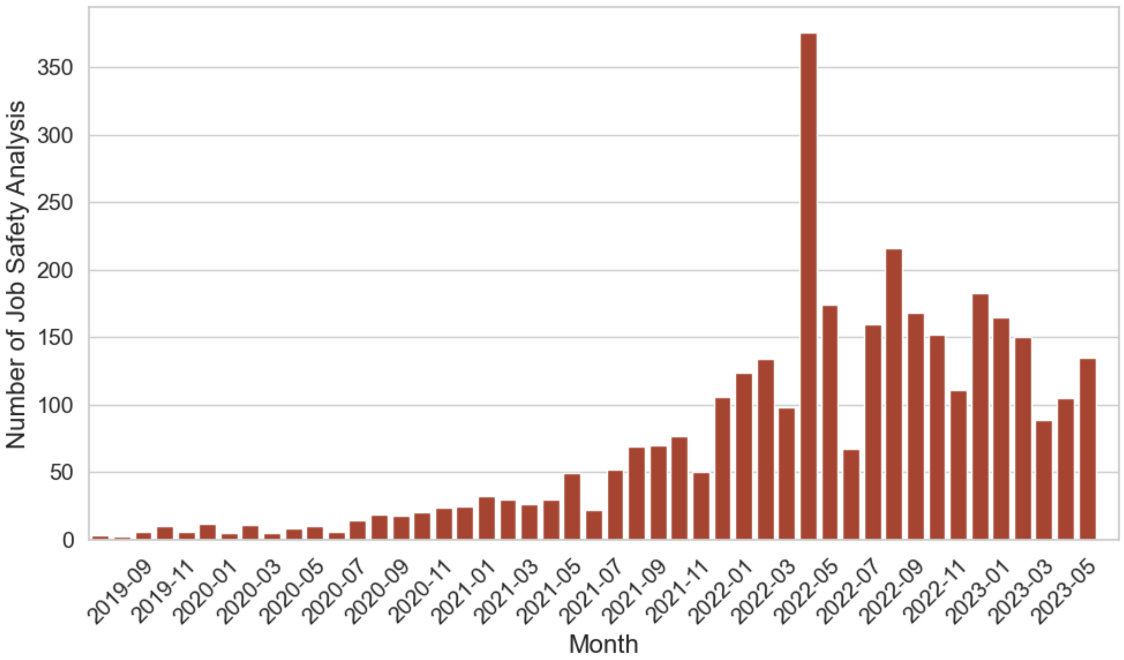


Figure 14: Monthly Distribution of JSAs Over Time

Further analysis is conducted on the proportions of different types of activities. These proportions are depicted in a pie chart presented in Figure 15. The chart reveals a relatively balanced distribution among the different types of activities. The categories in Figure 15 are translated to English, but their original content is available in Appendix B.

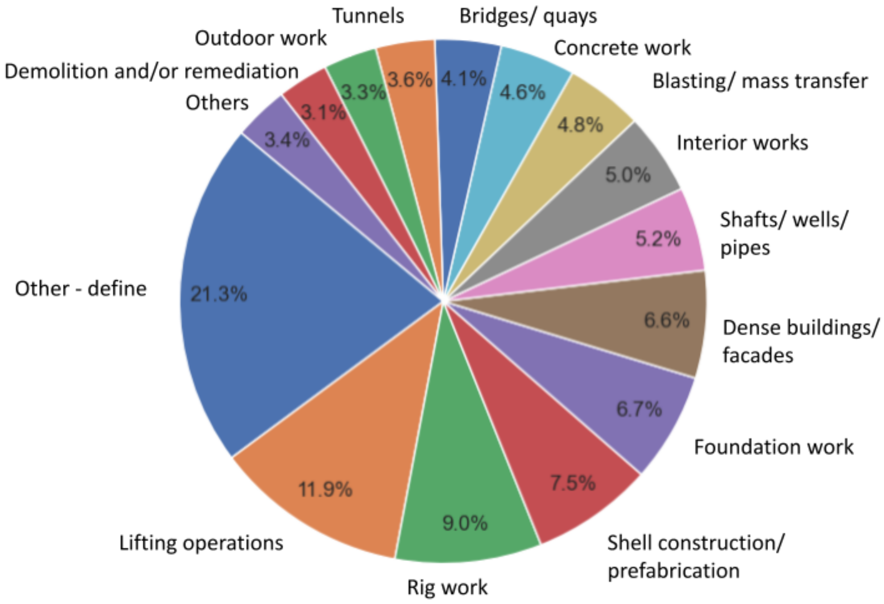
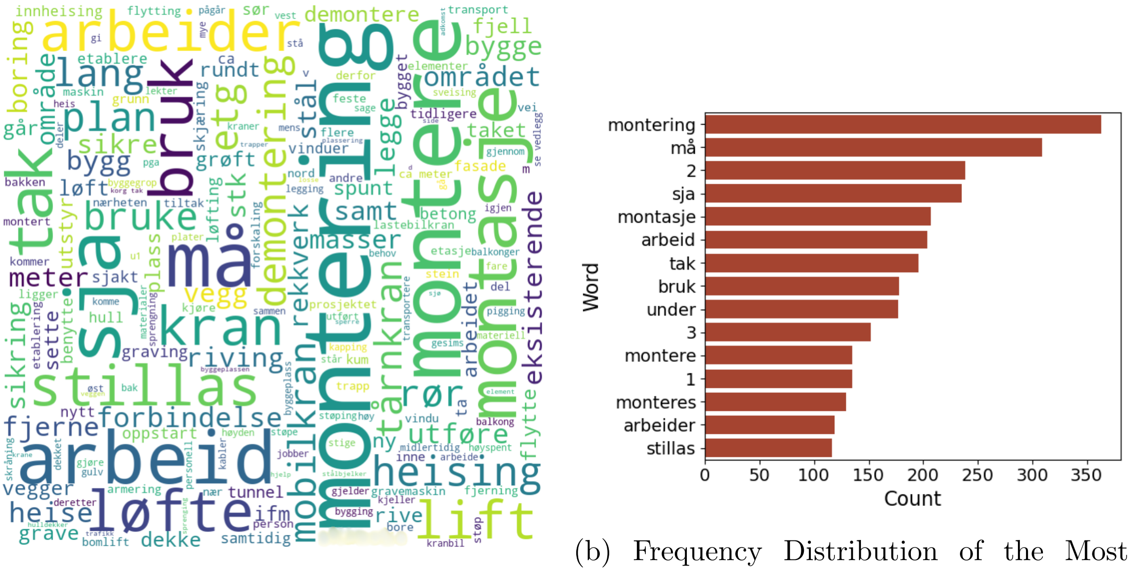


Figure 15: Pie Chart of Different Activity Types in JSA Reports

An observation is that the predominant category of activity is “others-defined”. The fact that the most common category is other presents challenges when using this variable in a ML model, due to the ambiguous nature of the value. Aside from this category, the distribution among the remaining categories is uniform. The “Others” segment, which aggregates all the least frequent categories, only accounts for 3.4 % of the data. Given the even distribution of the other categories, this variable, in conjunction with textual descriptions of the activities, can be used to possibly increase the predictive value of ML algorithms that try to find potential hazards.

Moving on one can plot the most common words in describing the activities. A word cloud of the most common words in activities is shown in Figure 16a. Subsequently, the analysis can include the frequency of word usage in the descriptions of activities. Figure 16 illustrates a word cloud and the most frequent words. It should be noted that the word cloud automatically filters stopwords, while the word count includes stopwords.



(a) Word Cloud of Words used JSA Activity Descriptions

(b) Frequency Distribution of the Most Common Words in JSA Activity Descriptions

Figure 16: Textual Analysis of Words from JSA Activity Descriptions

Figure 16a shows that various forms of the word “Assembly” are the most common in the activity description. Additionally, the terms “hoisting” and “lifting” are widely used, alongside “scaffolding” and “crane”. These findings show that the dataset is collected from the construction industry, because of its frequent use of construction industry-specific terminology. The frequency of the most common words within the dataset is depicted in Figure 16b. This depiction supports the findings in the word cloud. Hence, the text from the Figures confirms that the text is related to the construction industry.

Another attribute that should be examined closer is the length of the activity descriptions, quantified by the number of characters. Figure 17 shows the distribution of text lengths by the number of characters. The majority of the descriptions are about 50 characters long, which makes them short and concise. While some descriptions are more descriptive, reaching 500 characters, these are outliers and not the norm.

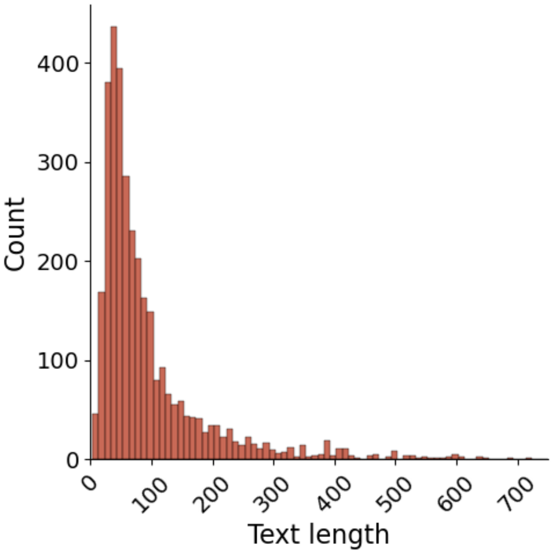


Figure 17: Character Count Distribution of JSA Activity Descriptions

3.4.3 Hazards Data

The following analysis focuses on the hazards dataset. A pie chart categorizes the nine most common hazards, with the remaining hazards aggregated to an “Others” category, as depicted in Figure 18. In Figure 18 translated hazards are used, but the original content is available in Appendix C

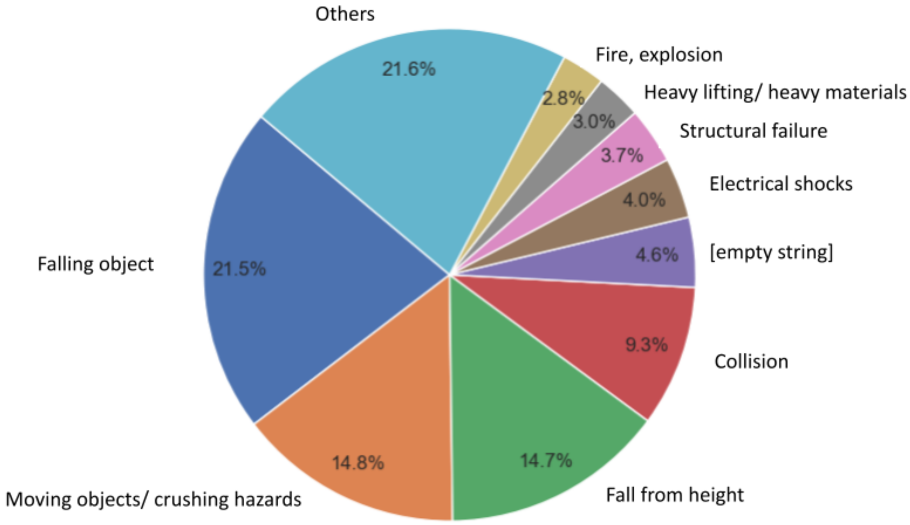


Figure 18: Pie Chart of the Most Common Hazards

Looking at the pie chart, it can be observed that the three most common hazards are falling objects, moving objects/ crush hazards, and falls from height. There is significant variation in the distribution of hazard categories, with some being more common than others.

Each JSA may have multiple hazards associated with it. To understand the relationship between JSA and the hazards identified, Figure 19 illustrates the distribution of the number of hazards identified per JSA. The identification of three hazards per JSA is the most common, with the occurrence of one and two hazards being identified being almost as common. After three identified hazards, there is a noticeable decline in the number of hazards identified per JSA.

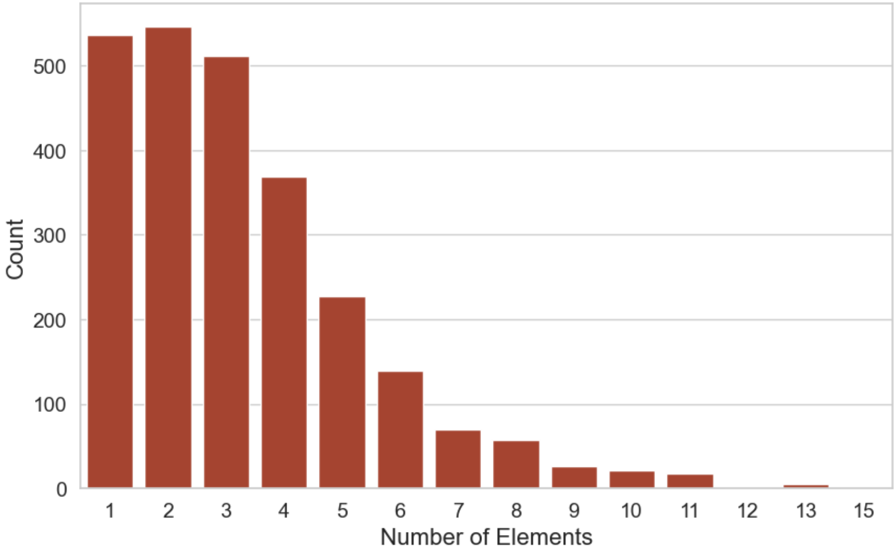


Figure 19: The Distribution of Identified Hazards per JSA

When analyzing the hazards identified within JSAs, it is important to remember that an activity may be associated with multiple hazards. Figure 20 depicts a confusion matrix that gives insight into which hazards often are identified within the same activity. A pattern emerges from the confusion matrix, which is that the identification of “falling objects” often occurs simultaneously with the identification of “fall from height”. This correlation is logical since activities performed at elevation increase the risk of both persons falling and falling objects. Similarly, there is a correlation between the presence of “falling objects”, and “moving objects/crush hazards”, which makes sense since the hazards are related. “Collision” hazards also tend to accompany “falling objects”. The relationships between the identified hazards can suggest that aggregating hazards into broader categories may improve the performance of ML algorithms when trying to identify hazards.

3.4.4 Preventive Measures Data

The EDA proceeds with an analysis of the “preventive measures” data. Firstly, the distribution of the number of preventative measures associated with each identified hazard can be seen in Figure 21. The visualization shows that it is most common for a single preventive measure to be identified for a hazard, however there are some instances where multiple preventative measures are identified to a hazard.

