NAMIBIA UNIVERSITY OF SCIENCE AND TECHNOLOGY

EXPLORING MACHINE LEARNING ON GEOCHEMISTRY DATA FOR EFFICIENT PREDICTION OF METAL CONCENTRATIONS IN COPPER DEPOSITS

by

Lydia Joel

222085010

Submitted in partial fulfilment of the requirements for the degree of

Master of Data Science

in the

Department of Informatics

at the

NAMIBIA UNIVERSITY OF SCIENCE AND TECHNOLOGY

Supervisor:

Dr R. Maliwatu

Date of submission:

(25th January 2024)

METADATA

Research Title:	Exploring machine learning on geochemistry data for efficient prediction of metal concentrations in copper deposits
Student full name:	Lydia Joel
Supervisor:	Richard Maliwatu
Department:	Informatics
Qualification:	Masters of Data Science
Main knowledge area:	Machine Learning
Keywords:	Machine Learning, Metal Concentration, Mineralisation, Geochemistry data
Research Type:	Design, Case Study Research
Methodology:	Mixed methods
Status:	Dissertation
Site:	Namibian University of Science and Technology
Document date:	25 January 2024
Sponsor	None

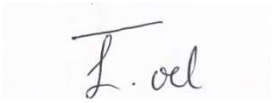
DECLARATIONS

Student

I, Lydia Joel, hereby declare that the work contained in this research for the Master's degree project, entitled "EXPLORING MACHINE LEARNING ON THE GEOCHEMISTRY DATA FOR EFFICIENT PREDICTION OF METAL CONCENTRATIONS IN COPPER DEPOSITS," is my own original work and that I have not previously, in its entirety or in part, submitted it at any university or other higher education institution for the award of a Masters.

I additionally certify that, in compliance with the Institution's policies, I have given full credit to all information sources used in the study. All additional contributions to this work have been duly acknowledged and noted.

I understand that any act of plagiarism or academic misconduct is a breach of trust and undermines the integrity of both the academic community and the knowledge creation process.



Signature_____ Date_____31st January 2024_____

Supervisor

I, Dr Richard Maliwatu confirms that this dissertation was prepared under my supervision.

Signature_____ Date____31st January 2024_____

ABSTRACT

Naturally occurring ore bodies like Copper often occur in compound form with other useful metals such as Silver, Lead and Zinc. Due to the cost, mining companies find it difficult to pay for analysis of various metals in their samples and end up focusing on analysing one metal or a few, leaving out a bunch of other associated metal concentrations in the deposit. Additionally, analysing different metals in samples can take time, and this increased turnaround time of receiving results from the laboratory can negatively affect production. The research used a geochemistry dataset comprising of 3,282 samples from the Kombat Copper deposit area in Namibia to predict copper (Cu) concentrations from zinc (Zn) and lead (Pb) concentrations. In addition to the metal concentrations, the dataset had sample coordinates and grid names features. The four machine learning algorithms used were Random Forest (RF), K-Nearest Neighbour (KNN), Decision Tree (DT), and Support Vector Machine (SVM). These models were used because they were the commonly employed models for similar purposes, in the literature reviewed. The learning task was a regression problem, therefore, the primary metric utilised to assess the machine learning model and draw performance conclusions was the regression score (R-squared), which quantifies how well the model explains the variance in the data. The R-squared score represents the percentage of variance in the dependent variable (target) that can be predicted from the independent variables (features). It ranges on a scale of 0 to 1, where 1 indicates a perfect fit. In addition Mean Squared Error (MSE), Root means squared error (RMSE), mean absolute error (MAE), Adjusted R-squared, and explained variance metrices were also looked at. Based on the R-squared metric, the KNN model outperformed the other three models, predicting 57% of the relationship between the dependent and independent variables. K-NN was followed by RF with 0.55 score, DT with a 0.49 score and the SVM with a 0.44 score. KNN model appeared to be the best choice among the four models for making predictions for the dataset. Further optimisation of the models improved their prediction accuracy, with the KNN model still with a superior performance of R-squared at 70% (0.70) with n-estimators set at 4 and the test size set to 10%. Predicting metal contents from geochemistry data with machine learning can

help mining companies reduce costs by supplementing lab-based analyses with model-based predictions in determining grades.

ACKNOWLEDGEMENTS

- I stand at the pinnacle of a remarkable journey, holding in my hands the culmination of two years of dedication, perseverance, and unwavering determination. As I reflect upon the achievement of my Masters Degree in Data Science, my heart brims with gratitude towards those who have played an invaluable role in shaping this remarkable chapter of my life.
- First and foremost, I offer my heartfelt thanks to the Almighty, whose boundless grace and unfaltering support have been the guiding lights on this path. The courage to pursue higher studies and the strength to overcome challenges have been gifts I cherish and attribute to divine benevolence.
- I extend my deepest appreciation to my research supervisor, Dr. Richard Maliwatu, whose wisdom, mentorship, and scholarly guidance have been instrumental in shaping the course of my academic journey. His unyielding commitment to excellence and his unwavering belief in my potential have fueled my aspirations and led me to new heights.
- To my beloved son, Tulonga, who at the tender age of six has been a beacon of inspiration and a wellspring of motivation, I express my profound gratitude. His innocent smile and unwavering support have been my driving force, reminding me every day that this pursuit is not just mine but a legacy I am building for us both.
- The Informatics faculty and lecturers in the department have been pillars of knowledge, enlightening my path with their expertise and fostering an environment of intellectual growth. I am indebted to their dedication and tireless efforts in imparting knowledge that has shaped my academic foundation.
- A special mention goes to my dear classmate, Lucia Kalipi, whose unwavering friendship, collaborative spirit, and boundless encouragement have transformed challenges into opportunities and hurdles into stepping stones. Her camaraderie has added colors of joy to this academic expedition, and I am grateful beyond words.

- To my family and friends, who have stood by me through the late nights of study, the moments of doubt, and the celebrations of achievements, I offer my sincere appreciation. Your unwavering support, patience, and belief in my abilities have been the wind beneath my wings.
- This Master's Degree is not just a testament to my efforts, but a symphony of support, guidance, and love from those who have played an indispensable role in my journey. As I step into the next chapter of my life, I carry forward these experiences, lessons, and relationships, armed with a newfound zeal to contribute meaningfully to the world of Data Science.

With boundless gratitude, Lydia

DEDICATION

In the tapestry of life, there are threads that weave moments of triumph and challenge, resilience and growth. As I stand here, holding the coveted emblem of my Masters Degree in Data Science, I am acutely aware of a thread that has run through every chapter of this journey – the memory of my late dad (Mr Theophelus Joel).

This degree is more than an academic milestone; it is a testament to the values, lessons, and unwavering support that my dad bestowed upon me. Though he is no longer here to witness this achievement, his presence has served as a constant source of motivation for me to persevere in my intense quest of knowledge and to conquer hurdles.

My dad, a beacon of wisdom and an epitome of hard work, instilled in me the belief that education is not just a means to an end, but a gateway to endless possibilities. He ignited in me a passion for learning, encouraging me to embrace challenges with tenacity and strive for excellence with unwavering determination.

In the solitude of late-night study sessions and in the triumphs of each breakthrough, I have felt his spirit beside me, urging me to push the boundaries of my capabilities. Every step I took on this academic path was guided by his memory, and every success I achieved is dedicated to his legacy.

As I accept this Masters Degree, I do so with a heart filled with gratitude for the countless sacrifices my dad made to pave the way for my education. This degree stands not just as a personal achievement, but as a tribute to his enduring love, his unyielding faith, and the indelible mark he has left on my journey.

In dedicating this Masters Degree to my late dad, I honor his memory, celebrate his unwavering belief in me, and carry forward the torch of his aspirations. This achievement is a reflection of the legacy he has left behind – one of resilience, courage, and an unwavering commitment to shaping a better future.

With heartfelt dedication,

Your daughter, Lydia

LIST OF PUBLICATIONS

Joel, L. and Maliwatu, R. (2024, April 23–25). Exploring Machine Learning on Geochemistry Data for Efficient Prediction of Metal Concentrations in Copper Deposits [Paper presentation]. Industrial Engineering and Operations Management (IEOM) Society International, 2024: Pretoria, South Africa. https://index.ieomsociety.org/index.cfm/conference/view/ID/58

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF TERMS AND ABBREVIATIONS

Abbreviation	Description
Cu	Copper
DT	Decision Tree
EDA	Exploratory Data Analysis
EDN	Earth Data Namibia
GSN	Geological Survey of Namibia
K-NN	K-Nearest Neighbour
MAE	Mean Absolute Error
MSE	Mean Squared Error
Pb	Lead
ppm	parts per million
RF	Random Forest
RMSE	Root Mean Squared Error
SVM	Support Vector Machine
UTM	Universal Transversal Mercator
Zn	Zinc

CHAPTER 1: INTRODUCTION

1.1 Introduction

Mining is an important industry that contributes significantly to the global economy, since it provides raw materials needed to produce a wide range of products. Finding the precise metal content in ore samples—a crucial step in assessing the viability of a mining operation from an economic standpoint—is one of the major problems the mining industry faces. Samples are collected from the orebody and sent to laboratories to undergo various analyses to determine the different metal concentrations in the sample. Depending on the amount of metals to be analysed in samples, the laboratory analysis can be costly for small-scale mining operations and exploration companies.

Naturally occurring ore bodies like Copper often occur in association with other useful metals such as Silver, Lead and Zinc. Due to the cost, the mining industry, especially the small-scale operations and exploration companies, find it difficult to pay for analysis of various metals in their samples and typically focus on analysing one metal, which is often Copper. This approach means that companies miss out on other essential metals such as Silver, Lead, Zinc, and Nickel. Additionally, analysing different metals in samples can take time, and this increased turnaround time of receiving results from the laboratory can negatively impact production.

This research makes use of machine learning on the geochemistry data to predict the presence of copper in copper ores using zinc and lead as additional metals in these ores. Predicting different metal content from the geochemistry data with machine learning would help companies reduce costs by not analysing many metals. It would also help the companies discover other metals associated with the main commodity. These additional discovered metals through predictions can be mined, processed and sold as by-products, which translate into more profits for the companies.

1.2 Statement of the Problem

The mining industry is faced with the challenge of the high cost of laboratory analysis for multiple metal concentrations in ore samples. This results in small-scale mining operations and exploration companies focusing on analysing only one metal, which is usually copper, and missing out on other important metals such as lead, zinc and silver. Additionally, the analysis of different metals in samples can take time, which increases the turnaround time of receiving results from the laboratory and negatively affects production.