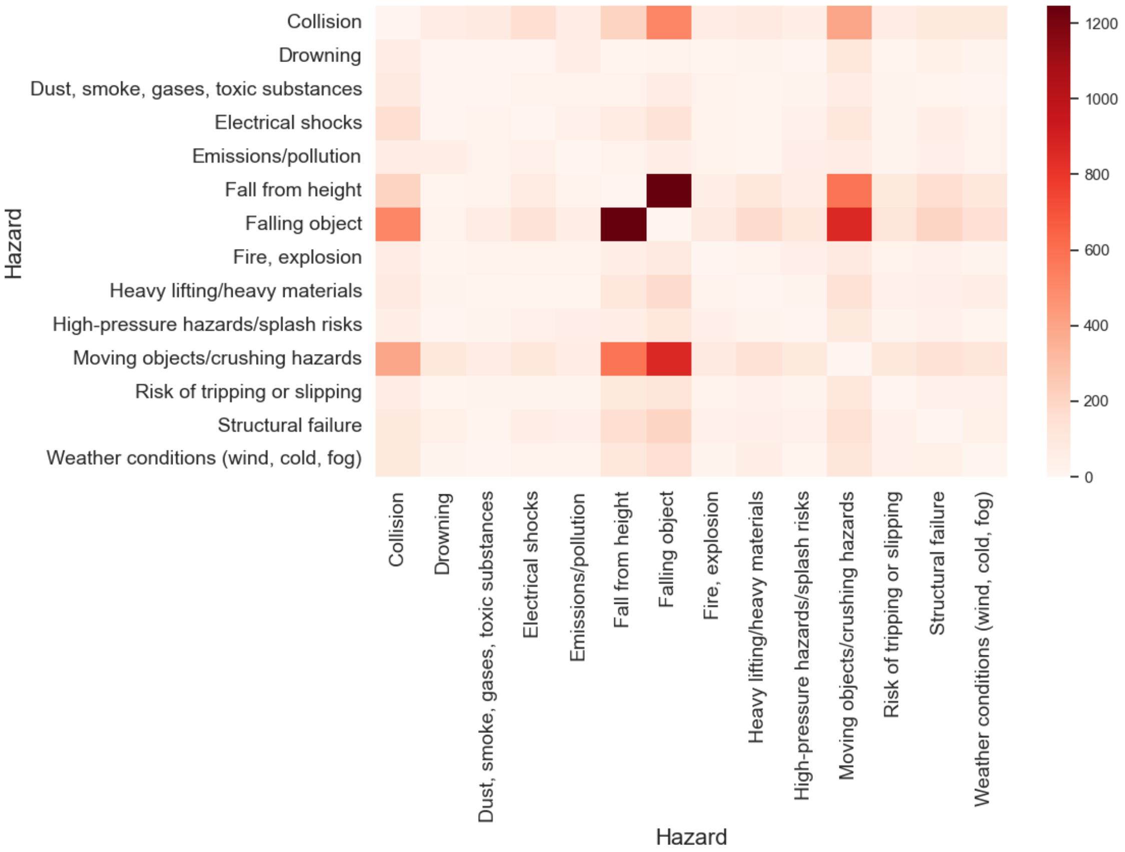


Figure 20: Heatmap of Hazards Co-occurrences in JSA

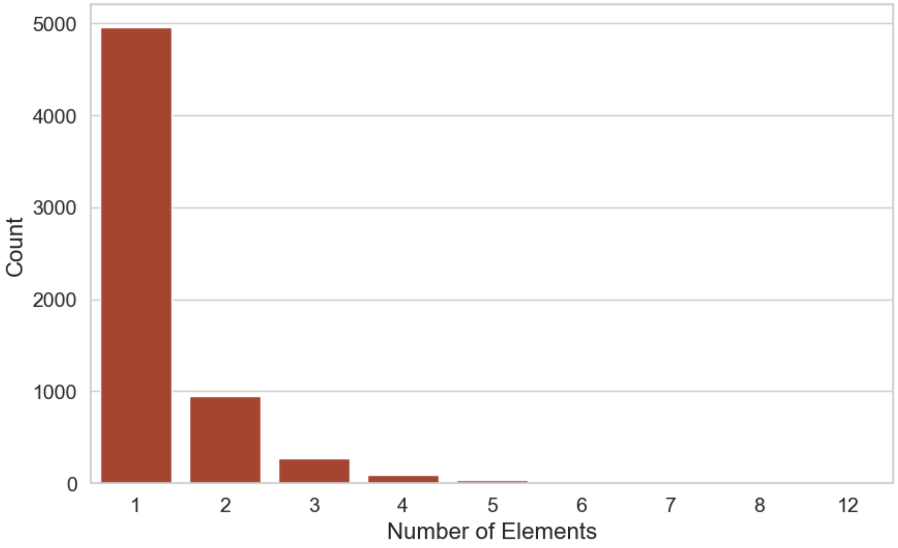


Figure 21: The Distribution of Preventative Measures Identified per Hazard

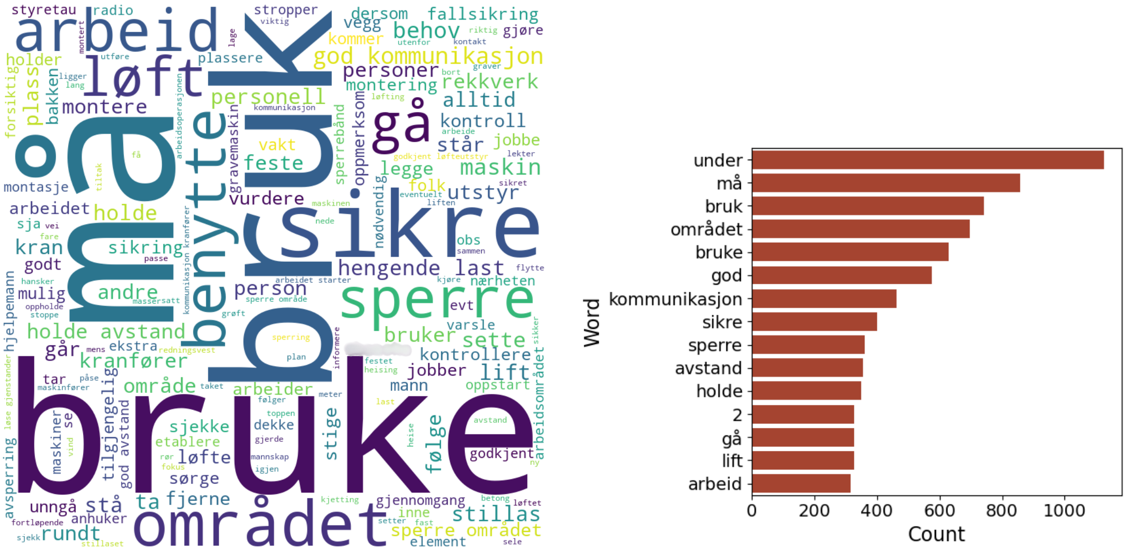
In Table 10, the number of JSA samples is shown, along with the number of hazards identified for those JSAs, and the number of measures identified for those hazards.

Type	Number of Observations
Job Safety Analysis	3425
Hazards identified for JSA	6313
Measures identified for Hazards	8319

Table 10: The Number of Observations of JSA, Hazards, and Preventative Measures

Text analysis measures

Examination of the word cloud generated from the preventive measures text reveals clear differences compared to the activities text data. In Figure 22a, the Norwegian word “bruke”, meaning “use” seems like the most common. Additionally, the word “must” is used frequently. In Figure 22b the most common observations are listed with their frequency. Notably, adjectives associated with preventive measures include words such as “use”, “secure”, and “block”, likely describing the actions used as preventative measures.



(a) Word Cloud of Words used in Preventative Measures

(b) Frequency Distribution of the Most Common Words in JSA Measures

Figure 22: Textual Analysis of Words from Preventive Measures in JSA

Figure 23 shows the character count for each preventative measure. Compared to the activity descriptions, the character lengths of the preventive measures are more evenly distributed. The most common length of a measure description is about 60 characters long, though the average length is likely higher. This suggests that JSA creators provide a more detailed description of preventive measures compared to the activities themselves.

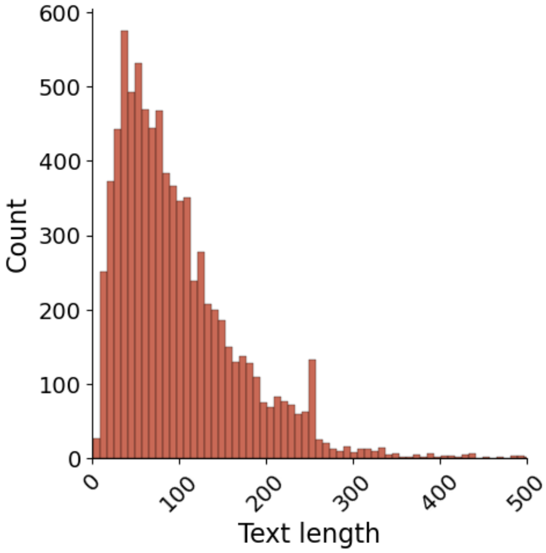


Figure 23: Character Length Distribution of Preventive Measures

3.4.5 Causes of Creating JSA

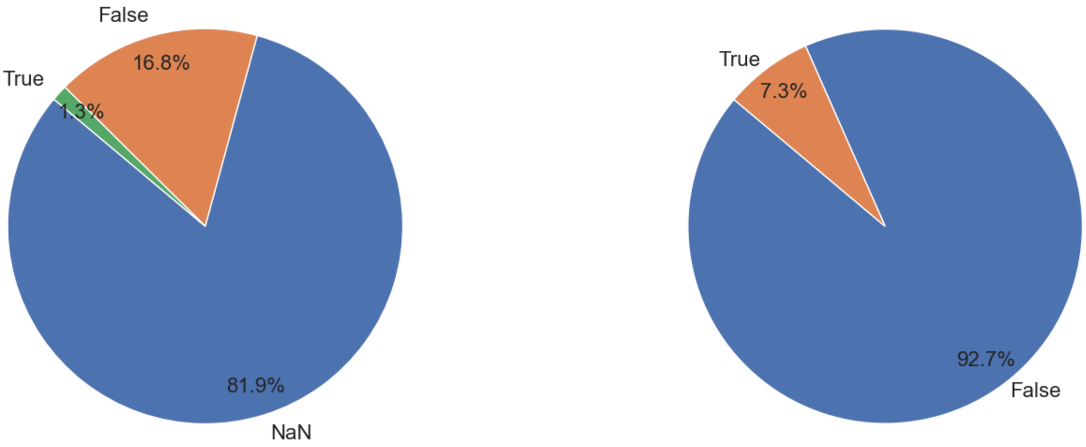
The dataset includes details on the motivations for conducting the JSA. Table 11 shows these reasons with the rates at how often each of them occurs, with the original Norwegian text being available in Appendix D. The three primary reasons for initiating a JSA are; “The consequences are serious if something goes wrong”, followed by “The work involves a risk of health damage/injury risk”, and lastly “Accidents or unintended incidents have occurred in the past during similar activities”. The two first reasons stem from the potential severity of the risks involved, whereas the last reason is a proactive approach to learn from historical incidents.

Reason	Count
The consequences are serious if something goes wrong	1473
The work involves a risk of health damage/injury:	1415
Accidents/unwanted incidents have occurred previously in similar activities:	1240
The activity is new and unknown	568
People who do not know each other have to perform a critical job together	522
Prerequisites for the activity have changed (weather conditions, time, sequence, other activities nearby):	342
The work entails deviations/changes from descriptions in procedures and plans:	313
Need to provide and document equipment-specific training:	221
Other:	110

Table 11: Reasons of Initiation of JSA

3.4.6 The Good Example Variable

The dataset includes a column that denotes whether the identified hazards and preventative measures associated with an activity are considered a “Good Example”. Given one of the RQs is to find indicators of the quality of a JSA, a detailed examination of this variable is fitting. Figure 24 displays the distribution of Not a Number (NaN), False, and True values within the variable.



(a) Proportion of “Good Example” Assessments Including NaN Values

(b) Proportion of “Good Example” Assessments Excluding NaN Values

Figure 24: Distribution of “Good Example” Assessments in JSA

Figure 24a reveals that a substantial amount of JSAs lack labels. This presents questions about whether the NaN values should be categorized as negative examples or be discarded as “unlabeled”. The following analysis will look at how JSAs labeled as “Good Examples” distinguish them-self from JSAs labeled as “False”, while the JSAs labeled with NaN is disregarded.

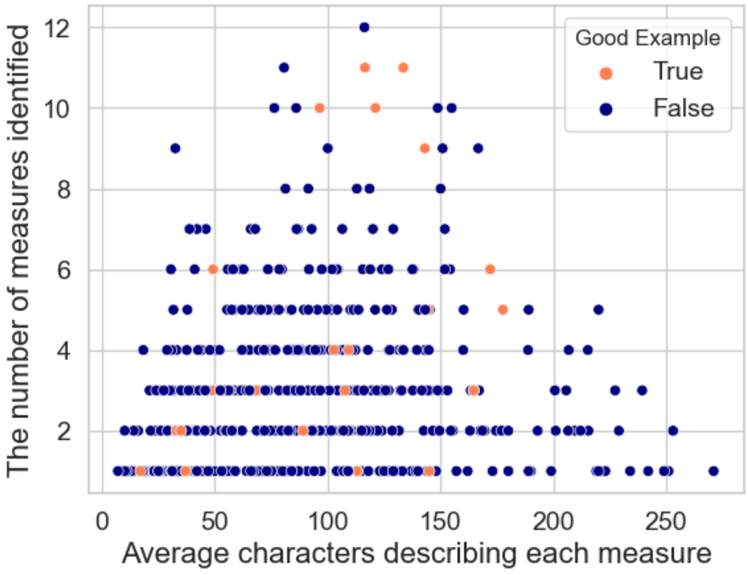
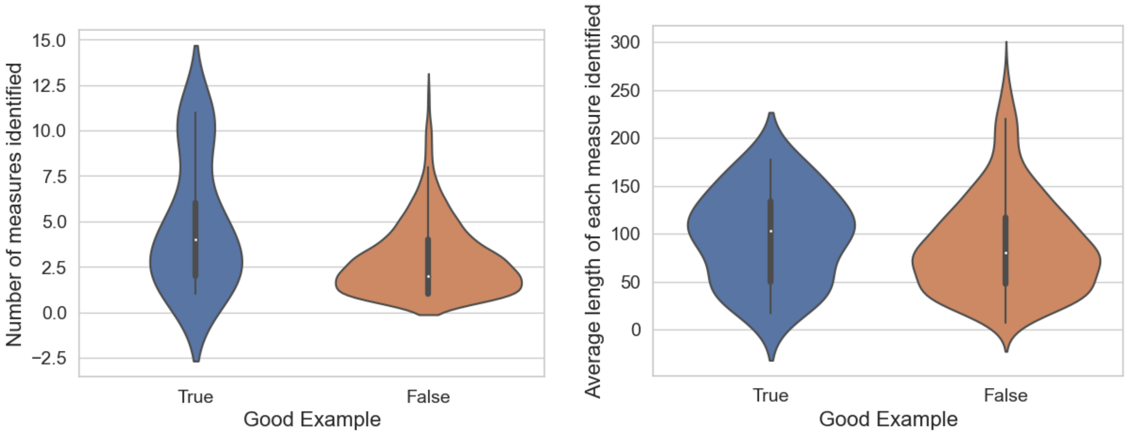


Figure 25: Scatter Plot of JSA “Good Example” classifications by the Number of Measures and Their Lengths

Observations of Figure 25 suggest that JSAs are more likely to be considered a "Good Example" with a greater number of measures identified. On the other hand, the average character length of measures seems to have a smaller effect on whether a JSA is a "Good Example" or not. A more detailed exploration of this relationship can be seen in Figure 26. From this figure it seems like a "Good Example" on average has a higher number of identified measures, and longer lengths of the measures descriptions. The number of measures identified seems considerably higher for the good examples, while they seem to have slightly more characters.



(a) Distribution of Number of Measures by "Good Example" Variable

(b) Distribution of Average Length of Measures by "Good Example" Variable

Figure 26: Violin Plots of the "Good Example" Variable

3.5 Data Cleaning

In this subsection, the methodologies used for data cleaning are presented. The cleaning of each dataset is explained in detail.

3.5.1 JSA-Reports Data

First, looking at the JSA reports data the dimensionality is that there is 3425 different observations, each with 22 variables captured in Table 12.

Columns	Rows
22	3425

Table 12: Dimensionality of Reports Data

Removal of JSAs containing the string "Test"

In the dataset, there are many activity descriptions which are called "Test". To clean the dataset, a query for the term "Test" was used, to find that 39 observations contained the word. Looking closer at these observations one can see that many of them are attempts to test system functionality, while others contain the word "test" in a contextually relevant way. Therefore, a manual review was conducted to differentiate between noise and meaningful data. Table 13 gives an example of entries that were typically removed and entries that have been kept.

Text content of row	Include in Analysis?
[Anonymized] is energized and **test** driving is ongoing on Tuesdays, Thursdays, and occasionally on Fridays. Work in/near the track requires an application. Travel on the track does not require an application	Yes
Test description	No

Table 13: Examples of Activity Descriptions Containing the String "test"

Detecting the language of activity description

The dataset examination revealed numerous observations consisting of random keystrokes, which need to be filtered to ensure high data quality. This thesis approaches the issue by labeling each activity description with a language. The outcomes of doing language labeling are shown in Table 14.

Language	Count
Norwegian	3164
Danish	99
English	44
Swedish	27
German	11
Indonesian	10
Tagalog	7
Dutch	7
Afrikaans	5
Unknown	4

Table 14: Language Distribution Among Activity Descriptions

In addition, the dataset contained 15 other languages, each with three or fewer instances. The entries labeled as Norwegian were assumed to be informative and therefore kept. Since language detection models often struggle with distinguishing between Norwegian and Danish due to the similarity in the languages written form, danish-labeled descriptions were also kept.

Considering only 44 English-labeled instances, the decision to remove these from the dataset was made, since ML algorithms need input in the same language. The Swedish samples were kept since some were mislabeled, and a few of them seemed like Swedish workers working in Norway. The rest of the languages were reviewed manually. Those incorrectly labeled, but written in Norwegian were kept. While the entries that were random keystrokes and in other languages were removed from the dataset.

Text Preprocessing: Tokenization, Special Character Removal and Lower Casing

The preprocessing started with tokenization, where the text was segmented into individual words. Then special characters, such as &, *,], were removed from the text strings. Finally, all text was converted to lowercase to ensure uniform word recognition in future ML processes.

Stopword removal

Following these techniques, stopwords were removed from the text. This was done by using a predefined list of Norwegian stopwords from the Natural Language Toolkit, where any words present in the list are removed from the tokenized text.

In preprocessing, stopword removal was selectively applied based on the type of ML model being used. When using LSTM networks, and traditional ML algorithms, stopwords were removed sometimes and kept sometimes. This was to test if the model would benefit from the removal of redundant words or not. During the training of the generative AI model stopwords were kept at all times, since transformer models have a complex and advanced contextual understanding, which yields better performance when having the full textual content. The reason for this is that transformer models are often trained on large amounts of textual data, and the text used to fine-tune to model, should have the format as the model was trained on.

3.5.2 Preventive Measures

The text data for preventive measures was processed using the same techniques as those used for activity descriptions. Given that the methodologies are similar, the steps of data cleaning for preventive measures will not be explained in detail.

3.5.3 Hazards Data

The hazards used in this thesis correspond to a list of predefined hazards from the provided data. A full list of these hazards, with the number of occurrences, is shown in Table 15.

Looking at the data, it seems like the JSA system has the option to either choose one of the predefined hazards in Table 15, or define a new hazard with text. A closer examination was performed to determine the proportion of self-identified hazards compared to the predefined list. The findings are that 7279 of the hazards identified align with the predefined list of hazards, while 1687 hazards identified do not match with the predefined list. When looking closer at the unlisted hazards many different unique hazards are found, where the most common is an empty string. Given the need to have many samples per category for ML algorithms, the free-text hazards were removed from the dataset used for classifying hazards. Moving onward in this chapter only hazards which match the predefined list will be used.

Hazard	Count
Falling object	1725
Moving objects/ entrapment risks	1205
Fall from a height	1182
Collision	764
Electrical shocks	325
Structural failure	302
Heavy lifting/heavy materials	243
Fire, explosion	227
High pressure, splash hazard	201
Risk of tripping or slipping	199
Weather conditions (wind, cold, fog)	181
Drowning	172
Dust, smoke, gases, toxic substances	146
Emissions/pollution	132
Sharp objects (cuts, stabs)	120
Noise, vibration	73
Working in confined spaces/oxygen deficiency	40
Biological health hazard	20
Collapse of excavation pit	14
Surfaces with extreme temperatures (high/low)	8

Table 15: Frequency of Each Hazard in the Dataset

When looking at Table 15 it can be seen that the different hazards are unevenly distributed. To make sure that every class has enough samples the hazards with less than 100 occurrences were removed. As a consequence of this, the following hazards did not meet the criterion and were removed:

- Noise, vibration
- Working in confined spaces/oxygen deficiency
- Biological health hazard
- Collapse of excavation pit
- Surfaces with extreme temperatures (high/low)

3.6 Machine Learning

To address the RQs in this thesis, three different ML algorithms were designed. The first algorithm is called the Good Example Classification algorithm, the second algorithm is called the Hazard Classification Algorithm, while the third algorithm is called the Preventative Measures Generation Algorithm.