There have been attempts to address the issue, as evidenced by related work by Arslan, Aslan, and Demirci (2021) and Wang, Sun, and Cheng (2020), who demonstrated the effectiveness of machine learning in predicting the presence of associated metals in ore deposits. The researcher could not however identify any published work on the subject in Namibia. As a result, the aim of this research is to expand on previous research by exploring the potential of machine learning in the mining industry for accurate mineralisation predictions. The application of machine learning in mineral exploration and prediction can help solve these challenges by accurately predicting the presence of different metal concentrations in ore samples, thus allowing mining companies to reduce costs by not analysing many metals. It would also help the companies discover other metals associated with the main commodity. These additional discovered metals through predictions can be mined, processed, and sold as by-products, leading to more profits for the companies.

Although machine learning is being heavily applied in various domains, at the time of this study there was no published research work found in Namibia with regards to applying machine learning tools for accurately predicting metal concentrations in Copper deposits. Deposits are site-specific in that deposits from different areas differ based on their deposition modes, grades, minerals etc. Therefore, this research aims to develop a machine learning model that utilises geochemistry data to predict the presence of copper using the associated metals in these copper deposits using the geochemistry data of the copper deposits in the Kombat area, Namibia. By accurately predicting different metal contents from the geochemistry data, the cost of analysing multiple metals in ore samples can be reduced. Moreover, the mining companies can discover

additional metals associated with the main commodity, which can be mined, processed and sold as by-products leading to increased profitability. The model developed can be applied to geochemistry data from other sites and yield results that can be interpreted, based on those sites' specific characteristics.

1.3 Motivation: Facts and quantitative metrics on the problem

In Namibia, the cost of laboratory analysis for metals such as copper, zinc and lead can vary depending on the lab, the method used and the number of metals being analysed. At local laboratories, analysis of Copper alone in a sample costs $ 29 (N$ 550) using the Inductively Coupled Plasma – Mass Spectrometry (ICP-MS) analyis method. Double element analysis costs $ 37 (N$ 700) and $ 47 (N$ 900) when analysing for 4 metal concentrations. These prices are for a local Anatech Laboratory. If samples are sent to other laboratories abroad, such as in South Africa or international, additional costs such as shipping and handling fees may also apply and a cost per metal concentration analysis in a sample can get to $ 37 (N$ 700) and double metal analysis to $ 47 (N$ 900). As an example, according to ALS Global, a global testing and analytical laboratory, the cost analysis for copper, zinc, lead, and nickel can range from $28 (N$530) to $44 (N$830) per metal concentration in a sample, depending on the method used and the sample matrix (ALS, 2023) .

Small-scale mining operations and exploration companies in Namibia may also lack the expertise and equipment required to analyse multiple metals in their samples. According to a report by the International Institute for Sustainable Development (IISD), small-scale miners often lack access to specialised equipment, training, and technical support, which limits their ability to accurately assess the mineral content of their ore and negotiate fair prices with buyers (Hilson & McQuilken, 2014). This can also hinder their ability to comply with regulations and obtain necessary permits for their operations.

Depending on the workload of the lab and the complexity of the analysis, the turnaround time for laboratory analysis can also vary, but it normally takes a few days to several weeks (Dalmia & Nayak, 2019). For instance, a study in 2018 by Bortey-Sam *et al.* revealed that the typical turnaround time in Ghana for ICP-MS analysis of soil samples was almost three weeks. Sending

samples from Namibia to laboratories abroad may also be impacted by customs and shipping times and the lead time can get to about 2 to 3 months.

1.4 Contextual Focus of The Study

Figure 1 presents the study area around Kombat. The area is located in the mining town of Kombat in the Grootfontein district of the Otjozondjupa region. It is a part of Namibia's Otavi Mountain land, which is mostly known for its copper riches. The research area is roughly 45 km west of Grootfontein and 37 km east of Otavi.

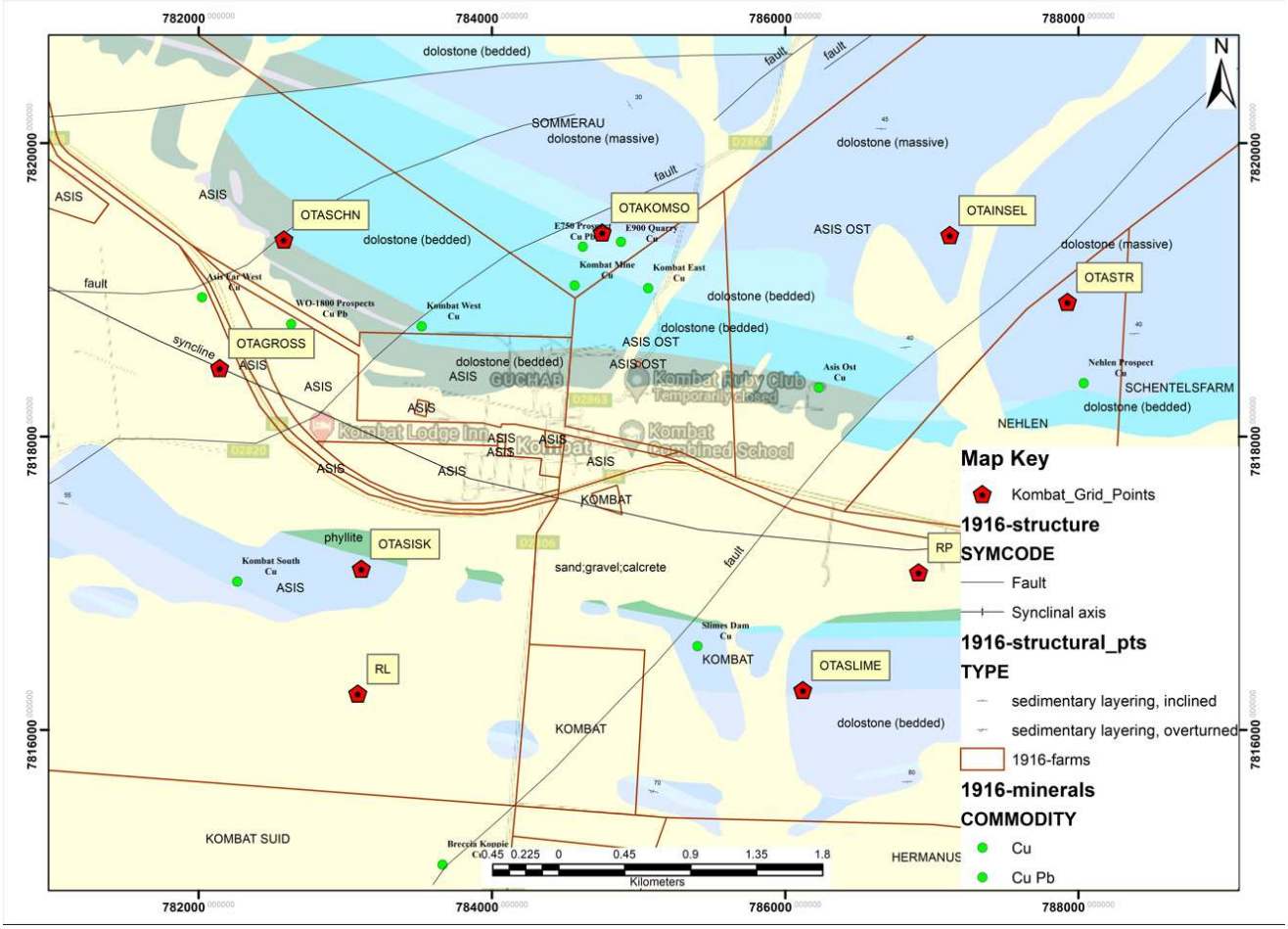


Figure 1: Area of study locality plan.

1.5 Objectives of the Study

Main Objective

The study's main objective is to explore machine learning on geochemical data and to evaluate how machine learning methods perform in predicting metal concentrations in Copper deposits.

Sub-Objectives

i. To explore the metal composition patterns in copper deposits.
ii. To evaluate performance of the four commonly used machine learning techniques (RF, DT, KNN and SVM) on geochemistry data.
iii. To determine the most suitable technique based on the performance metrics and further fine tune parameters for performance improvement.

Main research question

How can machine learning techniques be applied on samples' geochemistry data to efficiently make predictions of metal concentrations in copper deposits?

Research sub_questions

i. What are the patterns depicted from the metal concentrations in the different copper deposits?
ii. How well do the four machine learning models predict metal concentrations?
iii. Which machine learning technique is best suited for the metal concentration prediction in copper deposits based on performance?

1.6 Summary of findings

The data shows that, most concentrations are mainly below 500ppm. Further analysis shows that Pb and Zn concentrations are directly proportional to Cu concentration. For high Cu concentration, Pb concentration is in a similar range, whereas Zn concentrations are much lower. Furthermore, the data shows that Pb and Cu concentrations are predominantly in lower ranges (below 500 ppm) than the Zn concentrations. There are instances of high Cu concentrations

(above 500ppm), and these are common at Otasline, Otagross and Otainsel deposit sites while the lower Cu concentrations were more common at Oaasisk, RL, Otaschn, Otastr and RP locations. A more comprehensive discussion on achievement of sub-objective i) on metal concentration patterns in copper deposits is presented in chapter 5 section 5.4.

Evaluating how well the four machine learning techniques perform in predicting the metal concentrations was the second objective. The R -squared scores were 0.45, 0.49, 0.55 and 0.57 for SVM, DT, RF and KNN machine learning techniques respectively as presented in section 5.8.

The third sub-objective was to determine the most suitable technique based on the performance metrics and this was achieved through the comparison of the main study metric R-squared score. KNN had the best plot, as indicated by its regression line (R-squared) of 0.57. The RF came in second with an R-squared of 0.55, while SVM had the lowest R-squared of 0.45.

By comparing these metrics (R-squared score, RMSE, MAE, Adjusted R-squared, and explained variance scores) it shows that KNN technique outperforms the other three techniques. The target variable's greatest R-squared score of 0.57 it achieved means that the features account for around 0.57% of the variation in the variable. Additionally, by comparing other metrics (RMSE, MAE, Adjusted R-squared, and explained variance scores) of other models, the KNN model has the lowest RMSE (104.24) and MAE (32.83), indicating that, on average, the predictions have less mistakes. Additionally, the KNN model shows greater explained variance and adjusted R-squared score, demonstrating improved model fitting and the capacity to justify variance in target variable. Considering all these factors, the K-NN model appeared to be the best choice among the four models for making predictions in the copper deposits in the Kombat area. To further improve the prediction accuracy of this leading model (K-NN), the training and test set sizes and n-estimator parameters were further fine tuned. The best combination was when the training set was increased to 90% of the samples with test size set to 10% of the samples and n estimators set to 4 as this resulted in an R-squared score of 70% (0.70).

1.7 Significance of the Study

This work is important because it may help the mining sector overcome its difficulties in accurately determining the metal contents of ore samples. Mining companies face a significant challenge in analysing the metal content of their ore samples, as the cost of laboratory analysis is charged for each metal being analysed in the sample. This cost can be prohibitively high for small-scale mining operations and exploration companies, leading to the practice of focusing on the analysis of one metal, which can lead to missing out on other important metals. Additionally, analysing multiple metals in samples can take time, leading to delays in the receipt of results, which can negatively impact production.

By precisely predicting various metal concentrations in ore samples, machine learning applications in mineral exploration and prediction can assist address these issues. This would allow mining companies to reduce costs by analysing less metals and supplimenting lab analyses with model-based predictions to determine other associated metal concentrations.

An important achievement in the world of mineral exploration and mining is the establishment of an efficient model that can precisely predict various metal concentrations in ore samples. The potential benefits of such a model extend beyond cost reduction and increased profitability for mining companies. The ability to accurately predict the metal content of ore samples could lead to more efficient and sustainable mining practices as these discovered metals through predictions can be mined, processed, and sold as by-products, leading to more profits for the companies. Additionally, if the new mineral deposits are discovered from the metal concentration prediction, this could contribute to the global supply of critical metals. Overall, by addressing the difficulties in precisely identifying the metal content of ore samples and offering a more effective and sustainable method of mineral exploration and mining, this work has the potential to significantly benefit the mining sector.

1.8 Limitations of the Study

Since the Kombat area contains the majority of the mining copper deposits, the study's scope was limited to that area, with the specific focus to determine how well machine learning models perform using the geochemical data that was collected. Findings might be localised at the area of study level. However, the methodology used in this research can be applied to other deposit datasets with minor modifications. Whilst this approach of assessing different deposits is sensible, its effectiveness may be limited by differences in the deposits within the Kombat area.

1.9 Risk feasibility analysis and ethical considerations

The research proposal, detailing the intended methodology, was initially submitted to the university (NUST). The dataset utilised for the study was sourced from the Ministry of Mines and Energy (MME). The data is available to the public on request. Therefore, ethical clearance was not required as the study did not involve human subjects or primary data collection activities. Furthermore, the data acquistion method posed no risks, as the data was extracted from archived records in the database. The research's primary objective is scholarly and intended to be shared among the student (researcher), the research supervisor, the University, and the data custodian (Ministry of Mines and Energy) in an appropriate manner.

1.10 Dissertation Outline

This first chapter served as an introduction to the thesis, covering the problem at hand, significance of the work, constraints, aims, and questions, as well as background of the thesis and ethical clearance. The dissertation's remaining sections are arranged as follows: Literature review is covered in Chapter 2 and covers state of the art research on the difficulties the mining sector faces in identifying different metals in ore samples. It also covers the use of machine learning in mineralisation and prediction. Chapter 3 describes the methodology used to accomplish the research objectives. The outcomes and findings related to the objectives from the models are discussed in Chapter 4 while Chapter 5 concludes the dissertation and discusses possible directions for future research.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

The chapter lays out the appraisal of related work carried out for the purpose of the study and demonstrates how existing research relates to the problem being studied. The chapter include analysis of material on the subject area, methodology and tools used. The chapter will also include a summary of recent studies that address the challenge of predicting metal concentration through machine learning. It will examine studies that utilise various machine learning techniques, such as RF, SVM, DT and KNN to predict mineral deposits' metal content.

The following is how the literature review effort is contextualized: An explanation of its special goal for this particular research, comments on the earlier presentation of the broad topic of utilisation of machine learning on geochemical data to estimate metal concentration and an indication of breadth of the work provided in this chapter.

The surveying of earlier research on machine learning and metal concentration prediction was the primary goal of the literature review chapter. This was done in order to comprehend the machine learning methods that have been applied in other dissertation-like works.

Understanding the prior research conducted in this field fulfilled three other goals: First off, it reduced the possibility of overload during the study's initial phases of data collecting by offering guidance in the development of data collection instruments. A second helpful step in better understanding the problem domain was reading through similar works. The last step of the literature review process was to pinpoint chances for critically interpreting the data gathered during the data analysis phase of the study.

An outline of the research issue was given via a synthesis of the previous work. The background information for determining the needs for data collection and the identification of the data needed for the study was derived from the review. It was necessary to use a comprehensive literature review strategy for the research since the body of published work on the prediction of metal concentration in copper deposits using machine learning research at the area of study level was inadequate.

2.2 Review Scope

In order to find pertinent indexed peer-reviewed publications in significant research databases like Web of Science, Google Scholar, Scopus, and ScienceDirect, a thorough literature search was carried out. Keywords like "metal content prediction," "ore grade estimation," "machine learning," "Random Forest," "Support Vector Machine," "Decision Tree," and "K-Nearest Neighbour" were used to narrow down the search results.

To further enhance the search result, words (such as "ore grade estimation") and Boolean operators (AND, OR, and NOT) were used. Personal suggestions and forward and backward snowballing, which finds more pertinent publications based on initial search results, were two other search strategies used. Numerous journal, conference, and media articles were found regarding the topic.

Futhermore, Forward and backward chaining was employed such that citations of a particular paper were looked at to find more recent articles that have cited it. This helped with identification of newer research that has built upon the work that was being examined. References within particular papers were also looked at to find the older works that the author has used to build their argument and and this helped tracing the historical development of the topic.

With a few exceptions, when peer-reviewed papers were taken into consideration, the results were limited to peer-reviewed publications alone. Although there were papers published in other languages in the search results, particularly on Google Scholar, only English-language publications were assessed. Additionally, the evaluation period was restricted to recent work (2019–2023), while some older works from as far back as 2000 were still included. A paper's title, abstract, and conclusion are content analysed as part of the initial paper selection and review process to see if it meets the review requirements. The diagram in Figure 2 shows the requirements for the literature review. Papers meeting these requirements were carefully examined, paying close attention to the objectives, procedures followed, algorithm chosen and applied, discoveries and outcomes, conclusion, and suggestion.

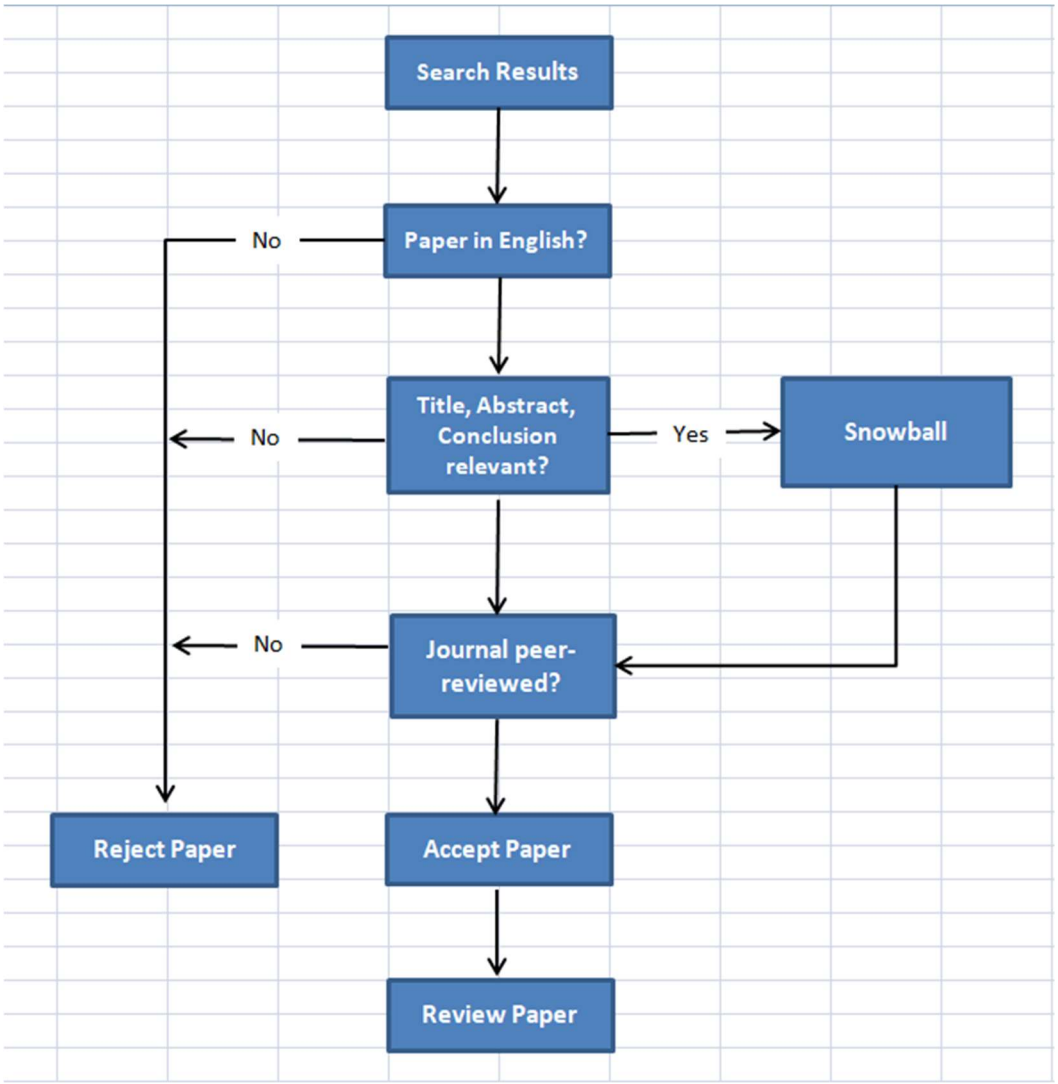


Figure 1: Literature review evaluation process schematic.

2.3 Mineralisations in Copper Deposits

Mineralisation refers to the process by which minerals, including metals, are deposited in rocks, creating concentrations of valuable elements such as copper. This process can occur through various geological mechanisms, including hydrothermal activity, magmatic processes, sedimentary processes, and metamorphism (Smith, 2018). When it comes to copper deposits, mineralisation typically involves the concentration of copper-bearing minerals within certain rock formations.

In many cases, a single rock may contain multiple metals with different metal contencentrations. The rocks in copper deposits may contain lead, zinc, and nickel, among other metals. Different metal concentrations are unevenly distributed throughout the rock. Samples are sent to the labs for analysis in order to ascertain the amounts of metals in the rocks.

2.4 Geochemical Data

Geochemical data include measurements of elements such as gold, copper, iron, zinc, and others that are commonly associated with different types of mineral deposits. The data may also include information about the abundance of certain minerals that are indicative of specific geological processes or mineralisation events.

The geochemical data functions as the input features for the machine learning model within the framework of machine learning techniques. Numerical values that represent concentrations of various elements and minerals are assigned to each sample. The system uses these data points to discover patterns and connections between the different types of mineral deposits and the geochemical properties.