3.6.1 Algorithm I: Good Example Classification

The Good Example algorithm is designed to determine the probability of a JSA qualifying as a “Good Example”. This task is framed as a classification task, where the goal is to assess the potential of a JSA to meet “Good Example” criteria. The algorithm outputs a value ranging from 0 to 1, representing the likelihood of that given JSA being in the “Good Example” category. To address this classification challenge numerous ML models were utilized, including RF, Gradient Boosting (GB), and LSTM.

$$\text{Activity description} + [\text{Preventative measures}] \mapsto \text{Probability of “Good Example”} \tag{27}$$

The model has to address the issue of a significant imbalance in classes as seen in Figure 24b. To fix this problem, two techniques were used, which are SMOTE and the use of weights, both of which are explained in detail in sub-chapter 2.4.5.

Different vectorization techniques were used to identify the optimal solution, including TF-IDF and Word2Vec. TF-IDF, a well-established method, converts text data into numerical values, while Word2Vec represents a more advanced vectorization approach, utilizing embeddings. A detailed explanation of both methods is presented in subchapter 2.3.4. In addition to using the text data, quantitative data was also used as input to ML models. These quantitative measures included the average length of preventative measures and the number of identified measures. The aim of using different data as input to models was to see if quantitative data alone could give accurate predictions, or if text data was needed. In the end, the three different types of data were used in the training; quantitative data, TF-IDF vectorized data, and Word2Vec embedded data.

During the ML models training, only a test set and a training set were utilized. The reason for not using a validation set is due to class imbalance and the few positive examples of “Good Example”. Splitting the dataset into three parts (training, validation, and test) instead of two could potentially increase the difficulties associated with class imbalance.

To prevent overfitting when only using a training set and test set, cross-validation was employed. The cross-validation algorithm used is K-folds cross-validation with k equal to 5, which is explained in sub-chapter 2.4.2. Pseudo code for Algorithm I: Good Example Classification is given in Algorithm 1.

Algorithm 1: Pseudo-code for Algorithm I: Good Example Classification

```
Data: Dataframe containing a list of measures, activity description, good
      example variable
Result: Model performance (Accuracy, Precision, Recall, F1 Score, ROC
        Curve, Confusion Matrix)
begin
    /* Data Preparation and Feature Engineering                    */
    Clean and prepare text data; Calculate text length and other numerical
      features
    Group and aggregate data by "rapport_id"
    Apply TF-IDF Vectorization and Word2Vec to process text data
    Scale numerical features using MinMaxScaler
    Combine text and numerical features
    /* Model Training and Evaluation                               */
    Initialize models (Random Forest, Gradient Boosting, LSTM)
    for each model do
        /* K-Fold Cross-Validation                                 */
        Set up K-Fold cross-validation
        foreach fold do
            Split data; Apply SMOTE if needed
            if model is LSTM then
                Tokenize and pad text sequences
                Define LSTM model with Embedding, LSTM, and Dropout
                  layers
            Fit model; Predict on test data
            Calculate performance metrics; Store ROC curve data
        Plot mean ROC curve; Calculate and display average metrics
    /* Results Aggregation and Display                             */
    Display aggregated confusion matrix
    Print average metrics across all folds
```

For the hyperparameter optimization of the Good Example algorithm, random search was selected as the preferred method. The specific parameters and ranges used for the search, and the hyperparameters used for the Word2Vec algorithm and LSTM are shown in Appendix E.

3.6.2 Algorithm II: Hazard Classification

The aim of the Hazard classification algorithm is to classify activity descriptions and work operations according to possible hazards. This classification process is represented in Equation 28.

$$\text{Activity Description} + \text{Work Operation} + \text{Subject Area} \mapsto \text{Hazard} \tag{28}$$

A challenging aspect of assigning different hazards to activities is the possibility of an activity containing an arbitrary amount of hazards. This type of problem is recognized as a multi-label classification issue and needs to be addressed. In this thesis, the multi-label classification problem is transformed into a series of binary classification problems using binary relevance. This is described in subsection 2.4.4, and as a consequence the model evaluates each hazard independently, classifying it as either True or False.

Since the hazard classification algorithm must be trained separately for each of the 15 hazards, LSTM was the only classification technique used. The choice of LSTM was made since it is viewed as a start-of-the-art approach, and had the best results for the previous classification algorithm classifying if a sample is a “Good Example”.

The transformation of the problem into 15 distinct binary classification tasks results in mostly imbalanced datasets. In most cases, an activity is not associated with a specific hazard, which leads to an over-representation of negative cases. This issue is similar to the challenges faced with the Good Example Algorithm. Therefore, the dataset was divided into only a training set and a test set, excluding a validation set. The reasoning is the same as for the good example algorithm, and in the same way as that algorithm cross-validation was used to stop overfitting. In addition, dropout and batch normalization are tools used with the LSTM model to prevent overfitting.

There was a problem during multi-label classification tied to class imbalance. For example, the “Falling object” class appeared ten times more often than “Drowning”, which leads to a biased model. Usually, a threshold is set, often at 0.5, if the class prediction is above that the sample is deemed as positive. Due to some classes coming more frequent than others, the model became overconfident in its predictions for classes with many samples, frequently assigning a positive label, while predicting a negative label often for rarer hazards. To avoid this, a custom function was created to determine unique thresholds for each hazard by optimizing the F1-score on the training set. This approach aimed to balance the model’s predictive performance across all types of hazards, ensuring that all hazards had positive and negative predictions.

To hinder overfitting the hyperparameter optimization was intentionally limited. Using extensive hyperparameter optimization could potentially improve the results, but without a validation set, the improvements could be due to overfitting, since the hyperparameters are optimized for the test set. Therefore there is likely room for improvements in the model by optimizing the hyperparameters further, and this study prioritizes a true representation of the model’s capabilities over an artificially inflated performance.

Pseudo code of the ML model can be seen in Algorithm 2, and the hyperparameters used to develop the Algorithms can be found in Appendix F.

Algorithm 2: Pseudo-code for the Hazard Classification Algorithm

```
Data: Dataframe containing activity descriptions, work operation, list of
      potential hazards
Result: Model performance (Accuracy, Precision, Recall, F1 Score, ROC
        Curve, Precision-Recall Curve, Confusion Matrix)
begin
    /* Preprocessing                                                    */
    Tokenize text data and pad sequences
    One-hot encode categorical data
    Binarize target labels
    /* Model Training and Evaluation                                    */
    for $(train\_index, val\_index) \in KFold(n\_splits = 5)$ do
        Split data into training and test sets
        Define and compile LSTM model with Embedding, Bidirectional
          LSTM, Dropout, BatchNormalization, and Dense layers
        Fit model on training data using early stopping
        Predict on test data
        /* Metrics Calculation                                          */
        Calculate precision, recall, F-1 score, and accuracy for each label
        Compute TP, TN, FP, and FN
        Aggregate metrics across all folds
        /* ROC Curve Analysis                                           */
        Compute and store ROC curves for each label
        /* Precision-Recall Curve Analysis                              */
        Compute and store Precision-Recall curves for each label
    /* ROC and PR Curve Interpolation and Plotting                      */
    Compute mean FPR values
    for each label do
        Interpolate TPR values
        Compute mean AUC
        Plot mean ROC curve and individual ROC curves for each fold
        Interpolate precision values at common recall points
        Calculate and plot mean Precision-Recall curve
    /* Confusion Matrix Visualization                                   */
    for each label do
        Create and display confusion matrix using heatmap
    /* Results Aggregation and Display                                  */
    Compute and display average metrics for each label
```

3.6.3 Algorithm III: Preventative Measures Generation

The Preventative Measure generation algorithm is designed to suggest preventative measures for given pairs of activity descriptions and hazards. This is a text-to-text task where both input (activity description and hazard) and output (preventative measure) are in text form. For this purpose, the transformer-based model T5 (Text-to-Text Transfer Transformer), made by Google was used. This is a LLM capable of generating highly realistic text. The latest version of T5 is called FLAN, and due to the high computational cost associated with fine-tuning and training a LLM, the two smallest versions were used, **google/flan-t5-small** and **google/flan-t5-base**.

There were challenges encountered with the language of the training data for the algorithm. Initially, the model was trained on the original text, since the documentation states that the model is multi-lingual and capable of processing Norwegian text. However, the early iterations of the model were confused with the Norwegian language and produced outputs in a mix of languages. Therefore the decision to translate the text from Norwegian to English was made. While translation can result in the loss of some information, it seems like the LLM has mostly been trained on English data, making translation appear beneficial.

When training the model, an important decision involved determining the format of the input data. The approach chosen was to frame the task as a Question-Answer problem, where the "question" would consist of the activity description and hazard, while the "answer" would consist of the corresponding preventative measure. For this purpose, the input data was merged into a string that took the form of a question, which can be seen in the next paragraph.

> For the construction activity **"Activity Description"** with the hazard **"Hazard"**, what preventative measure should be taken?

Before training the model the dataset was divided into three distinct subsets: a training set, a test set, and a validation set. The test set was isolated and not exposed to the model during the training phase. This ensures that the examples in the Results are from data the model has never seen, providing an accurate measure of how good the model is at generalization.

The model was trained using IDUN, a supercomputer at NTNU. The processing was allocated to the CPU queue, and the runtime of the small model was 13:59:15, while the base model recorded a run time of 35:38:43. The training involved fine-tuning the T5 model to adapt to the specific task of generating preventative measures, described in more detail in subsection 2.5.4. After that, the fine-tuned model was employed to produce responses to the test data.

Pseudo code for the preventative measures generation algorithm can be seen in Algorithm 3, and the hyperparameters used to create the Algorithm can be found in Appendix G.

Algorithm 3: Pseudo-code for Algorithm III: Preventative Measures Generation

```
Data: Dataframe containing activity descriptions, hazard and list of
      preventive measures
Result: Generated measures for given activities and hazards
begin
    /* Data Preparation                                              */
    Load dataset
    Clean text data
    Translate all text data
    /* Model Initialization                                          */
    Initialize T5 Tokenizer and T5 Model
    /* Data Preprocessing                                            */
    Define a function to format prompts
    Apply the function to create new text data
    Split data into training, validation, and testing sets
    /* Tokenization and Dataset Preparation                          */
    Tokenize training and validation datasets
    Prepare datasets for training and evaluation
    /* Training Configuration                                        */
    Set training arguments like epochs, batch size, logging, and evaluation
      strategy
    /* Model Training                                                */
    Initialize a Trainer with the model, tokenizer, and datasets
    Fine-tune the model on the training dataset
    /* Model Saving                                                  */
    Save the fine-tuned model and tokenizer
    /* Model Loading for Prediction                                  */
    Load the saved model and tokenizer
    /* Prediction                                                    */
    Define a function to generate predictions in batches
    Apply the function to the test dataset to get predicted measures
    /* Result Compilation                                            */
    Add predicted measures to the test dataset
    Save the test dataset with predictions to a CSV file
```

4 Results

This section presents the results of the research. Initially, the results of the good example algorithm are presented, where the algorithm tries to predict if a JSA can be considered a good example or not. Then the results of making a ML algorithm to identify hazards given the description of an activity will be presented. The last section will delve into the results of the preventive measures generation algorithm, which suggests safety measures in response to identified hazards.

4.1 Algorithm I: Good Example Classification

In this subsection, the predictions of the good example algorithm will be presented. The performance of the "Good Example" algorithm is presented in Table 16, which compares the performance of various ML models and data processing techniques. The model with the highest accuracy is uses RF with TF-IDF, and employs weights to address class imbalance. Since the F1-score is 0, it is clear that the model achieves that accuracy by predicting mostly "False" for all entries.

Given this imbalance, the F1-score emerges as the best metric for evaluating model performance, since it balances the Precision and Recall. The results show that the LSTM model outperforms all the other models when looking at the F1-score. Suggesting that DL methods have an edge over more traditional methods when working with text data.

It is interesting to note that models using quantitative data outperform those using text data when using GB and RF models. Another observation that can be made by looking at Table 16 is that the overall performance is poor. It seems like the models struggle to accurately label JSA as a "Good Example" or not.

ML	Data	Balance	Accuracy	Precision	Recall	F1-score
RF	Quant	weights	0.9304	0.0786	0.0786	0.0727
GB	Quant	SMOTE	0.9012	0.1155	0.1821	0.1398
GB	TF-IDF	SMOTE	0.9478	0.0	0.0	0.0
RF	TF-IDF	weights	0.9497	0.0	0.0	0.0
RF	Word2vec	weights	0.9458	0.0	0.0	0.0
RF	Word2vec	SMOTE	0.9188	0.0364	0.1	0.0533
GB	Word2vec	SMOTE	0.8937	0.02	0.05	0.0286
LSTM	Keras	weights	0.9284	0.1515	0.3400	0.1873

Table 16: Performance Metrics of Models Predicting "Good Example" Variable

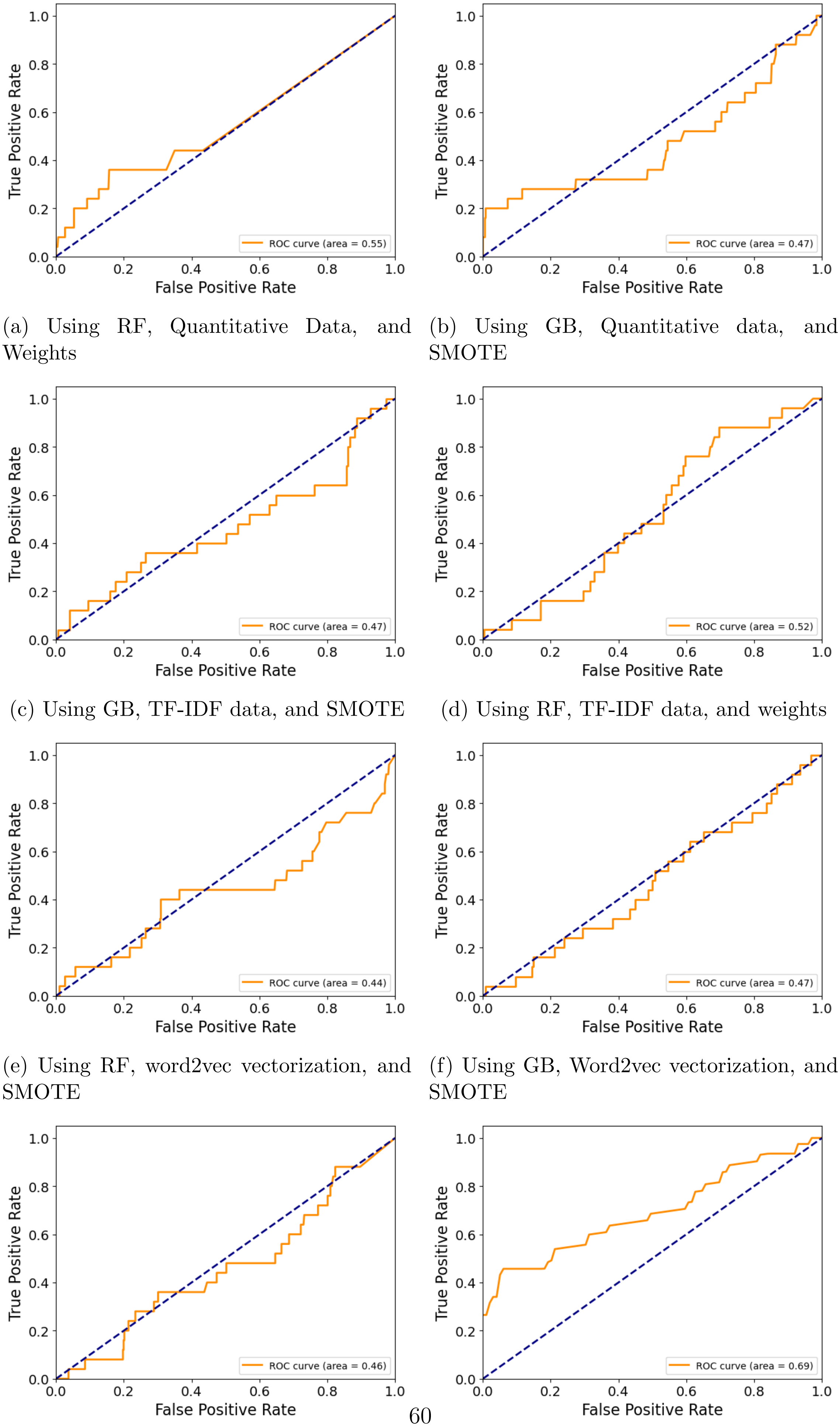
In Table 17 the confusion matrix for the "Good Example" algorithm is presented across different ML models, and using different preprocessing techniques. The results indicate that the LSTM model, which processes text data, outperforms other techniques. The table shows that TF-IDF vectorization gives the least effective results. When excluding LSTM from the comparison, models using quantitative data show a better performance over using vectorization techniques and text data.

ML	Data	Balance	TP	TN	FP	FN
RF	Quant	weights	2	479	13	23
GB	Quant	SMOTE	5	461	31	20
GB	TF-IDF	SMOTE	0	490	2	25
RF	TF-IDF	weights	0	491	1	25
RF	Word2vec	weights	0	489	3	25
RF	Word2vec	SMOTE	2	473	19	23
GB	Word2vec	SMOTE	1	461	31	24
LSTM	Keras	weights	6	474	18	19

Table 17: Confusion Matrix Results for “Good Example” Prediction

The performance of the “Good Example” classification can be visualised through ROC curves, as shown in Figure 27. The ROC curve shows the TPR against the FPR at different thresholds and is described in detail in sub-chapter 2.4.3.