2.5 Laboratory Analysis

Laboratory analysis of rocks is essential for several reasons: One of the primary objectives of analysing copper-bearing rocks is to determine the ore grade (Johnson & Brown, 2019). Ore grade refers to the concentration of copper or other metals within the rock. Understanding the metal concentration helps in assessing the economic viability of mining and processing the deposit. Secondly, laboratory analysis provides valuable information about the mineralogical and chemical composition of the rocks. This data is crucial for conducting metallurgical studies and optimising the extraction processes to achieve maximum metal recovery. Forthly, the accurate laboratory analysis helps in estimating the overall metal concentrations present in a deposit. This information is vital for mine planning and long-term resource management. However sending samples for laboratory analysis has its own challenges.

2.6 Cost and Logistical challenges

There are cost and logistical challenges associated with the laboratory analysis process: Laboratory analysis of rocks, especially for copper deposits, can be a costly and logistically challenging process due to various reasons as outlined by Johnson and Brown (2019):

a) Sample Collection: Collecting representative samples from the deposit requires careful planning and execution. The samples must be collected from different parts of the deposit to account for variations in mineralisation.

b) Sample Preparation: Once collected, the samples need to be prepared for analysis, which may involve crushing, grinding, and homogenizing. This step requires specialised equipment and skilled personnel.

c) Analytical Techniques: Rocks' mineralogy and metal concentration can be ascertained analytically using a variety of methods, including electron microscopy, inductively coupled plasma (ICP), and X-ray fluorescence (XRF). These systems can be time-consuming and frequently call for complex laboratory settings.

d) Data Interpretation: Interpreting the analytical data requires expertise in mineralogy and geochemistry. Geologists must carefully analyse the results to understand the deposit's characteristics fully.

e) Cost of Analysis: High-quality laboratory analysis using advanced techniques can be expensive, especially if a large number of samples need to be analysed. This cost can be a significant factor for exploration companies and mining operations.

f) Time Constraints: Timely analysis of samples is crucial for making informed decisions during exploration and mining operations. Delays in obtaining results can impact project timelines and budgets.

The above challenges can impact small-scale mining operations and exploration companies as they find it difficult to pay for the analysis of various metals in their samples and end up focusing on analysing only one metal, usually copper, to reduce the cost. This, in turn, leads to the

underestimation of the value of the ore deposit as other metals such as silver, lead, zinc, nickel, and silver are missed out.

Researchers have turned to machine learning techniques to predict associated metals in ore deposits, specifically those associated with copper, in order to address the challenges associated with relying solely on laboratory analysis for metal concentration determination. This is because machine learning techniques can accurately analyse and interpret large amounts of data. The ability of machine learning to predict metals in ore deposits has been shown in recent studies.

Researchers and mining corporations can greatly cut expenses and expedite their exploration operations by utilising machine learning algorithms to anticipate the presence of related metals and then estimate their concentration. The ability to accurately assess the presence and concentration of metals in ore deposits enhances the decision-making process in mineral exploration and resource assessment, leading to more effective utilisation of resources and improved efficiency in mineral exploration.

2.7 Machine Learning in metal concentration prediction

It is crucial to first determine the presence of the metal in the rocks before estimating the concentration of metals in rocks or ore samples. As previously said, applying machine learning methods can assist in ascertaining the existence of related metals. Once the metal's existence has been confirmed, more research can be done to use machine learning to forecast the metal's concentration and related metals.

Machine learning (ML) was defined by Antoine and Miranda (2017) as a technique that can recognize patterns and trends in datasets and then extrapolate predictions from those trends. SVM and RF are two popular machine learning techniques that have been utilised to forecast mineralisation (Dumakor-Dupley & Ayra, 2021).

Wenau, Spiess, Pape, and Fekete (2015), highlights that the ability of ML to continually improve the outcomes with the increase of input data into the system and not being limited by the mathematical calculations is one of the most recognized features of the tool. In addition, Cate, Perozzi, Gloguen, and Blouin (2017), mentioned that the main attracting characteristic of

Machine Learning for metal concentration prediction, is that ML requires minimal data pre-processing, secondly it is able to work with non-linear datasets, thirdly the tool is cheaper and faster, and fourthly it can handle incomplete data.

2.8 Commonly used machine learning techniques for metal concentration prediction

The goal of Adebayo *et al.* (2019) study was to forecast the existence of related metals with copper in ore resources by means of RF algorithm. Their prediction model was fed with geochemical data that was extracted from the deposit. The study's findings proved that the random forest algorithm is capable of reliably and very accurately forecasting the presence of related metals, which offers important insight into the possible existence of different metal resources in the studied area.

Liu, Ma, Ma, and Zhou (2019) looked at the applying neural network technique to forecast existence of related metals with copper in ore resources in a different study. They used geochemical data gathered from the deposit as the input for their prediction model, just like in the earlier work. The neural network method shown exceptional efficacy in accurately predicting the existence of linked metals, offering a dependable indicator of the presence or absence of particular metals in the ore deposit.

Sheng, Zhang, Niu, Wang, Tang, Duan, and Li (2015) employed RF with 100% prediction accuracy on iron ore samples. The silicon and tin emission spectral lines were utilised as input data and the ore class as the output data. With an accuracy of 100% for iron ore samples and 97.5% for the spectral data used as an input, the output Iron model showed superior predictions.

In a different study, Zaki, Chen, Zhang, Feng, Khoreshok, Mahdy, and Salim (2022) compared five machine learning algorithms, namely Gaussian Process Regression (GPR), Support Vector Regression (SVR), Decision Tree Ensemble (DTE), Fully Connected Neural Network (FCNN), and KNN, to predict highly askew gold data in a vein deposit. According to the ranking, krigging techniques are significantly outperformed by the GPR with logarithmic regularisation as the most effective technique for predicting grades.

The link between the independent and dependent variables in both procedures is indicated by the statistical parameter values of R-squared, which were found to be 0.4571 and 0.6889, respectively. The R-squared score changed to 0.8987 after fuzzy logic and neural networks were joint to form an adaptive neuro-fuzzy inference system. When testing data originate from a mixed or complicated distribution, this approach should result in a notable improvement (Zaki, *et al.,* 2022).

2.9 Model parameters

K-NN

The K value in K-NN

In KNN, the value of K is an important parameter that greatly affects the algorithm's performance. When predicting a new data point, K represents number of closest neighbours taken into account (Bansal M, Goyal A, Choundhary A, 2022). The features of the dataset and the current issue will determine which value of K to choose. The general things to keep in mind are listed below, according to Bansal M. et al (2022).

Small K (K=1, for example): The model will be susceptible to data noise and outliers. Overfitting will result from this, when the model underperforms on fresh, untried data because it catches the noise in the training set.

Big K (e.g., K=10 or more): The model smoothes over local patterns in the data yet becomes more resilient to noise. Too basic model to accurately represent the underlying structure of datasets, may result in underfitting.

The cross-validation technique is employed, according to Bansal M. *et al.* (2022), to ascertain the ideal value of K. By training and assessing the model with several values of K to determine which one works best on unseen data, the optimal K value is obtained through experimentation and validation. In this investigation, a K value of five was utilised. After further fine-tuning this parameter, the K value of the four closest neighbours produced a prediction accuracy of 70%.

Variants of K-NN

K-Nearest Neighbours has several variants and extensions that address specific challenges or adapt the algorithm for different scenarios. These variants address different challenges and trade-offs associated with the original K-NN algorithm, making them suitable for specific use cases and types of datasets (Cunningham P. & Delany S. J., 2021). The dimensionality of the data, the quantity of the dataset, and the available computer power all play a role in the variant selection process. Cunningham P. *et al.* (2021) reported a number of noteworthy variants, including:

Weighted K-NN: This variation gives each neighbour a varied weight depending on how far they are from the query point, as opposed to giving all neighbours the same weight. The prediction may be more influenced by closer neighbours.

Radius Neighbours Classifier/Regressor: This variation takes into account all neighbours within a given radius, as opposed to a fixed number of neighbours (K). When the density of data points varies across the feature space, this may be helpful.

K-Dimensional Trees: An Overview A fast data structure to arrange points in a k-dimensional space. Partitioning the space helps to expedite the process of locating nearby neighbours.

Brute-force K-NN: The simplest version of K-NN, in which each query point's distances to every point in the dataset are calculated. The technique is straightforward, but it can be costly to compute, particularly for big datasets.

Alternative data structures like the Ball Tree and Cover Tree are intended to increase the effectiveness of the nearest neighbours search in high-dimensional areas.

The K-NN regression approach, which predicts the target variable for a new data point by averaging the target values of the five nearest neighbours in space, was applied in this work in its basic version. After this parameter was adjusted even more, the four closest neighbours produced a 70% prediction accuracy.

Random Forest

In order to decrease overfitting and increase overall accuracy, the Random Forest ensemble learning method constructs several decision trees and combines their predictions (Boateng E. Y., Otoo J. & Abaye D. A., 2020). Tuning parameters in Random Forest is important inorder to optimise its performance for the dataset. Here are some key parameters that users typically tune according to Boateng E. Y., *et al.,* (2020).

n_estimators: Represents number of trees in the forest. The model's performance is often improved by tuning by adding more trees, but this comes at the expense of more computational complexity. It is customary to tune more if needed after beginning with a moderate value.

max_depth: Represents maximum depth of each individual tree. Reducing depth aids in avoiding overfitting. Although they may cause overfitting, deeper trees can capture more intricate relationships in the data. It's critical to test with a number of values in order to strike a balance.

min_samples_split: Minimal quantity of samples needed to separate an internal node. A more robust model can result from higher values because they stop small splits that capture noise. Setting it too high, meanwhile, could cause underfitting.

Min_samples_leaf: Minimum samples needed to be at a leaf node. It is comparable to min_samples_split, but it concentrates on the foliage. Larger values avoid tiny leaves, which helps to minimise overfitting.

max_features: Maximum number of features taken into account when determining the optimal split. The diversity of the trees can be impacted by managing the quantity of characteristics. Common selections include "log2" (base-2 logarithm of the total features) and "sqrt" (square root of the total features).

bootstrap: This refers to whether or not to create trees using bootstrapped samples, which are random sampling with replacement. If this is tuned to True, bootstrap sampling is enabled; if tuned to False, each tree's full dataset is used. The ensemble's trade-off between stability and diversity can be affected by adjusting this value.

random_state: This is the seed used to regulate randomization. Reproducibility is ensured by setting a specific seed. Building trees involves some randomness; however, by adjusting the seed, consistent outcomes can be achieved by running the algorithm repeatedly.

criterion: This is the unit of measurement for split quality. For regression situations, "mse" (mean squared error) is the default. It is customary to use "gini" or "entropy" for classification. The dataset's properties may influence the decision.

N_jobs: This indicates the number of jobs to run concurrently during fitting. When set to -1, all cores that are accessible are used. This can considerably accelerate training process, depending on the dataset size and available processing power. To determine a best-performing model for a particular problem, multiple combinations of parameter values are assessed in approaches like grid search and randomized search, which are frequently used to tune these parameters.