When GB and RF are used the models achieve an AUC-ROC curve of around 0.50, suggesting that these model’s performance is not much better than random chance. The only model that clearly outperforms random guessing is LSTM which demonstrates a better performance with a AUC of 0.69, shown in subfigure 27h. These results align with the idea of DL models, being better at capturing complex patterns in textual data compared to ensemble ML algorithms.



(a) Using RF, Quantitative Data, and Weights

(b) Using GB, Quantitative data, and SMOTE

(c) Using GB, TF-IDF data, and SMOTE

(d) Using RF, TF-IDF data, and weights

(e) Using RF, word2vec vectorization, and SMOTE

(f) Using GB, Word2vec vectorization, and SMOTE

(g) Using RF, word2vec data, and weights

(h) Using LSTM with weights

Figure 27: Combined ROC Curves

4.2 Algorithm II: Hazard Classification

In this section, the results of classifying hazards using activity descriptions are presented. In Table 18, the results of trying various methods can be seen. The "simple" model refers to a basic LSTM model, which is also the architecture used for every model except the complex model. The hyperparameters of both models can be found in Appendix F. The "Thresh" model refers to a model where the parameters used to calculate the optimal thresholds are altered, as discussed in subsection 3.6.2. Stemming refers to a model employing stemming, "NaN for others" refers to a model which filled work operations categorized as "others", as NaN, while the complex model denotes a more sophisticated LSTM architecture. Macro and Micro are denoted as respectively Mac. and Mic. in the table.

Using the F1-scores from Table 18 as the main performance metrics, suggests that altering the function for setting thresholds does not increase the overall performance of the model. Stemming the text data had a negligible effect, slightly improving the results. Filling work operations labeled as "others" with "NaN", did not have a positive effect on the results. A noticeable improvement is seen when using the complex LSTM architecture. Given the substantial increase in performance the "Complex model" is used as the primary model explored for further discussion in this subsection.

Model	Mac. acc.	Mac. prec.	Mac. recall	Mac. F1	Mic. prec.	Mic. recall	Mic. F1
Simple model	0.6846	0.2338	0.6022	0.3085	0.2843	0.7537	0.4128
Thresh(0.3, 0.2)	0.7382	0.2480	0.4267	0.2679	0.2758	0.4795	0.3502
Thresh(0.5, 0.1)	0.6852	0.2309	0.5967	0.3068	0.2789	0.7187	0.4018
Stemming	0.6865	0.2382	0.5918	0.3090	0.2871	0.7623	0.4171
NaN for others	0.6785	0.2340	0.6049	0.3079	0.2803	0.7561	0.4090
Complex model	0.8127	0.4097	0.4684	0.3926	0.4145	0.6628	0.5100

Table 18: Performance Metrics from Different Models Classifying Hazards

In Table 19 the performance of each hazard type can be seen, with the original Norwegian names of the hazard being available in Appendix C. Analysis of the table shows that hazards that occur less frequently in the dataset tend to have a higher accuracy, and lower F1-scores compared to more frequent hazards. The reason for this is because the algorithm often guesses "false" for rare hazards, which inflates the accuracy, while the model has few positive examples making it harder to find true cases, which is reflected in the F1-scores. Generally, the model shows good performance across most hazards, indicating that the model is significantly more effective than random guessing.

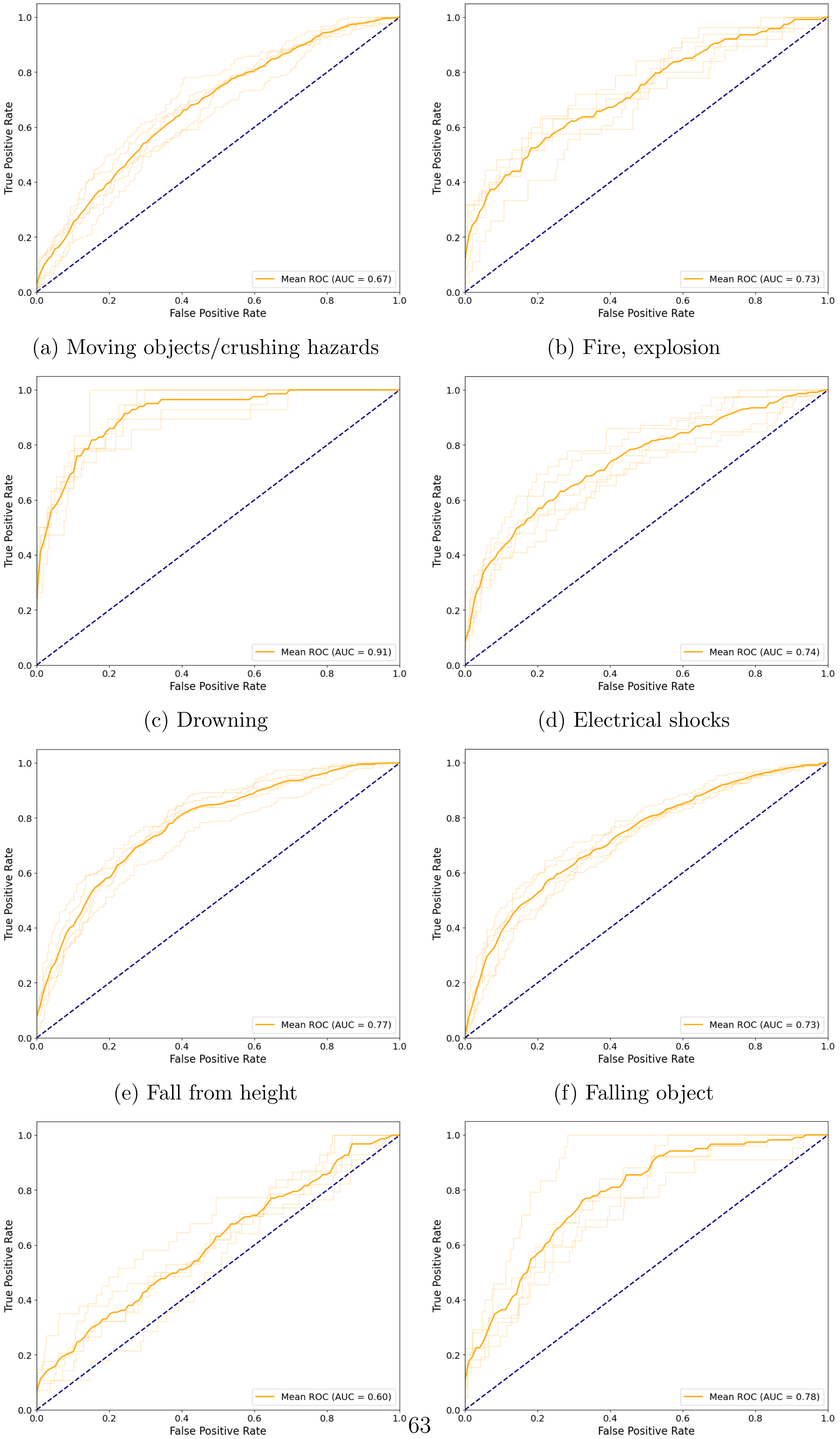
Figures 28 and 29 show the ROC curves for the classification of various hazards. The model has its best performance with the "drowning" hazard, achieving anAUC of 0.91. The model also demonstrates a strong performance when classifying less frequent hazards, such as "dust, smoke, gases, and toxic substances" (0.78 AUC), and "high pressure, splash hazards" (0.78 AUC).

Hazard	Accuracy	Precision	Recall	F1-score
Moving hazards/ entrapment risks	0.5415	0.4471	0.8769	0.5899
Fire, explosion	0.9233	0.5356	0.3071	0.3603
Drowning	0.9678	0.6864	0.4962	0.5622
Electrical shocks	0.8775	0.4127	0.3852	0.3938
Fall from a height	0.6909	0.5794	0.8024	0.6716
Falling object	0.6206	0.5827	0.9010	0.7073
Risk of tripping or slipping	0.8760	0.5029	0.2459	0.2396
High pressure, splash hazard	0.9097	0.4501	0.3249	0.2922
Structural failure	0.7700	0.2954	0.4026	0.2787
Collision	0.6733	0.3384	0.5902	0.4183
Sharp objects (cuts, stabs)	0.9009	0.3144	0.3075	0.2445
Dust, smoke, gases, toxic substances	0.9492	0.5762	0.2834	0.3530
Heavy lifting/heavy materials	0.8199	0.2221	0.3413	0.2519
Emissions/pollution	0.8944	0.3077	0.2963	0.2514
Weather conditions (wind, cold, fog)	0.8340	0.1994	0.4002	0.2622

Table 19: Performance Metrics for Predicting Every Hazard Using Complex Model

On the other hand, the model struggles more with certain types of hazards, such as “trip and slip” hazards, and “heavy lifts”. This indicates that the models accuracy is not tied to the number of occurrences, but that it can be effective at identifying hazards both a high and low number of occurrences.

Overall, the ROC curves suggest that the model is good at hazard classification. The ability to accurately classify hazards is promising, and it shows potential for a ML tool to be useful when doing JSAs.



(a) Moving objects/crushing hazards

(b) Fire, explosion

(c) Drowning

(d) Electrical shocks

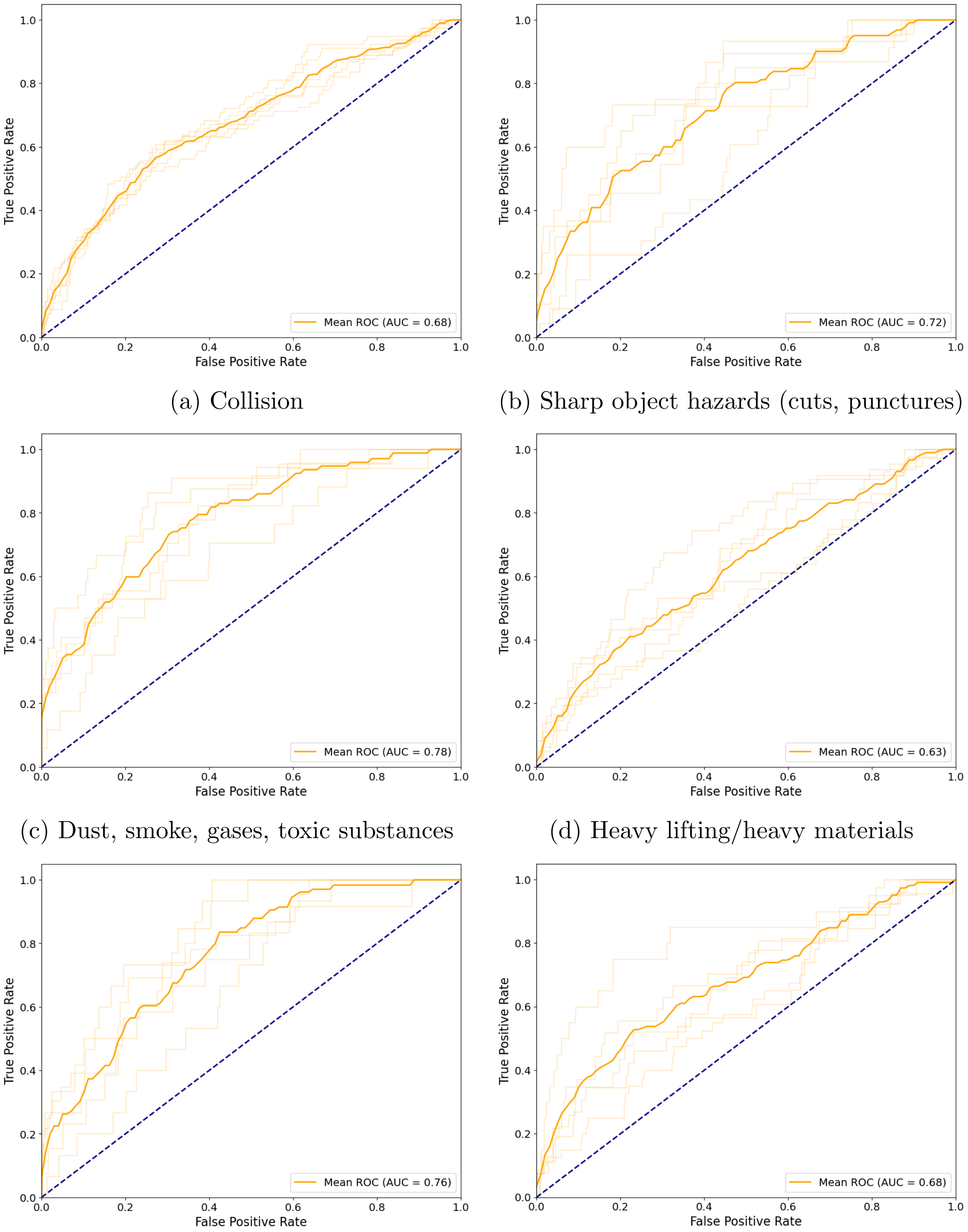
(e) Fall from height

(f) Falling object

(g) Risk of tripping or slipping

(h) High-pressure hazards/ splash risks

Figure 28: ROC Curves for Hazard Identification - Set A



(a) Collision

(b) Sharp object hazards (cuts, punctures)

(c) Dust, smoke, gases, toxic substances

(d) Heavy lifting/heavy materials

(e) Emissions/pollution

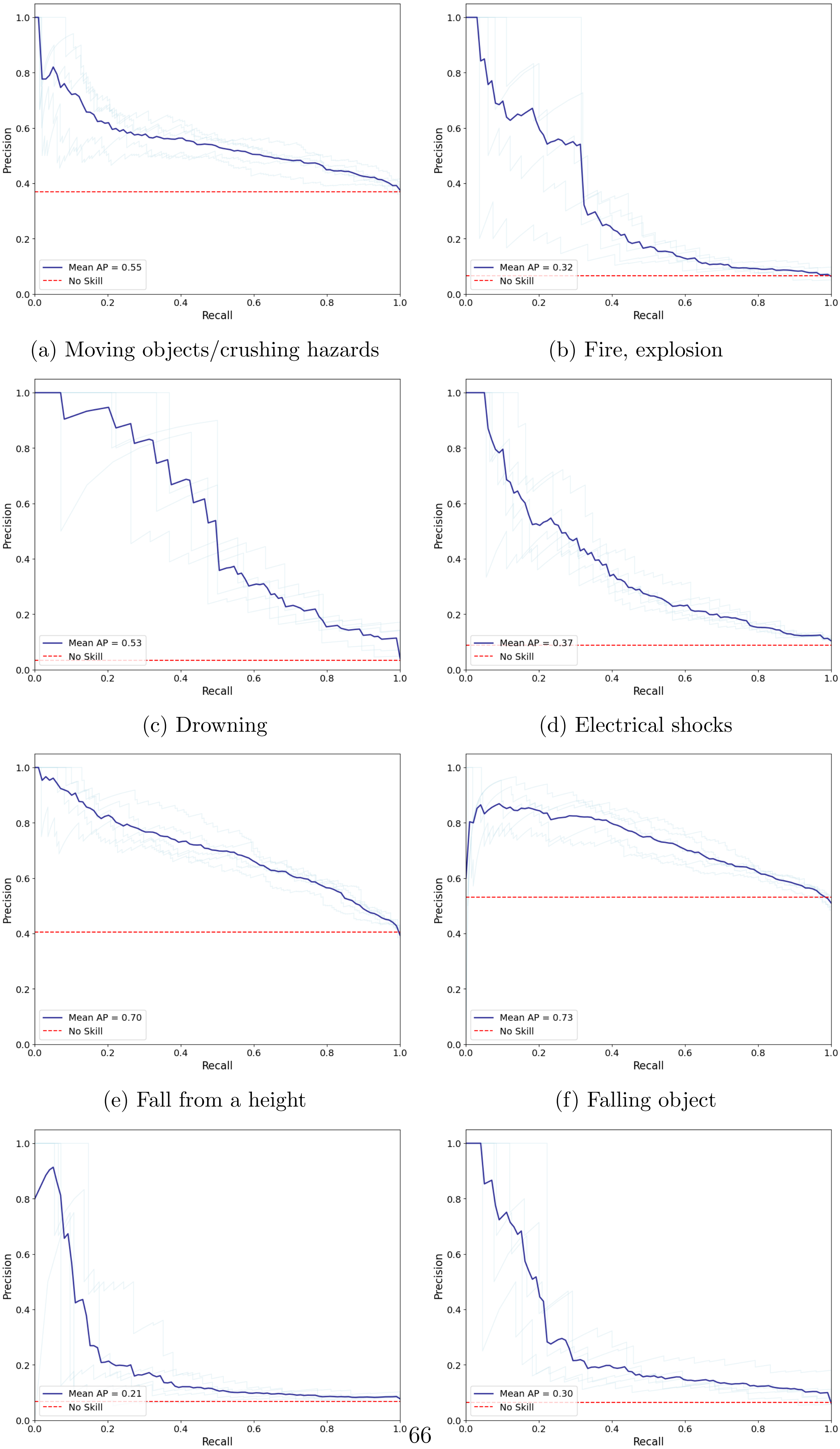
(f) Weather conditions (wind, cold, fog)

Figure 29: ROC Curves for Hazard Identification - Set B

Figures 30 and 31 show the PR curves for every hazard classified, looking at the trade-off between precision and recall for different thresholds. The PR curve is described in more detail in sub-chapter 2.4.3. The PR curve is often looked at as particularly important for imbalanced datasets, which makes it especially interesting here.

In both figures, the Average Precision (AP) is the metric denoted, showing the model's average precision across all levels of recall. The "no skill" line shows the performance against random guessing. Hazards with more frequent occurrences have a higher laying "no skill" line and therefore tend to have an AP score that is higher than hazards with fewer instances. The "Falling object" hazard achieves the highest AP score with 0.73, indicating that the model has a strong ability to predict this hazard.

For the most part, it seems that hazards that have a high performing ROC curve tend to have a high performing PR curve. Overall the results of the PR curves seem promising, and the model seems accurate in predicting hazards.



(a) Moving objects/crushing hazards

(b) Fire, explosion

(c) Drowning

(d) Electrical shocks

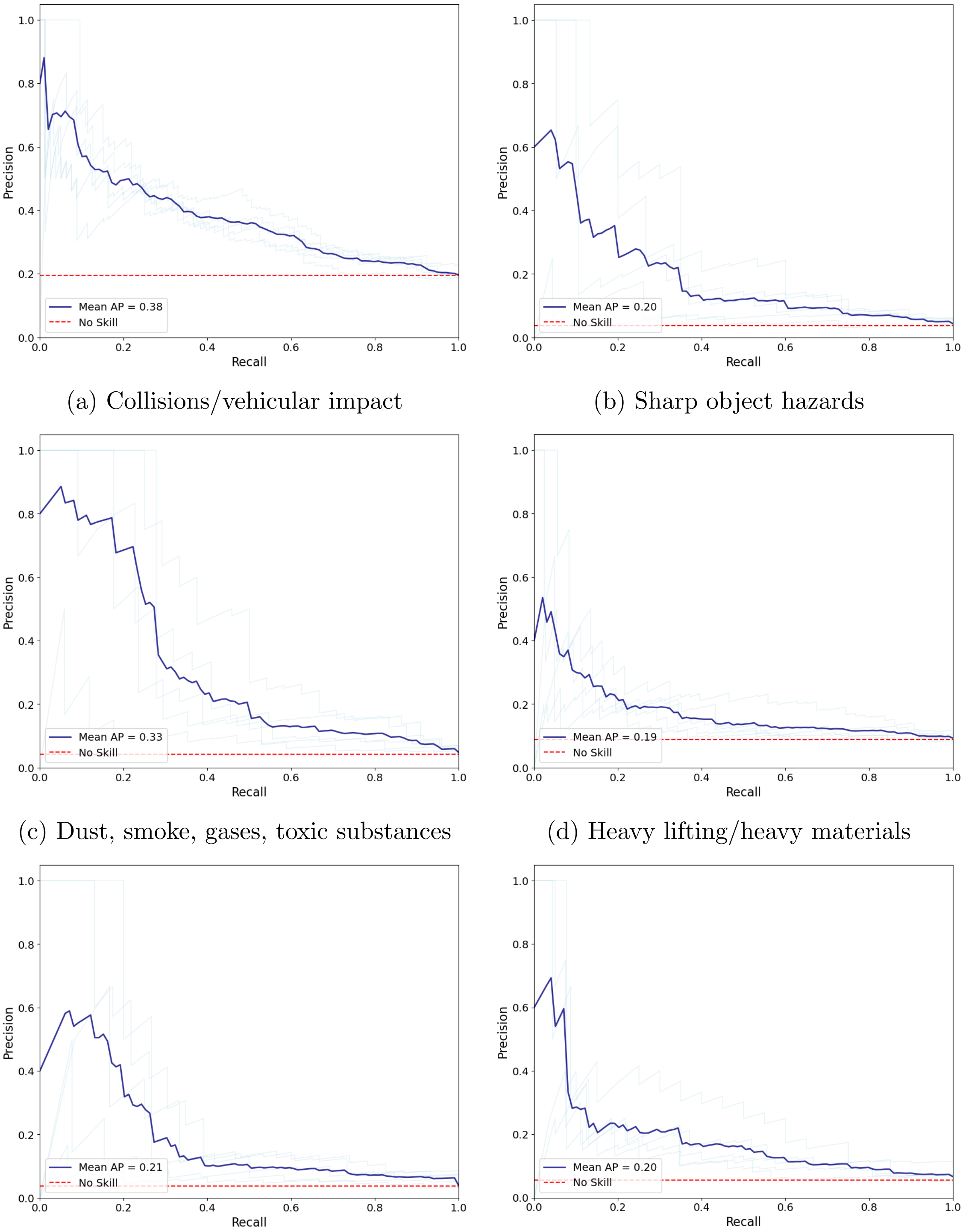
(e) Fall from a height

(f) Falling object

(g) Risk of tripping or slipping

(h) High-pressure hazards

Figure 30: Precision-Recall Curves for Hazard Identification - Set A



(a) Collisions/vehicular impact

(b) Sharp object hazards

(c) Dust, smoke, gases, toxic substances

(d) Heavy lifting/heavy materials

(e) Emissions/pollution

(f) Weather conditions (wind, cold, fog)

Figure 31: Precision-Recall Curves for Hazard Identification - Set B

Table 20 shows the instances where the hazard classification algorithm displays high confidence in its predictions, but the classification ultimately is incorrect. The text in Table 20 is translated, and the table with the original Norwegian text can be found in Appendix H. These instances can provide valuable information about which mistakes are made by the ML algorithm, and give a deeper insight into the data. The table shows the hazard classified by the algorithm with its certainty, in addition to the hazards given by human evaluators. It is observed that the model often has a higher certainty for the labels where there are many occurrences and has a lower certainty for hazards with fewer occurrences.

The results from Table 20 is interesting. The author has more expertise in the field of computer science compared to construction safety, so caution should be advised when interpreting the results. Some of the hazards identified by the model, despite being classified as incorrect, could represent actual risks. If that is correct, the table seems to suggest that the model has avoided overfitting and has an ability to generalize and make predictions independently of examples it was trained on.

Activity description	Label By algorithm	Certainty	Actual label
Assembly of 4 sandwich walls in axis B1, the elements should be mounted on a cast-in-place wall 6.8 meters above ground level. Planned assembly 02.03.22.	Moving Objects/Pinch Hazard	0.8485	Falling object
We will be blasting rock towards the building (approx 8 m high) where the rock is about 1-2 m from the building, where we will blast with a 4 m bench height. The most critical place is secured with drill rods.	Fire, Explosion	0.5115	Moving Objects/Pinch Hazard, Fall from Height, Falling Object, High Pressure, Splash Hazard
After blasting, cracks have formed in the wall against existing tanks, this will be removed on behalf of the client.	Fire, Explosion	0.3962	Fall from Height, Moving Objects/Pinch Hazard
[Anonymized] has been tasked with filling a canal inside [Anonymized], where we will lay 1200btg at -0.70 and install 3 manholes.	Drowning	0.8204	Moving Objects/Pinch Hazard, Dust, Smoke, Gases, Toxic Substances, Emissions/Pollution
Blasting of trench/road under and along high-voltage power lines. [Anonymized] owns the high-voltage line that carries 420kV. The trench is about 70 meters. Two layers and 18 mats will be laid to prevent stone scattering. The blasts will be adjusted according to the conditions with the water dams. There will be a number of small blasts.	Electric Shock	0.7540	Falling Object, High Pressure, Splash Hazard, Moving Objects/Pinch Hazard
The work will partly take place on scaffolding and atop a glass roof. Glass is to be adjusted in carrying profiles and secured with clamping strips/profiles. In addition, a profile is to be mounted at the front of the glass roof from the scaffolding. The glass has already been hoisted into place and lies down in the carrying profiles. Plates are to be laid on top of the carrying profiles to walk on when clamping strips are to be mounted.	Fall from Height	0.9257	Trip or Slip Hazard, Falling Object
Installation of a ventilation unit on the roof of building A, the unit is lifted onto the foundation built on the roof. Each section of the unit is assembled together and then side-shifted into place by [Anonymized].	Falling Object	0.9897	Moving Objects/Pinch Hazard
Excavators to work near gas pipeline. Everyone must be aware of the procedure in case of a gas leak.	High Pressure, Splash Hazard	0.4585	Fire, Explosion
Installation of steel beams for supporting modules	Structural Failure	0.7466	Falling Object, Collision/Impact, Weather Conditions (wind, cold, fog)
Onboarding/offloading and tipping from flat barge to sea	Collision/Impact	0.7720	Drowning
Cutting down bushes and trees and running through a compost shredder	Sharp Object (cut, puncture)	0.7862	Falling Object, Moving Objects/Pinch Hazard
Safe work with spiking and other work in the base tunnel	Dust, Smoke, Gases, Toxic Substances	0.8353	Fire, Explosion
Walls must be turned with tower crane and mobile crane	Heavy Lifting/Heavy Materials	0.8664	Structural Failure, Moving Objects/Pinch Hazard
Plastering in sea, in front of quay	Emissions/Pollution	0.6845	Drowning, Moving Objects/Pinch Hazard
Installation and hoisting of steel beams	Weather Conditions (wind, cold, fog)	0.7712	Dust, Smoke, Gases, Toxic Substances

Table 20: The Most Confident Misclassifications by the Model

Figure 32 presents a series of confusion matrices for every hazard in the dataset. Each matrix is divided into four categories: TP, TN, FP, and FN, showing the performance of the model comprehensively.

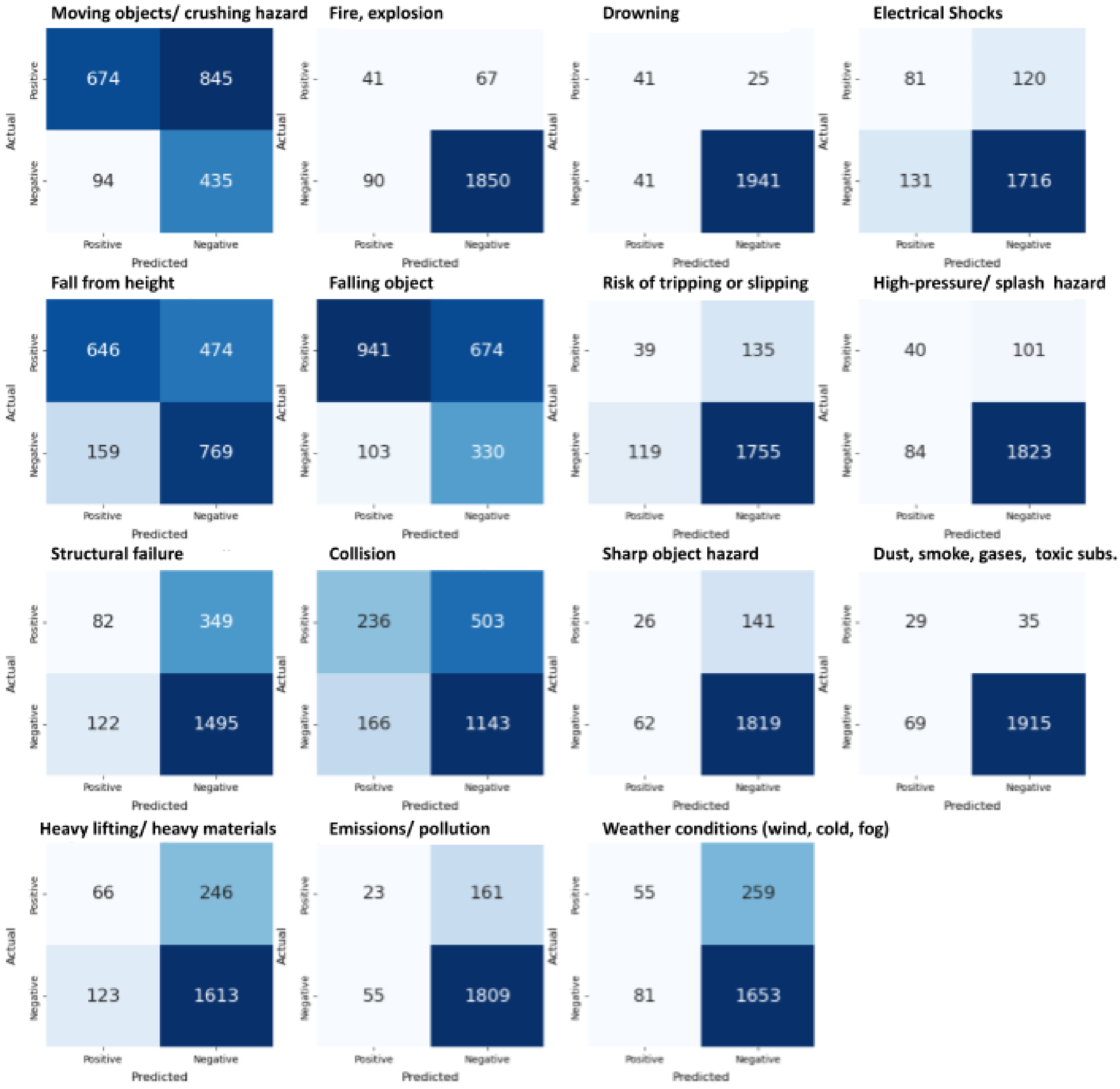


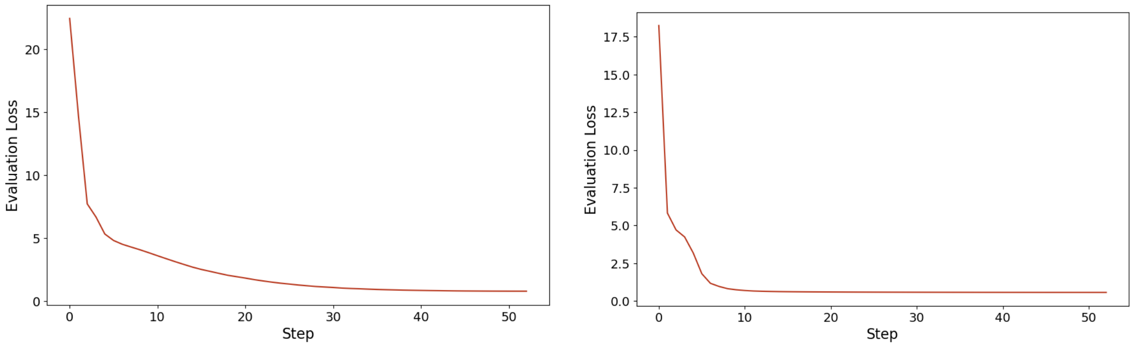
Figure 32: Collection of Confusion Matrices for Hazard Identification

This figure clearly illustrates a heavy class imbalance, with most of the actual instances being "negative". "Moving objects", "Falling", and "Falling objects" is the three hazards that occur most often. The model tends to have many FNs for classes with frequently occurring hazards. For less frequent hazards, the distribution of Type 1 and Type 2 errors is more balanced. The hazards have a different balance of classes, which creates the challenge of setting a suitable threshold for different hazards.

4.3 Algorithm III: Preventative Measures Generation

In this subsection, the results of the generative text algorithm made to propose preventive measures are shown. The text-to-text model was trained over 52 iterations for both versions trained. Figure 33 depicts the model's evaluation loss at each training iteration for both the small model and the base model. Initially, there is a fast decline in evaluation loss, before the rate of improvement flattens out. This suggests that the model learns quickly at the start and that the learning falls over each iteration. Learning slowing down is common in ML and could show that the model is approaching its learning capacity given the current data and model architecture.

The models have a different training behavior. It seems like the base model reaches a plateau in its learning curve quicker than the smaller model. This suggests that the base model is able to learn and converge quicker to a solution. Each step in training of the base model takes more time, this means that the number of steps is not an indicator of the actual time spent in training. The plot suggests that the base model could be trained for fewer steps possibly achieving similar performance.



(a) T5 Small Model

(b) T5 Base Model

Figure 33: Validation Loss per Training Step for T5 Models

Table 21 displays eight randomly selected samples from the test set, showing the model's generated preventive measures using the base model. These samples are not given to the model during its training phase and are unseen to the model. Table 21 reveals that the AI model is able to generate measures that make somewhat sense to the activity and hazard. The model can identify items such as a “fall protection harness” and “life jacket” which is consistent with the actual preventive measures.

It can be observed that the model often repeats the activity description instead of proposing a concrete and actionable measure. For instance, using the hazard “work in closed spaces/ lack of oxygen”, the models response is “Work in closed spaces/ lack of oxygen must be stopped”, only repeating the problem without any practical solution or preventive action. In addition, the model's generated measures often lack detail and are short, vague, and generic. The human-written measures are comprehensive and descriptive, compared to the AI-generated measures.

Activity	Hazard	Actual measure	AI predicted Measure
At the bottom of the atrium several parallel works can take place which can affect each other and create dangers Work in area 1 under level 3 and work in the atrium opening area 2 Subjects who work in close proximity to each other and pose a risk must carry out SJA adapt to their tasks and have control over the risks	Fall from a height	When using a laundry basket a fall harness must be used in the hook point on the basket Workers must have received training in washing baskets before use	Use a fall protection harness
Installation of piles near the railway approx 27m	Electric shocks	Lifted with a truck crane do not lift against the railway the pile is no longer than 8 metres the lift will not be higher than 9m	Ensure that there are no electrical shocks in the area
Remaining work in [Anonymized] installs two sensors [Anonymized] attaches the valve and removes blockage in the sump	Work in closed spaces/lack of oxygen	Always two people when working in the sump	Work in closed spaces/lack of oxygen must be stopped
Sheet piles must be frost insulated using winter mats 3 6 and 9m Played on the top back side of sheet piles and at the bottom of sheet piles foot girders Certain areas have been blasted out cutting 2 m from footings	Fall from a height	start by only attaching winter mats to the back side of the batten and ease over to the front side when attaching battens down to footings cables are attached along the pile as a fall block that can be secured in so you can move along the pile	Use a fall protection harness
Dismantling of scaffold construction in the atrium area with many simultaneous activities Transport of scaffolding parts to unloading zone 1 for removal Applies to works for subjects with tasks that have not been possible to carry out get to before	Falling object	Secure tools and materials from falling down Placement	Ensure that the scaffolding is secured in the atrium area
Sawing trees at boreholes 2	Sharp object (cut, stab)	Saw trousers helmet with visor and gloves	Use of a sledgehammer
The heaviest windows weigh 280 kg and must be lifted into place with a tower crane The windows are raised with straps around the frame of the window and the straps are secured so that it does not slip off the window with possibly straps and screws The window is lifted down to the correct position between the stile and the facade The vidue is secured by lowering When the window comes down to the correct position this is placed on blocks and the window is set in place Bricks are mounted on the inside against which the window stops at the top Straps are removed and the window is attached secured with blocks	Moving objects/trap hazard	Block off the area	The window is lifted into place with straps around the frame of the window
Filling masses into the sea for a new quay	Drowning	Lifejackets must be worn by drivers of machines and trucks Lifebuoy on land Marker on machines and cars Area lighting	Use of a life jacket

Table 21: Eight Randomly Selected AI-Generated Preventive Measures Using T5 Base Model

5 Discussion

This section provides a comprehensive discussion of the conducted research. It delves into the results of the different ML models, and the challenges encountered during the development of these algorithms, in addition to their practical implications in real-world scenarios.