A value of 100 trees was selected for the n_estimators in this research project. Up to a certain point, performance can be enhanced by increasing number of trees. Starting point of 100 is frequently used because it offers a fair balance between computational efficiency and model accuracy.

In order to guarantee reproducibility, a random state value of 42 was also employed. The same outcomes would be produced if the model were to be run several times using the same dataset and settings. This helps with debugging, code sharing, and maintaining consistency between model evaluations.

Decision Tree

Over time, numerous decision tree algorithms have been developed, each with unique advantages and disadvantages (Somvanshi M., Chavan P., Tambade S., & Shinde S. V., 2016). The ways in which these algorithms handle numerical and categorical features, partition data, and construct trees vary. The features of the dataset, the kind of problem (classification or regression), and the particulars of the current task all influence the choice of algorithm. Some of the primary decision tree algorithms described by Somvanshi M., *et al.* (2016) are listed below:
Iterative Dichotomiser 3 (ID3): An algorithim created by dividing the dataset according to entropy or information gain using a top-down, recursive method. The main purpose of ID3 is to do categorization jobs.

ID3 is expanded upon in Classification and Regression Trees (C4.5). In order to counteract the bias of information gain towards traits with greater levels, it created the idea of information gain ratio. Regression and classification problems can both be performed with C4.5 (Charbuty B., & Abdulazeez A., 2021).

Classification and Regression Trees (CART): Although it employs distinct splitting criteria, CART is a decision tree technique that is comparable to C4.5. Regression and classification are two applications for CART. Mean squared error is the splitting criterion for regression and Gini impurity for classification.

Chi-squared Automatic Interaction Detector (CHAID): This technique determines most important feature for dataset splitting by utilising the chi-squared statistic. It is very useful for categorical data and is frequently applied in social science and market research.

Multivariate Adaptive Regression Splines, or MARS, are a kind of decision tree method even though they are not strictly speaking decision trees. MARS can capture non-linear interactions by mixing simple linear functions to form piecewise linear models. Use of it in regression problems is common.

Random Forest: The technique constructs several decision trees and aggregates the forecasts from each one. A random subset of the features and data serves as the foundation for every tree in the forest. Because it reduces overfitting, Random Forest is renowned for its great accuracy and resilience.

Gradient Boosting Machines (GBM): Gradient Boosting is an ensemble technique that creates decision trees successively, each tree repairs the faults of the previous one. This technique is known as Gradient Boosting Machines (GBM). It's an effective approach for applications involving both classification and regression. XGBoost, LightGBM, and CatBoost are well-known gradient boosting implementations.

Decision Stump: A decision tree with just one level is known as a decision stump. It's important to note that although it's not a full-fledged technique, it can be utilised in some circumstances, especially when combined with ensemble techniques like AdaBoost.

SVM

SVM is a flexible technique that may be used for a range of tasks, including regression, anomaly detection, and both linear and non-linear classification (Tanveer M, Rajani T, Rastogi R, Shao Y H, Ganaie M A, 2022). The type of data, the existence of outliers, and the selection of suitable kernel functions are some of the variables that affect the task selection and SVM's efficacy (Navada A., Ansari A., Patil S. & Sonkambe B. A, 2011). The following are some uses for SVM, as described by Navada A., *et al.* (2011):

SVM for Linear Classification: SVM is an effective linear classification algorithm. It operates by locating the hyperplane in the feature space that best divides classes. Values that are closer to the hyperplane are known as support vectors, and the objective is to maximize the margin between classes. The linear kernel is frequently employed when the data are linearly separable.

In 2D, the decision boundary is a line; in 3D, it is a plane; and in higher dimensions, it is a hyperplane.

SVM for Non-linear Classification: By utilising kernel functions, SVM may be expanded to manage non-linear classification tasks. Non-linear decision boundaries can be created by plotting the input data into higher-dimensional space using kernel functions. Most popular kernels for capturing non-linear correlations in data are polynomial, sigmoid, and radial basis function (RBF) kernels.

Support Vector Regression, or SVM for Regression: By altering the objective function, SVM can be modified for use in regression applications. SVR seeks to minimise departures from the predicted values while fitting the greatest data points in a certain margin. Epsilon-Support Vector Regression: SVR allows for a certain degree of departure from the predicted values and introduces the parameter ε (epsilon) to adjust the breadth of the margin.

SVM for Anomaly Detection: SVM can be used for anomaly detection by treating the problem as a one-class classification task. The algorithm is trained on the majority class (normal instances) and aims to identify instances that deviate from the norm. Similar to classification, non-linear relationships can be captured using kernel functions.

SVM for Multiclass Classification: While standard SVM is inherently a binary classifier, several strategies utilised to extend it to handle multiclass classification. One-vs-One (OvO) and One-vs-All (OvA) are common approaches. OvO builds a classifier for every pair of classes, and the final forecast is based on common voting. OvA builds a separate classifier for every class, and the class with the uppermost decision function output is chosen.

2.10 Standard SVM as a Binary Classification:

A binary classifier, or standard form of SVM, divides data into two groups (Probst P., Wright M N., & Boulesteix A. L., 2019). It locates the hyperplane that maximizes the difference of instances that are positive and negative. Depending on which side of the hyperplane a new data point falls,

the decision function allocates it to one of the two classes. A number of techniques, including OvO and OvA, can be used to expand SVM for multiclass classification, as was previously indicated. Soft margin support vector machines (SVM) impose a penalty for instances that are on the incorrect side of the margin or hyperplane when the data is not perfectly separable (Probst P. *et al.,* 2019). This makes the model more durable and versatile.

Support vector regression (svr) version was used in this investigation. Finding a hyperplane in a higher dimensional space that best represents the connection between the input and output variables is the goal of the variation, specifically made for regression problems. Given that a linear kernel function was employed, a straight line is assumed to be the decision boundary. The linear kernel was chosen since it is anticipated that the study's input and output properties will have a roughly linear relationship.

2.11 Literature Review Summary

Table 1 below presents a summary of the key relevant literature documents related to machine learning and prediction of metal contents in deposits that were critically reviewed. An article was considered relevant if it covered work on machine learning and its application in the mineral resources domain.

Table 1: Summary of key literature reviewed

	Author(s)/Article Bibliographic details	How it was done - sampling, data collection, model, analysis	What were the findings and results
1	Nwaila, Zhang, Frimmel, Manzi, Dohm, Durrheim and Tolmay (2020): Local and target exploration of conglomerate-hosted gold deposits using machine learning algorithms: a case study of the Witwatersrand gold ores, South Africa.	The study integrated sedimentological and gold assay data. GS-Pred could forecast the gold grades at any position within its spatial coverage.	GS-Pred outperforms both standard kriging and existing machine learning algorithms in terms of accuracy. Clustering result maximizes the contrast in the inter-cluster prediction behavior. The clusters have a good spatial relationship with the known geology.
2	Zhang, Nwaila, Tolmay, Frimmel and Bourdeau (2021): Integration of machine learning algorithms with Gompertz Curves and Kriging to estimate resources in gold deposits.	This application combines sequential-kriging block modelling with machine learning methods to provide in situ grade estimation.	Good GS-Pred performance and flexibility compared to the original algorithms. Findings support the strong sedimentological control of Au content in the Witwatersrand Basin and are appropriate for numerical forecasting.
3	Kaplan and Topal (2020): A new ore grade estimation using combine machine learning algorithms.	The KNN algorithm was used to predict the rock kinds and changes at unsampled areas prior to grade estimation.	Accurate prediction grades on a test dataset with an MAE of 0.507 and R-squared = 0.528. In contrast, the traditional model, which solely utilises sample point coordinates as input, produced an MAE value of 0.862 and R-squared = 0.112.
4	Farhadi, Afzal, Boveiri, Konari, Daneshvar Saein and Sadeghi (2022): Combination of Machine Learning Algorithms with Concentration-Area Fractal Method for Soil Geochemical Anomaly Detection in Sediment-Hosted Irankuh Pb-Zn Deposit.	Three metrics—the R-squared, MAE and MSE have been used. The HML model beat previous algorithms by incorporating the benefits of individual regression models, resulting in prediction of Pb and Zn grades.	Core drilling data and mining activity were associated with the main anomalous locations of these components. The findings suggest that the approach to ore elemental distribution prediction shows promise.

5	Jafrasteh, Fathianpour and Suárez (2018): Comparison of machine learning methods for copper ore grade estimation.	The prediction accuracy is assessed on test instances situated in drill holes distinct from the training data to guarantee that these comparisons are pertinent and realistic.	Accurate forecasts were produced by specifically constructed Gaussian processes with an anisotropic kernel and a symmetric standardization of the spatial location inputs. Notable enhancements were achieved when the collection of predictor variables incorporates data on the type of rock in addition to location.
6	Zaki, Chen, Zhang, Feng, Khoreshok, Mahdy and Salim (2022): A Novel Approach for Resource Estimation of Highly Skewed Gold Using Machine Learning Algorithms.	MLA's accuracy is contrasted with geostatistical methods like indicator and conventional kriging. Normalization techniques (z-score and logarithmic) were applied to the pre-processed data to improve network training performance and reduce significant variations in the dataset's variable ranges on predictions.	Kriging techniques significantly outperformed by the GPR with logarithmic normalization as the most effective method for predicting gold grade.
7	Application of Adaptive Neuro-Fuzzy Inference System for Grade Estimation; Case Study, Sarcheshmeh Porphyry Copper Deposit, Kerman, Iran	The adaptive neuro-fuzzy inference system's parameters are iteratively adjusted through a hybrid learning process throughout training. Additionally, a comparison between this new technique (ANFIS) and other techniques (ANN and Krigging) was done.	The statistical parameter values of R were found to be 0.4571 and 0.6889 for ANN and Krigging respectively. The R2 value changed to 0.8987 after fuzzy logic and neural networks were merged to create an adaptive neuro-fuzzy inference system.
8	Dumakor-Dupey and Arya (2021): Machine Learning—A Review of Applications in Mineral Resource Estimation.	Studies that compared and contrasted machine learning and conventional methods.	Machine learning models outperform traditional methods in accommodating several geological parameters and approximating complex nonlinear interactions.