5.1 Discussion of JSA Quality Classification Algorithm

This research developed a ML classification model designed to assess if a JSA can be considered a “Good Example”. The motivation for creating the model is to explore the potential of using ML as a tool to assess the quality of JSAs, potentially functioning as a leading safety indicator in construction projects. The key findings are that LSTM achieved the best results, indicating that it is possible to create a ML model rating the quality of a JSA if trained correctly with data of high quality.

5.1.1 Algorithm Performance

Results presented in Table 17 show that the LSTM model outperforms the other models. This superiority led to the use of LSTM and DL techniques in the subsequent development of the following algorithm, since all algorithms use text data as input in this research. The LSTM model, further explained in subsection 3.6.1, shows that it is able to classify significantly better than random guessing, even though its effectiveness remains moderate.

The performance of the traditional ML algorithms, RF and GB, performs about as good as random guessing, seen in Figure 27. This outcome reveals the limitation of these traditional ML methods, particularly in handling sequential data. Unlike LSTM, RF and GB have not been developed specifically for processing sequential data, which makes them worse at interpreting textual semantic meaning. As seen in Table 16, their performance improves when using quantitative data, indicating a correlation between the number and the average length of identified preventive measures and the quality of the JSA. This observation suggests that traditional ML algorithms struggle with capturing the semantic meaning of textual data.

As described in section 3.4.6, there seems to be a relationship between the number of identified measures, and the measures lengths, with the quality of the JSA. It seems logical that the quality of a JSA increases with each preventive measure identified. In addition, it seems possible that the length of a measure may be an indicator of how detailed the measure is, potentially improving the overall JSA quality.

Analysing Figure 27 reveals that the overall performance of the algorithms is modest. The LSTM model outperforms random guessing slightly, and the result of the model’s other algorithms is just as good as random guessing. This observation raises questions regarding the suitability of ML as a tool to assess the quality of JSAs. The underlying reasons for the limited performance will be further explored in the following subsection.

5.1.2 Limitations and Future Research

The moderate performance of the models in assessing JSA quality can largely be explained by the quality of the data used. A key challenge is the scarcity of samples labeled as “Good Examples”. In total, only 1.3% of all samples, 7.3%, when excluding NaN labels, are marked as “Good Examples”, as detailed in subsection 3.4.6. This scarcity makes the model tasked with detecting anomalies, and the model is subsequently not trained on a rich set of positive examples, which clearly has a negative effect on the performance of the ML algorithm.

Additionally, the integrity of the samples labeled as “Good Examples” is questionable. For instance, a JSA labeled as a good example had one preventative measure, which was ”Use common sense”. This raises concerns whether some of the samples labeled as good examples are accurately labeled. Despite most “Good Example” labels seeming to be appropriate, the presence of mislabeled examples will confuse and hinder learning for the ML model.

Constructing a better dataset both in terms of size and quality is essential to improving ML model performance. Expanding the dataset poses new challenges, there is a need for expert evaluation to accurately label JSA by its quality. The capabilities of any ML model are strongly tied to the quality of its training data. Therefore using a safety expert in labeling JSA documents is a crucial step for ensuring better performance of ML algorithms.

Previous research has mostly focused on creating ML model that uses leading safety indicators to predict lagging safety indicators, as seen in 2.2. However, this thesis tries to predict the quality of JSA, a leading safety indicator itself. A premise for this algorithm to be effective is a significant correlation between the quality of JSA and lagging indicators such as the frequency of accidents during construction activities. An aspect of future research is to investigate the relationship between the quality of JSA, as assessed by safety experts, and the actual risk of injuries and accidents associated with the respective activity.

A potential improvement for future models involves transitioning from a classification approach to a regression approach. When thinking of JSA quality it seems logical to represent the quality as a continuous scale, and not as one out of two states. Therefore framing the problem as a regression problem, and rating the JSA quality on a scale from 1 to 5, may provide a more nuanced understanding of the actual quality of the JSA. A regression-based approach would take into consideration the varying degree of quality in JSA, enabling models to capture distinctions more easily than a binary classification system.

5.1.3 Practical Implications

The development of a ML tool to assess the quality of JSA could be a valuable instrument in construction safety. Using injury reports opens the possibility to examine the correlation between JSA quality and occurrence of accidents for those activities. The usefulness of a model would depend on a correlation between high JSA quality and reduced accident rates. By establishing this connection, the model would be able to bridge the gap between a leading indicator, JSA quality, and lagging indicators such as injury occurrences. The model would then be able to enhance safety performance by identifying JSAs of low quality.

Implementing an ML-based system to evaluate the quality of JSAs could offer valuable feedback to the authors of the JSA. For instance, a potential system could require a JSA to achieve a score over a fixed threshold to be accepted. However, the "black-box" nature of many ML algorithms could be a challenge, since the algorithm offers no feedback to the user other than a classification or a score. This limitation makes it challenging for the user to improve the quality of the JSA since the user does not have concrete points for improvement. To address this issue, future systems could try to integrate generative AI models that can give textual feedback on ways to improve the JSA quality.

Additionally, integrating a system that assesses the quality of JSA could enable nuanced analysis. With additional data connected to the JSA, it could reveal patterns such as some departments producing lower quality JSAs compared to other departments. Such insight can be useful in identifying areas of improvement. Overall, the utilization of a ML tool to assess the JSA quality could have a positive effect on safety management, providing well-informed workers and hopefully creating a culture of continuous improvement in safety standards.

5.2 Discussion of Hazard Classification in JSA

This subsection discusses the performance, limitations, and practical implications of the ML algorithm developed for identifying potential hazards. The model shows effectiveness in classifying various hazards, with a ROC-AUC spanning from 0.60 to 0.91 depending on the hazard. This suggests that ML has the possibility to enhance workplace safety by helping to identify hazards.

5.2.1 Evaluation of Classification Performance

The performance of the hazard classification algorithm, using activity description and work operation, appears promising. The complex LSTM model has an accuracy of 0.8127, a macro and micro F1 of 0.3926 and 0.5100, respectively, as seen in Table 18. The observed high accuracy stems from the class imbalance present in the dataset, where most activities are labeled as "False" for a given hazard. Consequently, this imbalance highlights the importance of the F1-score as a particularly relevant performance metric, as discussed in section 2.4.3. Another observation is the model's tendency towards a higher recall in comparison to precision, which indicates that the model is more effective at identifying positive samples, while at the same time generating some FPs.

The ML model's performance shows variability when classifying different hazards. As depicted in Figures 28 and 29, the ROC-AUC spans from 0.60 for slip hazards to 0.91 for drowning hazards. This variation in performance could be linked to the specificity of the hazard. For example, in predicting the hazard "drowning", the model can search for words such as "sea" and "water", which it might find closely correlated with the hazard. In contrast, "slip hazard" hazard might be harder to classify due to fewer distinct words being closely associated with the hazard. This suggests that ML algorithm is more suitable to classify hazards with clear and distinct indicators while being less effective when dealing with more obscure and context-dependent hazards.

In comparison to other studies like Chi et al., 2014, which also categorize hazards, the algorithm designed in this thesis demonstrates superior performance. Chi et al., 2014 uses mainly precision and recall as performance measures, but by combining these metrics the F1-score can be calculated as seen in section 2.4.3. For instance, their F1-score for classifying falls from height is 0.039. In contrast, the algorithm developed in this thesis achieves a considerably higher F1-score of 0.6716 for the same hazard, as detailed in Table 19. When looking at the classification of the "Heavy Equipment", a hazard where Chi et al., 2014 achieved better results with an F1-score of 0.245, compared to this thesis 0.2519. For this hazard, it seems like the performance of both models is more comparable. The performance classifying different hazards varies more in Chi et al., 2014, compared to this thesis, but it seems like the models developed in this research tend to outperform Chi et al., 2014. This observation can likely be attributed to the significant advancements in ML and NLP techniques over the past decade. The rapid development of these technologies has enhanced the capabilities of ML methods, and it seems like incorporating ML tools in the JSA creation process is more beneficial now than previously.

Table 20 presents an insight into the model's mislabelings, offering a deeper understanding of the model's limitations. Analysing these mislabelings there seem to be different causes for them. In some cases, it seems like multiple hazards could have been applicable to an activity. For instance, in the first sample where "moving object" is identified as a hazard by the human author, the ML model classifies it as a "falling object" hazard. These hazards overlap, and any "falling object" can be considered a "moving object", and these hazards are strongly correlated as seen in Figure 20. Because of this, the ML algorithm and human authors classify differently, and it could be considered to combine the two hazards into one hazard. Furthermore, the ML model seems to struggle to understand the full context of the construction activity in some cases. For instance, in the third row of Table 20, a problem has occurred as the result of previous blasting. The ML algorithms recognize the word "blasting" and automatically want to classify it as a "Fire, Blasting" hazard. In this case, the ML algorithm does not fully understand the context of the activity description and therefore, wrongly identifies the hazard. The model not fully understanding the context of the task at hand is a recurring issue in several of the observations.

Additionally, Table 20 highlights potential challenges encountered by JSA authors in accurately identifying hazards. Looking at multiple samples in Table 20 reveals that certain hazards might have been overlooked by JSA authors. Direct comparison between the ML model's performance and human accuracy is difficult due to human errors possibly being present in the test set. The model seems to possibly identify hazards overlooked by human evaluators. If the observation is correct, the possibility of utilizing a hybrid system to identify hazards opens up. The system could integrate human expertise with a ML algorithm, which could potentially outperform either method used in isolation, enhancing hazard identification within the JSA process.

5.2.2 Limitations

One challenge encountered in the development of the ML model for hazard classification was finding a threshold suitable across different hazards. This problem stemmed from the fact that the model generally produced higher certainty levels for often occurring hazards while producing lower certainty for less common hazards. Consequently, the model tended to overpredict frequent hazards and rarely identify rare hazards. To address this issue, a function was developed, setting a unique threshold for each hazard, and optimizing the F1-score for each hazard separately. This resulted in less frequent hazards requiring a lower threshold to be classified compared to more common hazards. This threshold setting highlights the trade-off between precision and recall, which is discussed in 2.4.3.

Performance metrics such as accuracy, recall, and confusion matrices consider the Precision-Recall trade-off. On the other hand, plots like the ROC curve and PR curves offer a comprehensive view of how the model will perform across various thresholds. These curves are helpful to understand the behavior of the model and can be used to find an appropriate balance between Precision and Recall. Balancing both is important to identify both rare and frequent hazards while balancing the number of FPs and FNs in an appropriate manner. In this research, the balance was tackled by optimizing the F1-score, which seems like an appropriate strategy as it ensures equal emphasis on both precision and recall.

The selection of LSTM as the model employed was made due to its historical recognition as a SOTA model for processing sequential data. However, due to the rapid developments in DL there has been introduced new SOTA models utilizing the attention mechanism, showing promising results in classification tasks using textual data. These developments indicate that a Transformer-based model potentially could outperform the LSTM model used in this thesis. Some newer LSTM models even incorporate the attention mechanism enabling focused processing of text segments. Given the fast-moving technological frontier within DL, it is possible that newer models could offer an enhanced performance.

Further, there is potential for more intensive optimization of hyperparameters and to utilize larger and more complex LSTM networks. As explained in section 3.6.2, only a training and test set was used, due to the limited number of positive samples. This constraint increases the risk of overfitting, leading to no extensive hyperparameter optimization. The decision was made to prioritize realistic results over inflated results. Regardless, this opens up the possibility of optimizing hyperparameters more heavily and possibly creating deeper networks for hazard classification, which potentially could improve the performance.

5.2.3 Practical Implications

The integration of a ML system to enhance hazard identification during a JSA is a promising concept. The model developed in this research seems to have the potential to detect some hazards that humans overlook. In a potential implementation, the JSA team would initially describe the activity and identify hazards themselves. Following this, the activity description could be sent to a ML system, which evaluates the probability of potential hazards. If the system identifies a hazard over a certain threshold, which the JSA team has not recognized, the hazard could be suggested for further evaluation by the JSA team, which then could decide if the proposed hazard is correctly identified.

This approach could potentially enhance hazard identification since the ML model might identify hazards overlooked by humans. Assuming this system improves hazard identification, a new ML model could be trained using data that ML helped label, enabling continuous learning and improvement of the system. Implementing a system like this could modernize the JSA process, helping the process become more data-driven. A potential system should be a tool for safety experts, helping them identify hazards, instead of replacing them.

5.3 Discussion of Preventative Measures Generation

This subsection delves into the performance, limitations, and potential of the generative AI model used to propose preventive measures. The key finding is that generative AI has the potential to help propose measures to activities in construction activities, but that the quality of the measures generated currently falls short compared to measures created by human experts.

5.3.1 Model Performance

The comparison between the preventive measures generated by a LLM and those written by humans reveals mixed results. The LLM successfully identifies basic measures relevant to a certain hazard, for instance recommending a fall protection harness for height-related risks. However, the model lacks the nuanced and detailed context often provided by human authors, which often makes the responses generic, compared to more specific human measures. As a result the AI model struggles to provide as clear and descriptive actionable steps as human experts to mitigate hazards.

In some cases, the LLM exhibits a “creative” tendency to address hazards. For example in row 6 in Table 21, the model suggests using a sledgehammer instead of a saw to cut a tree to mitigate the risk of a sharp object. While using a sledgehammer would technically mitigate the risk of sharp objects, it would be impractical for solving the task of cutting a tree. As shown in subsection 2.5.4, the model is sequentially generating words with the highest probability, without having any deep understanding of the task at hand.

Figure 33 shows the training process of both models. The base model training, which lasted over 30 hours running on IDUN, seems to reach a plateau relatively early in the learning phase. The observation that both model do not improve in performance after a certain point in their training, indicates the potential of reducing the training duration. In addition, it could mean that a larger model trained for the same time as the base model could yield better results.

5.3.2 Challenges and Limitations

The effectiveness of the generative AI model is linked with the size of the model used. Initially, the "small" model was used resulting in a low quality of the measures being generated. As a consequence, the "base" model was employed, leading to a significant improvement in performance, even though the training time and computational demands increased significantly. Larger models such as the "large", "XL", and "XXL" of T5 could potentially improve the output of the model, but this would require a large amount of training and vast computational resources, possibly taking days or even weeks to train.

Another method for creating better models might be to use patented models such as GPT and PaLM. These models are developed by major tech companies and are at the forefront of AI development. Using these models could potentially generate better and more sophisticated output. These models are not open-source, meaning only the owning company knows exactly how they operate, and this restricts you from being able to train models locally. Running models on their external servers might cost money, and it raises concerns regarding data privacy and GDPR compliance. Personally sensitive data should not be sent to third parties, and therefore the only viable option for this thesis is to train open-source LLMs locally on personal computing resources and NTNU's servers.

A problem with a generative AI tool in generating preventive measures is removing the human element of the JSAs creation. A part of the effectiveness of a JSA is that the authors are supposed to reflect on measures and hazardous activities. Removing the human-driven process with an AI-driven process could potentially decrease the sense of awareness of workers, and remove a sense of responsibility. This could potentially impact the safety awareness of the workers at a construction site negatively.

Furthermore, the use of AI tools in safety management raises ethical and legal concerns. For instance, if an AI-system fails to identify a hazard or the preventive measure is insufficient, and someone is injured, determining the accountable person or entity becomes difficult. The legal implications of someone being injured in an environment using AI-safety tools, must remain largely unexplored since the field is recently developed.

5.3.3 Practical Application and Potential

The model developed in this thesis for generating preventive measures in JSA shows a way generative AI can be used in construction safety. However, the measures generated in this thesis would have limited value in a JSA process. The model often repeats the activity description or suggests a relatively obvious measure likely to be proposed by a human expert. This indicates that a generative AI model in this form, would not give a significant amount of value to the JSA creation process today based on this research's model.

Despite mixed results, the model shows the potential of generative AI enhancing safety in construction projects. As AI technology develops, it is expected that AI models will improve and increase their capability to provide insightful and contextually relevant safety measures. At a certain point in time, AI model will reach a point, where the models has the possibility to be a valuable tool in the JSA creation process.

In the future, a model could be created to produce multiple preventive measures using an activity description and a hazard. The JSA author could then review the AI-generated measures, and select the relevant measures. This approach could help the JSA author identify non-obvious preventive measures.

To summarize, the current generative AI model might not be the sole solution to creating better JSA, but it marks an interesting and promising direction for future AI-assisted safety management. With further advancements and research within the field, AI-assistance has the potential to possibly become a valuable tool for future construction safety workers.

6 Conclusion

The objective of the research was to look into how JSA-data can be utilized with the use of AI to enhance safety performance. For that purpose, three different ML algorithms were developed. In the conclusion each research question is addressed in a paragraph with its belonging algorithm before future work is delved into.

6.1 Findings

The first research question regards the utilization of ML algorithms to measure the quality of JSA. Multiple ML algorithms were used to classify JSA as either exemplary or not. The variable "Good Example" which labels a JSA as exemplary was used for training. The algorithm using LSTM demonstrated a moderate yet promising capability to distinguish between JSA of low and high quality, outperforming traditional ML approaches such as RF and GB. The model's performance was restricted by challenges regarding data scarcity and quality. The insights gained show that ML techniques can be utilized to assess JSA quality, and reveal the importance of high-quality, and well-labeled datasets when training ML algorithms. The practical implications of this algorithm could be a ML-based JSA quality assessment tool that can be deployed in real-time. The research highlights the need for further improvement of the data collection processes to realize the potential of AI. Further research should look into the correlation between JSA quality, as assessed by ML models, and the actual safety outcomes. In addition to this using a regression approach, predicting a floating number instead of a class, could potentially offer a more nuanced understanding of JSA quality.

The second research question explores the application of ML for the identification of potential hazards during construction activities. For this purpose a LSTM model was developed, using multi-label classification of activity descriptions and work operation, demonstrating promising results in detecting a wide variety of potential hazards in construction activities. The performance of the ML algorithm varies with the type of hazard being classified, where the ROC-AUC spans from 0.60 to 0.90 for different hazards. Compared to prior studies, this model shows an enhanced capability for hazard classification. However, it seems like the model struggles with contextually understanding certain activities, and therefore mislabels them. The practical implications of integrating such an ML system into the JSA process could be beneficial, offering a data-driven approach for detecting potential hazards. The results indicate the ML model is capable of detecting hazards that may be overlooked by humans, making it a possible tool used by safety experts. Since the model might mislabel hazards humans would not, the tool should work as a tool for quality assurance of human work rather than acting as a replacement, possibly helping the JSA creation process.

Lastly, the third research question investigates how generative AI can be used to propose preventive measures for identified hazards. The research approaches this problem by fine-tuning a generative AI model using activity descriptions, hazards, and the corresponding preventive measures. The results show that generative AI models are capable of being trained to propose relevant and practical preventive measures. A weakness of the model is that the proposed measures often are generic and lack details compared to measures written by human authors. In addition, the model occasionally proposes unconventional solutions, which could mitigate the hazard, but are impractical in solving the task at hand. Despite these challenges, the research showcases the potential of using generative AI to enhance construction safety. As AI technology advances, it is expected that AI models will be able to propose more insightful, and contextually relevant measures. This progression could make generative AI models a valuable tool for future safety management in the construction industry.

6.2 Further Work

This section suggests potential directions for future work. The following suggestions could help build upon and extend the research in this thesis:

- **Exploring alternative ML models for Classification:** While LSTM was the chosen model for classification tasks in this thesis, the rapid development of other text classification models suggests that another model possibly could offer a superior performance.
- **Exploring Other LLM Models:** T5 was chosen as the LLM employed in this thesis, but exploring larger variants of T5 or other open-source LLMs could possibly yield better results. Furthermore, using non open-source models such as GPT could potentially improve the results.
- **Incorporating ML Tools into the JSA Process:** An interesting aspect of future work is to delve deeper into the integration of AI-driven tools into the JSA workflow. There are possibilities to look into how a ML indicator of JSA quality can be incorporated into the JSA process. In addition to integrating hazard identification and preventive measure proposals into the JSA process.
- **Expanding Datasets Used in the Study:** The datasets, especially the one used to evaluate JSA quality, suffer from its limited size. Gathering a larger dataset could improve the model's performance. The quality of the JSA should be assessed by safety experts. Using a scale ranging from 1 to 5, instead of binary classification, could lead to a more nuanced indicator for measuring JSA.
- **Examining the Relationship between JSA Quality and Lagging Indicators:**

  Investigating the correlation between JSA quality and lagging indicators, such as accident rate, is necessary to determine the effectiveness of a tool assessing JSA quality. Analyzing this relationship would provide valuable insight into the possible effectiveness of the ML tool.

Reference List

Abioye, Sofiat O et al. (2021). 'Artificial intelligence in the construction industry: A review of present status, opportunities and future challenges'. In: *Journal of Building Engineering* 44, p. 103299.

Aggarwal, Charu C. (2018). *Neural networks and deep learning: a textbook*. Spinger.

Albert, Alex et al. (2017). 'Empirical measurement and improvement of hazard recognition skill'. In: *Safety science* 93, pp. 1–8.

Albrechtsen, Eirik (2012). 'Occupational safety management in the offshore windindustry–status and challenges'. In: *Energy Procedia* 24, pp. 313–321.

Albrechtsen, Eirik, Ingvild Solberg and Eva Svensli (2019). 'The application and benefits of job safety analysis'. In: *Safety science* 113, pp. 425–437.

Alkaissy, Maryam et al. (2023). 'Enhancing construction safety: Machine learning-based classification of injury types'. In: *Safety science* 162, p. 106102.

Alpaydin, Ethem (2020). *Introduction to Machine Learning*. 4th. MIT Press.

Alruqi, Wael M and Matthew R Hallowell (2019). 'Critical success factors for construction safety: Review and meta-analysis of safety leading indicators'. In: *Journal of construction engineering and management* 145.3, p. 04019005.

Alzubaidi, Laith et al. (2021). 'Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions'. In: *Journal of big Data* 8, pp. 1–74.

Andreassen, E et al. (2020). *Forutseende sikkerhetsindikatorer-Digitalisering i bygg og anlegg*. Safetec. Available at: https://www.prosjektnorge.no/wp-content/uploads/2020/12/Digitalisering-Forutseende-sikkerhetsindikatorer.pdf. Accessed: 26 Nov 2023.

Arbeidstilsynet (2006). *Arbeidsmiljøloven - aml.* § 3-2. Særskilte forholdsregler for å ivareta sikkerheten.

— (2016). *Forskrift om utførelse av arbeid.* § 10-4. Krav til utstyrsspesifikk opplæring.

Bahdanau, Dzmitry, Kyunghyun Cho and Yoshua Bengio (2014). 'Neural machine translation by jointly learning to align and translate'. In: *arXiv preprint arXiv:1409.0473*.

Baldi, Pierre et al. (2000). 'Assessing the accuracy of prediction algorithms for classification: An overview'. In: *Bioinformatics* 16.5, pp. 412–424.

Bansal, Vijay Kr (2011). 'Application of geographic information systems in construction safety planning'. In: *International Journal of Project Management* 29.1, pp. 66–77.

Bhole, SA (2016). 'Safety problems and injuries on construction site: a review'. In: *International Journal of Engineering and Techniques* 2.4, pp. 24–35.

Bishop, Christopher M. (2006). *Pattern recognition and machine learning*. Springer.

Bobick, Thomas G (2004). 'Falls through roof and floor openings and surfaces, including skylights: 1992–2000'. In: *Journal of Construction Engineering and Management* 130.6, pp. 895–907.

Branco, Paula, Luıs Torgo and Rita P Ribeiro (2016). 'A survey of predictive modeling on imbalanced domains'. In: *ACM computing surveys (CSUR)* 49.2, pp. 1–50.

Carter, Gregory and Simon D Smith (2006). 'Safety hazard identification on construction projects'. In: *Journal of construction engineering and management* 132.2, pp. 197–205.

Chao, EL and JL Henshaw (2002). ‘Job hazard analysis, OSHA Publication 3071 2002 (Revised)’. In: *Occupational Safety and Health Administration, US Department of Labor, Washington* 29, p. 30.

Chatfield, Chris (1986). ‘Exploratory data analysis’. In: *European journal of operational research* 23.1, pp. 5–13.

Chi, Nai-Wen, Ken-Yu Lin and Shang-Hsien Hsieh (2014). ‘Using ontology-based text classification to assist Job Hazard Analysis’. In: *Advanced Engineering Informatics* 28.4, pp. 381–394.

Chicco, Davide (2017). ‘Ten quick tips for machine learning in computational biology’. In: *BioData mining* 10.1, p. 35.

Chowdhary, KR (2020). ‘Natural language processing’. In: *Fundamentals of artificial intelligence*, pp. 603–649.

Davis, Jesse and Mark Goadrich (2006). ‘The relationship between Precision-Recall and ROC curves’. In: *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240.

El Kafrawy, Passent, Amr Mausad and Heba Esmail (2015). ‘Experimental comparison of methods for multi-label classification in different application domains’. In: *International Journal of Computer Applications* 114.19, pp. 1–9.

Fei, Juntao and Cheng Lu (2017). ‘Adaptive sliding mode control of dynamic systems using double loop recurrent neural network structure’. In: *IEEE transactions on neural networks and learning systems* 29.4, pp. 1275–1286.

Flach, Peter and Meelis Kull (2015). ‘Precision-recall-gain curves: PR analysis done right’. In: *Advances in neural information processing systems* 28.

Glorot, Xavier, Antoine Bordes and Yoshua Bengio (2011). ‘Deep sparse rectifier neural networks’. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, pp. 315–323.

Goodfellow, Ian, Yoshua Bengio and Aaron Courville (2016). ‘Deep Learning’. In: *Adapt. Comput. Mach. Learn.*

Graves, Alex (2013). ‘Generating sequences with recurrent neural networks’. In: *arXiv preprint arXiv:1308.0850*.

Haddi, Emma, Xiaohui Liu and Yong Shi (2013). ‘The Role of Text Pre-processing in Sentiment Analysis’. In: *Procedia Computer Science* 17, pp. 26–32.

Hadikusumo, BHW and Steve Rowlinson (2004). ‘Capturing safety knowledge using design-for-safety-process tool’. In: *Journal of construction engineering and management* 130.2, pp. 281–289.

Halim, S Zohra et al. (2018). ‘In search of causes behind offshore incidents: Fire in offshore oil and gas facilities’. In: *Journal of Loss Prevention in the Process Industries* 54, pp. 254–265.

Han, Hui, Wen-Yuan Wang and Bing-Huan Mao (2005). ‘Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning’. In: *International conference on intelligent computing*. Springer, pp. 878–887.

Hasanin, Tawfiq et al. (2019). ‘Severely imbalanced big data challenges: investigating data sampling approaches’. In: *Journal of Big Data* 6.1, pp. 1–25.

Hassler, Marcus and Günther Fliedl (2006). ‘Text preparation through extended tokenization’. In: *WIT Transactions on Information and Communication Technologies* 37.

He, Haibo et al. (2008). 'ADASYN: Adaptive synthetic sampling approach for imbalanced learning'. In: *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence).* Ieee, pp. 1322–1328.

Herrera, Ivonne Andrade (2012). *Proactive safety performance indicators.* Norges teknisk-naturvitenskapelige universitet.

Hickman, Louis et al. (2022). 'Text preprocessing for text mining in organizational research: Review and recommendations'. In: *Organizational Research Methods* 25.1, pp. 114–146.

Hinze, Jimmie, Samuel Thurman and Andrew Wehle (2013). 'Leading indicators of construction safety performance'. In: *Safety science* 51.1, pp. 23–28.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). 'Long short-term memory'. In: *Neural computation* 9.8, pp. 1735–1780.

Holt, Allan St John (2008). *Principles of construction safety.* John Wiley & Sons.

Hoo, Zhe Hui, Jane Candlish and Dawn Teare (2017). *What is an ROC curve?*

Huang, Shigao et al. (2020). 'Artificial intelligence in cancer diagnosis and prognosis: Opportunities and challenges'. In: *Cancer letters* 471, pp. 61–71.

Hughes, Phil and Ed Ferrett (2012). *Introduction to health and safety in construction.* Routledge.

Huntingford, Chris et al. (2019). 'Machine learning and artificial intelligence to aid climate change research and preparedness'. In: *Environmental Research Letters* 14.12, p. 124007.

Isachsen Berntsen, Øyvind (n.d.[a]). *Statistisk sentralbyrå: Arbeidsulykker med dødelig utfall, etter tilsynsmyndighet og næring (SN2007) 2000 - 2022.* https://www.ssb.no/statbank/table/10913/. Accessed: 13 Oct 2023.

— (n.d.[b]). *Statistisk sentralbyrå: Rapporterte arbeidsulykker, etter kjønn, alder, fravær og næring (SN2007) 2014 - 2022.* https://www.ssb.no/statbank/table/10914/. Accessed: 13 Oct 2023.

— (n.d.[c]). *Statistisk sentralbyrå: Rapporterte arbeidsulykker, etter næring (SN2007) og type ulykke 2015 - 2022.* https://www.ssb.no/statbank/table/11343/. Accessed: 13 Oct 2023.

Jafari, P et al. (2019). 'Leading safety indicators: Application of machine learning for safety performance measurement'. In: *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction.* Vol. 36. IAARC Publications, pp. 501–506.

Jain, Sushma and Harmandeep Kaur (2017). 'Machine learning approaches to predict basketball game outcome'. In: *2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall).* IEEE, pp. 1–7.

Jin, Ruoyu et al. (2019). 'A science mapping approach based review of construction safety research'. eng. In: *Safety science* 113, pp. 285–297. ISSN: 0925-7535.

Juba, Brendan and Hai S Le (2019). 'Precision-recall versus accuracy and the role of large data sets'. In: *Proceedings of the AAAI conference on artificial intelligence.* Vol. 33. 01, pp. 4039–4048.

Jurafsky, Daniel and James H. Martin (2007). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.* Prentice Hall.

Kaur, Jashanjot and P Kaur Buttar (2018). ‘A systematic review on stopword removal algorithms’. In: *International Journal on Future Revolution in Computer Science & Communication Engineering* 4.4, pp. 207–210.

Kiviniemi, Markku et al. (2011). *BIM-based safety management and communication for building construction.* VTT Technical Research Centre of Finland.

Kjellen, Urban and Eirik Albrechtsen (2017). *Prevention of accidents and unwanted occurrences: Theory, methods, and tools in safety management.* CRC Press.

Krzeszewska, Urszula, Aneta Poniszewska-Marańda and Joanna Ochelska-Mierzejewska (2022). ‘Systematic Comparison of Vectorization Methods in Classification Context’. In: *Applied Sciences* 12.10, p. 5119.

Lin, Ken-Yu, Jeong Wook Son and Eddy M Rojas (2011). ‘A pilot study of a 3D game environment for construction safety education’. In: *Journal of Information Technology in Construction (ITcon)* 16.5, pp. 69–84.

Lingard, Helen and Steve Rowlinson (2004). *Occupational health and safety in construction project management.* Routledge.

Luque, Amalia et al. (2019). ‘The impact of class imbalance in classification performance metrics based on the binary confusion matrix’. In: *Pattern Recognition* 91, pp. 216–231.

Manning, Christopher and Hinrich Schutze (1999). *Foundations of statistical natural language processing.* MIT press.

Manning, Christopher D (2009). *An introduction to information retrieval.* Cambridge university press.

Manuele, Fred A (2009). ‘Leading & lagging indicators’. In: *Professional Safety* 54.12, pp. 28–33.

Martín Abadi et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* Software available from tensorflow.org. URL: https://www.tensorflow.org/.

Marucci-Wellman, Helen R, Helen L Corns and Mark R Lehto (2017). ‘Classifying injury narratives of large administrative databases for surveillance—A practical approach combining machine learning ensembles and human review’. In: *Accident Analysis & Prevention* 98, pp. 359–371.

McCullum, Nick (2020). *Deep Learning Neural Networks Explained in Plain English.* https://www.freecodecamp.org/news/deep-learning-neural-networks-explained-in-plain-english/. Accessed: 14 Dec 2023.

Mikolov, Tomas et al. (2013). ‘Efficient estimation of word representations in vector space’. In: *arXiv preprint arXiv:1301.3781.*

Mitchell, Tom M (1997). *Machine learning.*

Mohammed, Roweida, Jumanah Rawashdeh and Malak Abdullah (2020). ‘Machine learning with oversampling and undersampling techniques: overview study and experimental results’. In: *2020 11th international conference on information and communication systems (ICICS).* IEEE, pp. 243–248.

Mostue, Bodil Aamnes et al. (2022). *Ulykker i bygg og anlegg–rapport 2022. Samarbeid for sikkerhet i bygg og anlegg.* Arbeidstilsynet.

Müller, Andreas C and Sarah Guido (2016). *Introduction to machine learning with Python: a guide for data scientists.* " O'Reilly Media, Inc."

Nawaz, Sha (2023). ‘The Implications of Artificial Intelligence on Job Markets’. In: *EPRA International Journal of Multidisciplinary Research (IJMR)* 9.8, pp. 240–243.

Nielsen, Michael A (2015). *Neural networks and deep learning*. Vol. 25. Determination press San Francisco, CA, USA.

NLTK Contributors (n.d.). *Natural Language Toolkit - Stopwords Documentation*. https://www.nltk.org/search.html?q=stopwords&check_keywords=yes&area=default. Accessed: 01 May 2023.

NTNU (2023). *Sustainable Value Creation by Digital Predictions of Safety Performance in the Construction Industry: DiSCo*. https://www.ntnu.edu/iot/sustainable-value-creation-by-digital-predictions-of-safety-performance-in-the-construction-industry-disco-. Accessed: April 27 2023.

Olah, Christopher (2015). *Understanding LSTM Networks*. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed: 23 Nov 2023.

Pereira, Rafael B et al. (2018). 'Correlation analysis of performance measures for multi-label classification'. In: *Information Processing & Management* 54.3, pp. 359–369.

Pinto, Abel, Isabel L Nunes and Rita A Ribeiro (2011). 'Occupational risk assessment in construction industry–Overview and reflection'. In: *Safety science* 49.5, pp. 616–624.