9	Sun, Li, Wu, Chen, Zhu and Hu (2020): Data-driven predictive modelling of mineral prospectivity using machine learning and deep learning methods: a case study from southern Jiangxi Province, China.	The prediction models were evaluated using a confusion matrix, receiver operating characteristic curve, and success-rate curve to provide a thorough evaluation.	The CNN model (92.38% accuracy) outperforms the RF model (87.62%) in terms of classification ability. RF model performs better than the other ML models.
10	Chatterjee, Bhattacherjee, Samanta and Pal (2006): Ore grade estimation of a limestone deposit in India using an artificial neural network.	The NN model integrated both the spatial position and the lithological information.	This analysis showed that the NN model beat the kriging model by a significant margin.
11	Sayom, Mfenjou, Ngounouno, Etoundi, Boroh, Ngueyep and Meying (2023): A coupled geostatistical and machine learning approach to address spatial prediction of trace metals and pollution indices in sediments of the abandoned gold mining site of Bekao, Adamawa, Cameroon.	Samples of surface sediment totalling thirty-one (31) were taken in to calculate overall amounts of As, Cr, Cu, Fe, Mn, Ni, Pb, Sn, and Zn. The sediment pollution index (SPI), the Nemerow index (NI), the modified contamination degree (mCD), and the prospective ecological risk assessment (RI) are among the pollution indices that are calculated using these trace metals.	The ANN model is the most effective technique for predicting trace metal concentrations and pollution indices in sediments.
12	Caté, Perozzi, Gloaguen and Blouin (2017): Machine learning as a tool for geologists.	Six machine learning algorithms have been applied to forecast the presence of gold mineralisation in drill core.	An appropriate success rate can be achieved in the identification of gold-bearing intervals by combining ensemble machine learning algorithms with a set of rock physical parameters that are measured at closely spaced intervals in drill cores.

CHAPTER 3: METHODOLOGY

3.1 Introduction

The approach employed for the research is presented in this chapter. The chapter also covers the methods, tools, data analysis techniques and research design that were employed in the study. This research's methodology draws from Saunders, Lewis and Thornhill (2017)'s research onion model as per Figure 3. The layered model helps to explain the decisions and guides the researcher to make right decisions regarding the data to be used.

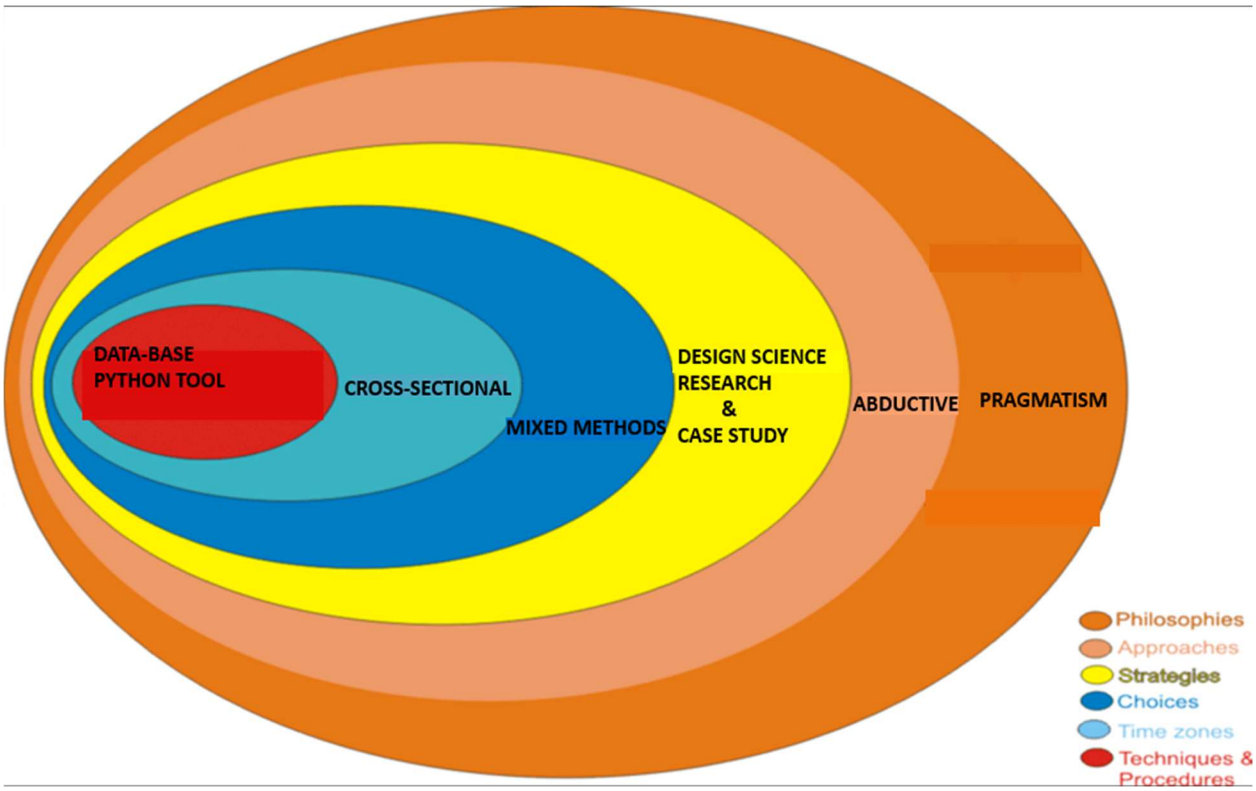


Figure 2: The methodology based on Saunders, Lewis and Thornhill (2019)'s Research Onion.

3.2 Research philosophy

This study utilised pragmatism as the research philosophy. According to Glasgow (2013) and Saunders *et al.* (2019), pragmatism research deals with application and context. Since the goal of this research was to attempt solve the real problem in the mining industry, the pragmatist view was suitable for this research. Glasgow (2013) defines pragmatism as the process of repurposing theory from practice, which aligns well with this research. Saunders *et al.* (2019) endorses this by

claiming that pragmatism contributes practical solutions that result in future decisions. This indeed agrees with the context of this research, as the work entailed usage of already known assay values in order to determine the connection between the two and apply machine learning to predict metal concentration of the other metals.

3.3 Approach

Saunders *et al.* (2019) defines research approach as the use of theory in research. The research employed the abductive approach, which involves both inductive and deductive approaches. With the abductive approach, collection of data takes place, after which trends are identified and explained. For this research data was collected and trends in the data were explored and explained as per the sub-objective i) of the study. The approach is flexible in nature and it is therefore mostly underpinned by pragmatism. According to Saunders *et al.* (2019), abductive approach allows creation of new theories or modification of existing theories.

3.4 Research Paradigm

The research adopted the regulation perspective paradigm dimension. According to Saunders *et al.* (2019), regulation research dimension establishes how an operation's affairs can be improved with the procedures already in place. This is in line with the goal of the research, which is to use machine learning to the current setup rather than to question or completely replace the current method of using laboratory analysis to check for metal concentrations.

3.5 Research Metholodical Choice

This study employs a quantitative methodology, which De Vos *et al.* (2002) defines as an investigation into a problem by testing a theory made up of numerical variables that are statistically analysed to ascertain whether the theory's predictive generalisations come to pass. When interpreting quantitative data, figures are frequently viewed as solid proof of how a phenomenon worked or did not happen. It is further claimed, on the basis of these premises, that it is challenging to support numerical facts with theory.

This research simply adhered to the machine learning model building cycle shown in Figure 4. Defining the problem was the first step in the framework. Relevant data was gathered, prepared, pre-processed, and transformed after the problem was established. Feature selection, which entails choosing algorithms and parameters that primarily aid in the output variable's prediction, is an additional component of the framework. The data was then divided into sets for testing, validation, and training. Following that, the techniques were assessed, and the best technique was selected using the outputs from each model.

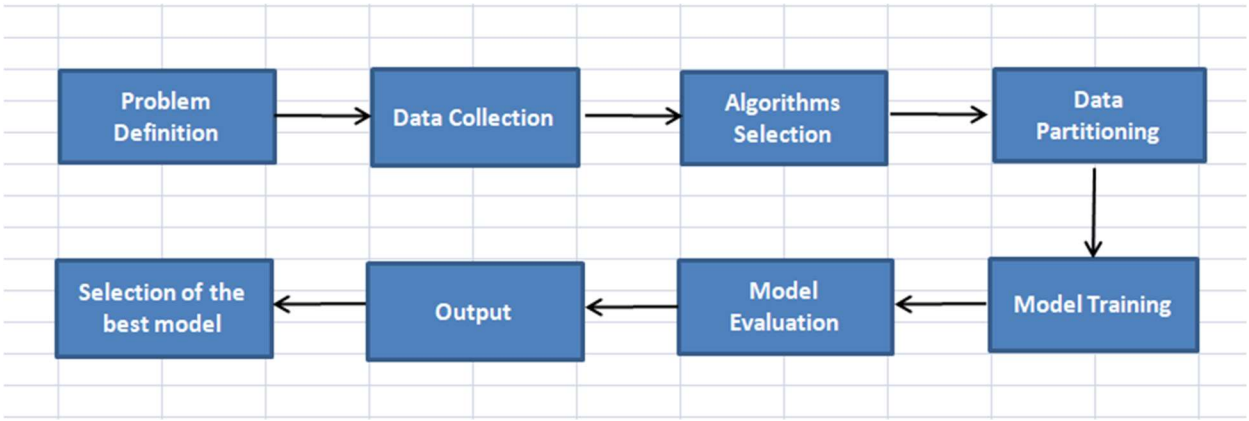


Figure 4: The research framework.

3.6 Research Design/strategy

The approach used in this study blended case study with design science research. Design science research strategy, according to Saunders *et al.* (2009), enhances research by assessing elements like models that address operations' problems. Case Study strategy was used along in this research because as it is an in – depth inquiry used along design research in order to establish rich knowledge about an aspect (Saunders *et al.,* 2019). A case –study was used to refine work and use data of one area inorder to understand and get more in-depth knowledge on the data for the area (Kombat Area).

3.7 Time Horizon

Due to time constraints of the study, the cross-sectional time horizon was applied as this is the one whereby research's data is collected at one point in time (Saunders *et al.,* 2019). This is true

for this research because all data utilised in this research was already gathered together and in the organisation's database and it was all collected at once.

3.8 Techniques and Tools

3.8.1 Data Source

The analytical/geochemical data of the area of study was obtained from the Earth Data Namibia (EDN) database, which is a comprehensive database of geological data, including mineral deposits, exploration and mining licenses, drilling data, geochemistry, maps, and reports. To store and manage this factual, geometrical and unstructured information, the database uses ORACLE and ARCVIEW as platforms. The principal source of geological data for Namibia is the Ministry of Mines and Energy's Geological Survey of Namibia (GSN), which is also in charge of maintaining this database. Professionals at GSN can use this secure server-hosted database, which provides data to interested clients: at no cost to students and at a price to private clients. The database is up to date (new data is added on a regular basis), accurate (gathered by trained geologists and subjected to stringent quality control procedures), and dependable (hosted on a secure server and routinely backed up). 3282 samples from the Kombat region that constituted of Cu, Pb and Zn were retrieved using extraction queries and saved to an Excel spread sheet.

3.8.2 Data Selection

The data was extracted from the Geochemistry Module in the EDN database which currently holds over 800 000 data points from mineral exploration, with assays for 32 elements. The EDN database user interface allows filtering of records based on the area the samples were collected from as shown below in Figure 5.

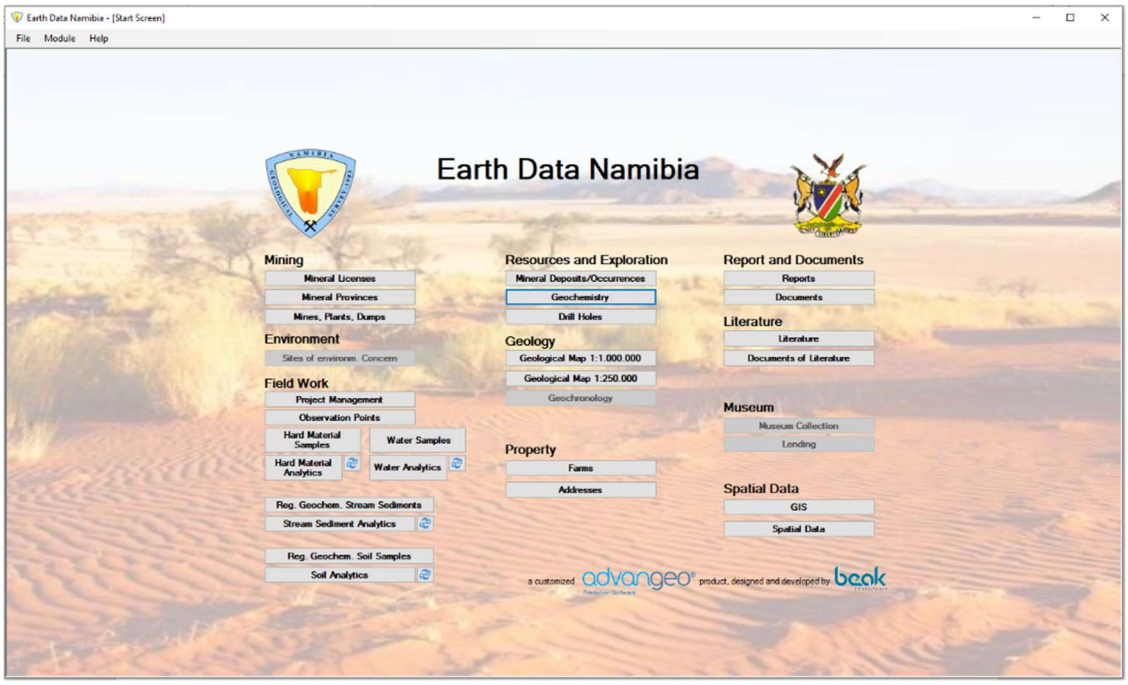


Figure 5: The EDN database interface.

The area of interest, Kombat area was selected by choosing mineral licences on a 1:250 000 map sheet. More search options were found under “Samples” (e.g. type – stream sediment/soil, etc.) and “Analytics” (specific elements – Cu, Pb, Au, etc.). The Kombat area was the study's area of interest, and analytics data related to Cu, Pb, and Zn were filtered out.

The database is built up with various tables that can be joined or split using the SQL commands. To view sampling parameters, such as depth and fraction, the highlighted data points are found under “Samples” tab. However, this information is not available for all data points. The analytical results are viewed under the “Samples – Analytics” or “Analytics” tabs. All samples in UTM33S which refers to the Universal Transverse Mercator projection, Zone 33, were collected in the southern hemisphere. UTM33S is a specific coordinate reference system used for mapping and navigation, irrespective of actual UTM Zone and in case of samples without individual Sample Numbers, the UTM coordinate is given as Sample ID. As shown in Figure 6 below, data was gathered by obtaining filtered tables from the database.

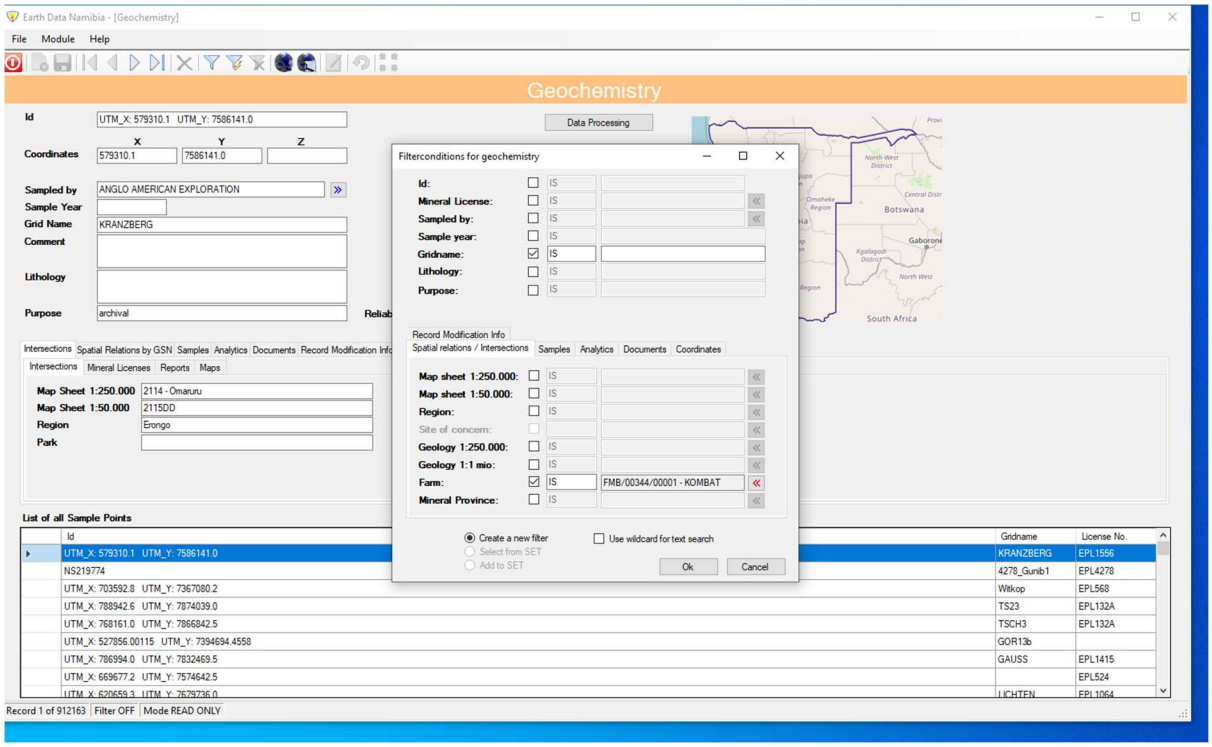


Figure 6: The EDN database interface tables and filters

3.8.3 Programming Environment

Jupyter as an interactive computing environment which was pre-installed in Anaconda was used for the research. The key component of Jupyter is the Jupyter Notebook, which is a web application that allows creation and interaction with notebooks containing executable codes, visualisations, and narrative text. Notebooks provide an excellent platform for data analysis, data visualisation, machine learning, and other computational tasks. One of Jupyter's primary benefits is its ability to show outputs—like graphs or tables—straight within the notebook, which facilitates comprehension and communication of the research's conclusions. Jupyter also encourages a narrative-style of programming, allowing explanations of the thought process and results in a coherent manner.

The name "Jupyter" comes from the combination of the three core programming languages it supports: Julia, Python, and R. Python programming language was used for this study for several reasons as outlined by Breiman (2001), it is:

1. Easy to use: Python is known for its easy-to-learn syntax, making it a popular choice among beginners and experienced programmers alike. Jupyter notebooks provide an interactive environment for exploring data and building models, making it easy to prototype and experiment with different models and techniques.
2. Large ecosystem: Python has many libraries and tools for data analysis, visualisation, and machine learning. It includes popular libraries like Scikit-learn, TensorFlow, Keras, and PyTorch. These libraries make it easy to build and train machine learning models with minimal code.
3. Tool integration: Python works well with other technologies and tools that are frequently used in data science and machine learning, including Hadoop, Spark, and SQL databases. Working with big datasets and scaling up models as needed is made simple by this.
4. Support from the community: Python boasts a sizable and vibrant developer community that produces and manages tools, documentation, and libraries for data science and machine learning. Both novice and seasoned developers can find a lot of information and assistance in this group.
5. Sharing and collaboration: Jupyter notebooks make it easy to share and collaborate on data analysis and machine learning projects. Notebooks can be shared via GitHub, Dropbox, or other cloud-based platforms, and they can be run and modified by other users with minimal setup.

3.8.4 Machine Learning Techniques

The following four machine learning algorithms were assessed: RF, KNN, DT and SVM. According to the literature review, these are the algorithms that have been shown to function well and are frequently employed in related work. The four machine learning algorithms are covered below:

1 Support Vector Machine (SVM): Applied to regression and classification applications, Support Vector Machine is a potent supervised learning technique. The aim is to identify an ideal hyperplane to divide data points into several classes. In order to create a decision boundary, it maximises the margin between each class' nearest data points, or support vectors. Using kernel functions, SVM can handle both linear and non-linear data

separation. It is renowned for its robust performance on small datasets and its capacity to handle high-dimensional data (Hastie, Tibshirani, & Friedman, 2009).

2 K-Nearest Neighbour (KNN): Ideal for classification and regression applications, K-Nearest Neighbour is a straightforward and easy-to-understand non-parametric technique. Common class or mean of the k-nearest data points in the feature space determines the class or value of an unknown data point in a KNN. The number of neighbours to take into account depends on the value of 'k'. KNN can perform well on small to medium-sized datasets and is easy to implement. However, for big datasets, it can be computationally costly and dependent on the distance measure selected (Hastie, Tibshirani, & Friedman, 2009).

3 Decision Tree: Raschka and Mirjalili (2017) state that a popular supervised learning technique for tasks involving regression and classification is the decision tree. It generates a model that looks like a tree, whereby each node is standing for a feature, each branch for a choice, and leaf node for a numerical value or a class label. Decision trees are helpful for understanding the decision-making process since they are simple to understand and depict. They may, however, experience overfitting, particularly with intricate datasets.

4 Random Forest: Several trees are combined in RF, an ensemble learning technique, to increase model accuracy and decrease overfitting. By training the decision trees on various random subsets of the data and features, it generates a diversified collection of decision trees. The total of all the individual trees' projections is used to get the final forecast. When compared to a single Decision Tree, Random Forest is more reliable, accurate, and resistant to overfitting. Large and high-dimensional datasets yield good performance (Breiman, 2001).