Poh, Clive QX, Chalani Udhyami Ubeynarayana and Yang Miang Goh (2018). 'Safety leading indicators for construction sites: A machine learning approach'. In: *Automation in construction* 93, pp. 375–386.

Raffel, Colin et al. (2020). 'Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer'. In: *Journal of Machine Learning Research* 21.140, pp. 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

Rane, Nitin (2023). 'Role of ChatGPT and Similar Generative Artificial Intelligence (AI) in Construction Industry'. In: *Available at SSRN 4598258*.

Rausand, Marvin (2013). *Risk assessment: theory, methods, and applications*. Vol. 115. John Wiley & Sons.

Read, Jesse, Antti Puurula and Albert Bifet (2014). 'Multi-label classification with meta-labels'. In: *2014 IEEE international conference on data mining*. IEEE, pp. 941–946.

Roughton, James and Nathan Crutchfield (2013). *Safety Culture: An Innovative Leadership Approach*. Butterworth-Heinemann.

Rozenfeld, Ophir et al. (2010). 'Construction job safety analysis'. In: *Safety science* 48.4, pp. 491–498.

Russell, Stuart J. and Peter Norvig (2010). *Artificial intelligence a modern approach*. Pearson Education, Inc.

Safety, Occupational and Health Administration (2011). 'Construction focus four training'. In: *US Dept. of Labor, Washington, DC*.

Saha, Sourav et al. (2021). 'Hierarchical deep learning neural network (HiDeNN): An artificial intelligence (AI) framework for computational science and engineering'. In: *Computer Methods in Applied Mechanics and Engineering* 373, p. 113452.

Saito, Takaya and Marc Rehmsmeier (2015). 'The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets'. In: *PloS one* 10.3, e0118432.

Santos, Miriam Seoane et al. (2018). 'Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches [research frontier]'. In: *ieee ComputatioNal iNtelligeNCe magaziNe* 13.4, pp. 59–76.

Santosh, K. C. and Ravindra S. Hegadi (2018). *Recent Trends in Image Processing and Pattern Recognition.* Springer Nature Singapore Pte Ltd. 2019.

Sarica, Serhad and Jianxi Luo (2021). 'Stopwords in technical language processing'. In: *Plos one* 16.8, e0254937.

Schofield, Alexandra, Måns Magnusson and David Mimno (2017). 'Pulling out the stops: Rethinking stopword removal for topic models'. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, short papers*, pp. 432–436.

SentencePiece Contributors (2023). *SentencePiece Documentation.* https://github.com/google/sentencepiece. Accessed on 26 Nov 2023.

Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding machine learning: From theory to algorithms.* Cambridge university press.

Sofaer, Helen R, Jennifer A Hoeting and Catherine S Jarnevich (2019). 'The area under the precision-recall curve as a performance metric for rare binary events'. In: *Methods in Ecology and Evolution* 10.4, pp. 565–577.

Sokolova, Marina and Guy Lapalme (2009). 'A systematic analysis of performance measures for classification tasks'. In: *Information processing & management* 45.4, pp. 427–437.

Stock, James H and Mark W Watson (2008). *Business cycles, indicators, and forecasting.* Vol. 28. University of Chicago Press.

Sutskever, Ilya (2013). *Training recurrent neural networks.* University of Toronto Toronto, ON, Canada.

Sutskever, Ilya, Oriol Vinyals and Quoc V Le (2014). 'Sequence to sequence learning with neural networks'. In: *Advances in neural information processing systems* 27.

Svensli, Eva and Ingvild Solberg (2016). 'Nytteverdi av sikker-jobb-analyse i bygg-og anleggsprosjekter'. MA thesis. NTNU.

Thoma, Martin (2018). *Evaluation of binary classifiers.* https://martin-thoma.com/binary-classifier-evaluation/. Accessed: 15 Nov 2023.

Tinmannsvik, RK, E Albrechtsen and K Wasilkiewicz (2016). *SJA Sikker jobbanalyse. Et opplæringshefte.*

Tixier, Antoine J-P et al. (2016). 'Application of machine learning to construction injury prediction'. In: *Automation in construction* 69, pp. 102–114.

Tsoumakas, Grigorios and Ioannis Katakis (2007). 'Multi-label classification: An overview'. In: *International Journal of Data Warehousing and Mining (IJDWM)* 3.3, pp. 1–13.

Uddin, SM et al. (2020). 'Hazard recognition patterns demonstrated by construction workers'. In: *International Journal of Environmental Research and Public Health* 17.21, p. 7788.

Uddin, SM Jamil (2023). 'Leveraging Social Media Platforms and Generative AI Technologies for Construction Industry Applications'. PhD thesis. North Carolina State University.

Van Derlyke, Peter, Luz S Marin and Majed Zreiqat (2022). 'Discrepancies between implementation and perceived effectiveness of leading safety indicators in the US dairy product manufacturing industry'. In: *Safety and health at work* 13.3, pp. 343–349.

Vaswani, Ashish et al. (2017). 'Attention is all you need'. In: *Advances in neural information processing systems* 30.

Vujović, Ž et al. (2021). 'Classification model evaluation metrics'. In: *International Journal of Advanced Computer Science and Applications* 12.6, pp. 599–606.

Winge, Stig and Eirik Albrechtsen (2018). 'Accident types and barrier failures in the construction industry'. eng. In: *Safety science* 105, pp. 158–166. ISSN: 0925-7535.

Winge, Stig, Eirik Albrechtsen and Bodil Aamnes Mostue (2019). 'Causal factors and connections in construction accidents'. eng. In: *Safety science* 112, pp. 130–141. ISSN: 0925-7535.

Xia, Wei et al. (2019). 'High-resolution remote sensing imagery classification of imbalanced data using multistage sampling method and deep neural networks'. In: *Remote Sensing* 11.21, p. 2523.

Xu, Yun and Royston Goodacre (2018). 'On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning'. In: *Journal of analysis and testing* 2.3, pp. 249–262.

Yogish, Deepa, TN Manjunath and Ravindra S Hegadi (2019). 'Review on natural language processing trends and techniques using NLTK'. In: *Recent Trends in Image Processing and Pattern Recognition: Second International Conference, RTIP2R 2018, Solapur, India, December 21–22, 2018, Revised Selected Papers, Part III 2*. Springer, pp. 589–606.

Zhang, Min-Ling et al. (2018). 'Binary relevance for multi-label learning: an overview'. In: *Frontiers of Computer Science* 12, pp. 191–202.

Zhang, Sijie, Frank Boukamp and Jochen Teizer (2015). 'Ontology-based semantic modeling of construction safety knowledge: Towards automated safety planning for job hazard analysis (JHA)'. In: *Automation in Construction* 52, pp. 29–41.

Zhu, Min et al. (2018). 'Class weights random forest algorithm for processing class imbalanced medical data'. In: *IEEE Access* 6, pp. 4641–4652.

Appendix

A Appendix Exhibit A - Laws from The Norwegian Labour Inspection Authority

§ 3-2. Særskilte forholdsregler for å ivareta sikkerheten

(3) Hvis det skal utføres arbeid som kan innebære særlig fare for liv eller helse, skal det utarbeides en skriftlig instruks om hvordan arbeidet skal utføres og hvilke sikkerhetstiltak som skal iverksettes.

§ 10-4. Krav til utstyrsspesifikk opplæring

Arbeidsgiver skal sørge for at arbeidstaker får nødvendig opplæring på det spesifikke arbeidsutstyret vedkommende skal bruke. Opplæringen skal tilpasses arbeidsutstyrets art og sikre at arbeidstakeren kan bruke arbeidsutstyret på en forsvarlig måte. Det skal dokumenteres skriftlig hvilket arbeidsutstyr det er gitt opplæring på, hvem som har gitt opplæringen og hvem som har fått opplæring.

B Appendix Exhibit B - Translations of Work Operations

Original Norwegian words	Translated English
Annet - definer	Other - define
Løfteoperasjoner	Lifting operations
Riggarbeider	Rig work
Råbygg/prefab.	Shell construction/ prefabrication
Grunnarbeider	Foundation work
Tett bygg/fasader	Dense buildings/facades
Grøfter/kummer/rør	Shafts/wells/pipes
Innvendige arbeider	Interior works
Sprengning/masseforflytning	Blasting/mass transfer
Betongarbeider	Concrete work
Bruer/kaier	Bridges/quays
Tunneler	Tunnels
Utomhusarbeider	Outdoor work
Riving og/eller sanering	Demolition and/or remediation

C Appendix Exhibit C - Translation of Hazards

Actual Norwegian words	Translated English
Fallende gjenstand	Falling object
Bevegelige gjenstander/klemfare	Moving objects/crushing hazards
Fall fra høyde	Fall from height
Sammenstøt/påkjørsel	Collision
Elektriske støt	Electrical shocks
Konstruksjonssvikt	Structural failure
Tunge løft/tunge materialer	Heavy lifting/heavy materials
Brann, eksplosjon	Fire, explosion
Høyt trykk, sprutfare	High-pressure hazards/splash risks
Fare for å snuble eller skli	Risk of tripping or slipping
Værforhold (vind, kulde, tåke)	Weather conditions (wind, cold, fog)
Drukning	Drowning
Støv, røyk, gasser, giftige stoffer	Dust, smoke, gases, toxic substances
Utslipp/forurensning	Emissions/pollution
Skarp gjenstand (kutt, stikk)	Sharp object hazards (cuts, punctures)
Støy, vibrasjon	Noise, vibration
Arbeid i lukkede rom/oksygenmangel	Confined space work/oxygen deficiency hazards
Biologisk helsefare	Biological health hazard
Kollaps av gravegrop	Collapse of excavation pit
Overflater med høy/lav temperatur	Surfaces with extreme temperatures (high/ low)

D Appendix Exhibit D - Translation of JSA causes

Norwegian words	Translated English
Konsekvensen er alvorlig dersom det skjer noe galt	The consequences are serious if something goes wrong
Arbeidet innebærer risiko for helseskade/skade	The work poses a health or injury risk
Ulykker/uønskede hendelser har skjedd tidligere ved tilsvarende aktiviteter	Accidents or unintended incidents have occurred in the past during similar activities
Aktiviteten er ny og ukjent	The activity is new and unknown
Folk som ikke kjenner hverandre skal utføre en kritisk jobb sammen	Individuals unfamiliar with each other must collaborate on a critical task
Forutsetninger for aktiviteten er endret (værforhold, tid, rekkefølge, andre aktiviteter i nærheten)	The conditions for the activity have changed (weather conditions, time, sequence, other nearby activities)
Arbeidet medfører avvik/endringer fra beskrivelser i prosedyrer og planer	The work deviates or changes from the established procedures and plans
Behov for å gi og dokumentere utstyrtspesifikk opplæring	There is a need to provide and document training specific to the equipment
Annet...	Other...
Behov for å gi dokumentert-/ utstyrtspesifikk opplæring	Need to provide documented/equipment-specific training

Translation of reasons to make JSA

E Appendix Exhibit E - Hyperparameteres for Good Example Algorithm

Model	Hyperparameters	Values
Random Forest	n_estimators	[10, 50, 100, 200]
	max_depth	[None, 10, 20, 30]
	min_samples_split	[2, 5, 10]
	min_samples_leaf	[1, 2, 4]
	bootstrap	[True, False]
	criterion	Gini
Gradient Boosting	n_estimators	[50, 100, 200, 300]
	learning_rate	[0.01, 0.05, 0.1, 0.5]
	max_depth	[3, 4, 5, 6]
	max_features	['auto', 'sqrt', 'log2', None]
	loss	log_loss
LSTM	loss	'binary_crossentropy'
	optimizer	'adam'
	embedding size	32
	units	128
	activation	'sigmoid'
	early_stopping	(monitor='loss', patience = 3)
word_2_vec	vector size	200
	window	5
	min_count	1

Hyperparameter Optimization Ranges in the Good Example Algorithm

F Appendix Exhibit F - Hyperparameters for Hazard Classification Model

Model Component	Hyperparameters/Settings	Values
Categorical Encoding	Work operation	OneHotEncoder()
	Subject area	OneHotEncoder()
	Hazards	Multi Label Binarizer (mlb)
Data Splitting	Test Size	0.2
	Training Size	0.8
	Random State	42
LSTM Model	Embedding Dimension	100
	LSTM Units	64
Categorical Inputs	Dense Layer Units	32
	Activation Function	relu
Layers	Dense Layer Units	128
	Output Activation Function	sigmoid
Model Compilation	Loss Function	binary_crossentropy
	Optimizer	adam
	Metrics	accuracy, weighted accuracy
Model Training	Epochs	10
	Batch Size	32

Hyperparameters and Settings for Simplified Hazard Classification Model

Model Component	Hyperparameters	Values
General	Test Size	0.2
	Training Size	0.8
	Random State	42
	Number of Folds (KFold)	5
LSTM Model	Embedding Dimension	200
	LSTM Units	64
Categorical Inputs	Dense Layer Units	32
	Activation Function	relu
Concatenation Layer	Dense Layer Units (Layer 1)	128
	Dense Layer Units (Layer 2)	64
	Activation (Layer 1 and 2)	relu
	Dropout Rate	0.5
Model Training	Loss Function	binary_crossentropy
	Optimizer	adam
	Metrics	accuracy, weighted accuracy
Training Parameters	Epochs	10
	Batch Size	32

Hyperparameters for Complex Hazard Classification Model

G Appendix Exhibit G - Hyperparameters for Preventative Measures Generation

Model	**Hyperparameters**	**Values**
T5 Model	Model type	Base
	Training data	70%
	Validation data	20%
	Test data	10%
	Tokenizer max_length	150
	Tokenizer truncation	True
	Tokenizer padding	max_length
T5 Training Args	num_train_epochs	3
	per_device_train_batch_size	32
	per_device_eval_batch_size	64
Prediction Generation	max_length	300
	num_beams	6
	early_stopping	True

Hyperparameters for Training T5 Model

H Appendix Exhibit H - Original language: Most Confident Misclassifications

Activity description	Label By algorithm	Certainty	Actual label
Montasje av 4 stk sandwichvegger i akse B1, elementene skal monteres på plasstøptvegg 6,8 meter over gulv på grunn. planlagt montasje 02.03.22.	Bevegelige gjenstander/klemfare	0.8485	Fallende gjenstand
Vi skal sprenge ned fjell mot bygg (ca 8 m høgt) der fjellet er ca 1-2 m fra bygg, der vi salver at vi tar 4 m pallhøyde. Mest kritisk plass er sikret med borstenger.	Brann, eksplosjon	0.5115	Bevegelige gjenstander/klemfare, Fall fra høyde, Fallende gjenstand, Høyt trykk, sprutfare
Etter sprengning har det dannet seg sprekker i mur mot eksisterende tanker, denne skal fjernes på oppdrag fra byggherre.	Brann, eksplosjon	0.3962	Fall fra høyde, Bevegelige gjenstander/klemfare
Vs har fått i oppdrag og fylle kanal inne på [Anonymized], her skal vi legge 1200btg på -0,70 og montere 3 kummer.	Drukning	0.8204	Bevegelige gjenstander/klemfare, Støv, røyk, gasser, giftige stoffer, Utslipp/forurensning
Sprengning av grøft/vei under og langs høyspentledning. [Anonymized] er eier av høyspentledningen som fører 420kV. Grøften er ca 70 meter. Det blir lagt 2 lag og 18 matter for å forhindre spredning av stein. Salvene blir justert etter forholdene med vanndammene. Det blir en del små salver.	Elektriske støt	0.7540	Fallende gjenstand, Høyt trykk, sprutfare, Bevegelige gjenstander/klemfare
Arbeidet skal foregå delvis på stillas og oppå glasstak. Glass skal justeres i bære profiler og sikres med klemmlister/profiler. I tillegg skal det monteres profil i front av glasstak fra stillaset. Glass er allerede heis på plass og ligger nede i bæreprofilene. Oppå bæreprofilene skal det legges plater for å gå på når klemlister skal monteres.	Fall fra høyde	0.9257	Fare for å snuble eller skli, Fallende gjenstand
Montering av ventilasjons aggrigat på taket av bygg A, Aggrigatet løftes opp på fundamentet som er bygget på taket. Hver seksjon av aggrigatet monteres sammen og deretter settes dette sammen og sideforskyves på plass av KBS.	Fallende gjenstand	0.9897	Bevegelige gjenstander/klemfare
Graver skal arbeide nær gassledning. Alle må være klar over prosedyre dersom gasslekkasje oppstår.	Høyt trykk, sprutfare	0.4585	Brann, eksplosjon
Montering av stålbjelker for bæring av moduler	Konstruksjonssvikt	0.7466	Fallende gjenstand, Sammenstøt/påkjørsel, Værforhold (vind, kulde, tåke)
Ombordkjøring/ilandkjøring og tipping fra flatlekter til sjø	Sammenstøt/påkjørsel	0.7720	Drukning
Sage ned busker og trær og kjøre gjennom kompostkvern	Skarp gjenstand (kutt, stikk)	0.7862	Fallende gjenstand, Bevegelige gjenstander/klemfare
Sikkert arbeid med pigging og annet arbeid i såle tunnel	Støv, røyk, gasser, giftige stoffer	0.8353	Brann, eksplosjon
Vegger må vendes med tårnkran og mobilkran	Tunge løft/tunge materialer	0.8664	Konstruksjonssvikt, Bevegelige gjenstander/klemfare
Plastring i sjø, foran Kai	Utslipp/forurensning	0.6845	Drukning, Bevegelige gjenstander/klemfare
Montering og heising av stålbjelker	Værforhold (vind, kulde, tåke)	0.7712	Støv, røyk, gasser, giftige stoffer

The Most Confident Misclassifications by the Model - Original Norwegian text

NTNU

Norwegian University of
Science and Technology