3.8.5 Evaluation Metrics

Six indicators were used to assess the four machine models' performances in order to compare and determine how effective each machine learning model was. To provide a more thorough assessment of the models, performance comparisons using several metrics (Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-Squared (Coefficient of Determination), Mean

Absolute Error (MAE), Adjusted R-Squared, and Explained Variance Score) were carried out. According to Raschka and Mirjalili (2017), the six measures are explained as follows:

1. Mean Squared Error (MSE): In a regression problem, average squared difference between the actual and predicted values is measured using this common metric. Because of the squaring process, it significantly penalizes large errors and is hence susceptible to outliers. MSE = (1/n) * Σ(actual - predicted)^2
2. Another often used statistic for assessing how well regression models work is the Root Mean Squared Error (RMSE). It is closely connected to Mean Squared Error (MSE). Primary distinction is that the metric produced by RMSE is more easily interpreted and comprehended because it takes the square root of MSE and stores result in same unit as the target variable. RMSE = RMSE = √ (MSE)

   Since RMSE expresses the mean magnitude of errors in the same units as the target variable, it is frequently chosen over MSE. This makes it simpler to understand the model's performance in a real-world setting because RMSE will also be in parts per million (ppm) for the target variable.
3. R-Squared: Percentage of the variance in the dependent variable that can be predicted from the independent variables (features) is represented by a statistical metric known as R-squared. It gauges how well the data variance is clarified by the model. R-squared values fall between 0 and 1, where 1 is a perfect fit and 0 shows no variation in the model's explanation. R-squared = 1-(SS_residual / SS_total), where SS_total denotes the total sum of squares and SS_residual represents the sum of squared residuals.
4. Mean Absolute Error (MAE): This additional statistic is employed in regression jobs. In contrast to MSE, it computes the mean absolute variations between the actual and predicted values, which makes it less susceptible to outliers. MAE = (1/n) * Σ|actual - predicted|
5. Adjusted R-Squared: An R-Squared measure adjusted to account for number of independent variables in the model. It penalizes the addition of unneeded variables to the model, allowing for a more accurate assessment of the model's effectiveness when taking the total amount of features into account. Adjusted R-squared = 1 - [(1 - R^2) * (n

- 1) / (n - p - 1)], 'p' denotes independent variables and 'n' denotes the number of data points.

6. Explained Variance Score: This metric computes percentage of the dependent variable's volatility that the model can work with. It is frequently used to evaluate the best fit of the model in regression tasks. Perfect fit is indicated by a score of 1, which runs from 0 to 1. Explained Variance Score = 1 - (Var (actual - predicted) / Var (actual).

Table 2: Summary of evaluation metrics.

Metrics	**Description**	**Desired value**
MSE	Calculates mean squared difference between the predicted and actual values.	Low
RMSE	Square root of MSE	Low
R-Squared	Evaluates the degree to which the data variance is explained by the model.	High
Adjusted R-Squared	Adjusted version of the R-Squared metric.	Lower
MAE	Determines the mean absolute variations between the predicted and actual values.	Low
Explained Variance	Percentage of the dependent variable's variation that the model can account for.	High

3.9 Data Analysis

Python's libraries: Pandas, NumPy, Matplotlib, and ScikitLearn were used for exploratory data analysis, data cleaning, and visualisation as well as to build the models.

3.10 Summary

The Chapter focused on the methodology and the motivation as to why such methodologies were applicable for this study. Table 3 below summarissummarises the methodologies used to address the questions of the research. The chapter included information regarding the data as well as a discussion of the procedures used during data collection. The findings and discussions are presented in the following chapter: Chapter 4.

Table 2: Study methodology summary.

Research question	**Research activity**	**Evaluation criteria**
i)What are the patterns depicted from the metal concentrations in the different copper deposits?	Exploratory Data Analysis using python.	Observe the patterns and trends exhibited by the data.
ii)How well do the different machine learning models predict metal concentration?	Narrow the focus down to four commonly used techniques (RF, KNN, SVM and DT)	MSE, RMSE, MAE, R squared, adjusted R-squared, explained variance metrics noted for all the four algorithms.
iii)Which machine learning technique is best suited for the metal concentration prediction in copper deposits based on performance?	Comparison and ranking of the six metrics for the four algorithms.	Metrics values for all algorithms noted MSE (desirable less value) RMSE (desirable less value) MAE (desirable less value) R-squared(desirable high value) Adjusted R-squared(desirable lower value) Explained variance(desirable high value)

CHAPTER 4: RESULTS AND EVALUATION

4.1 Introduction

The amount and kind of data gathered for this research are discussed in this Chapter. Outcomes of the analysis and effectiveness of selected machine learning techniques are also covered in this Chapter.

4.2 Analysis Tools

Different libraries in Python as presented in Figure 7 were used for different purposes such as data pre-processing, preparation, models packages, plots generation and metrics.

```
In [ ]: import pandas as pd
        #Data pre-processing,preparation
        from sklearn.preprocessing import StandardScaler
        from sklearn.preprocessing import OneHotEncoder
        from sklearn.model_selection import train_test_split
        from sklearn.compose import ColumnTransformer
        from sklearn.pipeline import Pipeline
        from sklearn.impute import SimpleImputer
        #Models
        from sklearn.tree import DecisionTreeRegressor
        from sklearn.ensemble import RandomForestClassifier
        from sklearn.neighbors import KNeighborsRegressor
        from sklearn.svm import SVR
        #Plots
        import matplotlib.pyplot as plt
        # Metrics
        from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error, explained_variance_score
        #Data
        df = pd.read_csv("C:/Users/LYDIA/Downloads/Analytical_Dataset.csv")
```

Figure 7: The Python packages used.

4.2 Exploratory Data Analysis (EDA)

4.2.1 Data Shape

The dataset is composed of 3282 records with eight useful fields for this study. Appendix B shows the statement snippet and output of the data shape.

4.2.2 Data fields and tpyes

Table 4 below lists the eight relevant fields from the dataset.

Table 3: Fields in the dataset.

Field	Description	Data Type	Example in the dataset (Row 1)	Unit
Sample Number	A unique identifier for the sample.	Numerical	1	-
Northing	northing geographic coordinate in UTM.	Numerical	786553.1	-
Southing	southing geographic coordinate in UTM.	Numerical	7816087.5	-
Grid Name	The location where the sample was collected.	Categorical	OTASLIME	-
Type	Type of the sample	Categorical	soil	-
Zn	Concentration of zinc in the sample	Numerical	174	ppm
Cu	Concentration of copper in the sample	Numerical	39	ppm
Pb	Concentration of lead in the sample	Numerical	89.0	ppm

The data summary and types of data contained in different fields was queried with the df.info () Python statement as shown in Figure 8 below.

```
In [7]: df.info()
        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 3282 entries, 0 to 3281
        Data columns (total 8 columns):
         #   Column           Non-Null Count  Dtype
        ---  ------           --------------  -----
         0   Sample Number    3282 non-null   int64
         1   Northing         3282 non-null   float64
         2   Southing         3282 non-null   float64
         3   Grid Name        3282 non-null   object
         4   Type             3282 non-null   object
         5   Zn (ppm or g/t)  3282 non-null   int64
         6   Cu (ppm or g/t)  3282 non-null   int64
         7   Pb (ppm or g/t)  3281 non-null   float64
        dtypes: float64(3), int64(3), object(2)
        memory usage: 205.2+ KB
```

Figure 8: The data summary and types code.

A quick inspection of the data in order to get an initial sense of its structure was done by querying the first 5 rows of the dataset the df.head (5) statement. Working with a large dataset of this research, printing the entire DataFrame is impractical. Using head() allowed a quick check on a small portion of the data for debugging purposes. This allows a rapid glimpse of titles of columns, the sorts of data they include, and their contents.

This provides a preview of the dataset, helping to verify that the data has been loaded correctly and that there are no obvious issues or inconsistencies. Furthermore, by looking at the first few rows, patterns or trends are identified, missing values are checked, and preliminary observations about the dataset are made.

The first 5 rows of the dataset were queried as shown in Figure 9 below.

```
In [8]: df.head(5)
Out[8]:
```

	Sample Number	Northing	Southing	Grid Name	Type	Zn (ppm or g/t)	Cu (ppm or g/t)	Pb (ppm or g/t)
0	1	786553.1	7816087.5	OTASLIME	soil	174	39	89.0
1	2	786042.8	7816414.5	OTASLIME	soil	88	83	67.0
2	3	783969.3	7816937.5	OTAASISK	soil	70	39	49.0
3	4	783992.8	7816981.5	OTAASISK	soil	131	56	40.0
4	5	784068.8	7816487.0	OTAASISK	soil	56	303	43.0

Figure 9: The first 5 rows of the dataset.

4.3 Data Preparation

To further prepare the data for the models, the following steps were undertaken.

4.3.1 Missing values

Treating missing values in a dataset involves tasks such as removing the rows with missing data or imputation (replacing missing values with estimated values). Handling missing data is an important step for various benefits. Firstly it helps with accurate analysis. Missing values in datasets can lead to inaccurate and biased analyses. By removing rows with missing values, the data used for analysis is complete and reflects the actual observations. Secondly, treating missing data helps with improving the model performance. When using a DataFrame for machine learning, the performance of the model can be adversely affected by missing values. Cleaning the

data is crucial before putting it into the model because some machine learning algorithms might not be able to handle missing data. Thirdly, handling missing data results in good visualisation. Missing values can create gaps in visualisations, which can be misleading and affect the interpretation of data patterns. By dropping missing rows, more accurate visualisations are created. Lastly, treated missing values is important because some operations may produce errors or unexpected results when missing values are present. By handling missing data appropriately, potential errors are avoided and smooth data processing is guaranteed.

Removing the rows with missing data was considered more appropriate because imputation was going to give unrealistic metal concentarations and affect the prediction results. Furthermore, the missing values were distributed across a smaller portion of the dataset and it was not going to lead to significant data loss.

The '.dropna()' method in figure 10 below was used to drop rows with missing values. By default, the query removes any row that contains at least one NaN or None value.

```
In [3]: # Drop rows with missing values
        df.dropna(inplace=True)
```

Figure 10: Dropping of the missing values' code.

The 'inplace=True' argument is optional. When inplace=True, it means the DataFrame (df) will be modified in place, and the operation will not create a new DataFrame. Instead, it will update the existing DataFrame directly. If this argument is ommited or set to False, the operation will return a new DataFrame with the missing rows dropped, leaving the original DataFrame unchanged.

4.3.2 Feature Importance

Removing irrelevant or redundant columns from the dataset is a common data preprocessing step that can simplify the data, improve analysis efficiency, and enhance model performance. There were columns in the dataset that have nothing to do with the prediction of metal content. Eliminating these superfluous columns can streamline the information and facilitate concentration on the crucial fields. Retaining irrelevant information can also cause problems with multicollinearity in statistical models and increase the computational overhead of machine

learning techniques. Furthermore, eliminating unnecessary or duplicate columns might aid in lowering the dataset's dimensionality, which can be essential for productive and successful analysis—particularly when working with sizable datasets. Additionally, processing and analysing data can be sped up by working with a smaller selection of columns, particularly when working with huge datasets. Finally, characteristics that are noisy or unnecessary might have a negative effect on model performance in machine learning. Removing these features can lead to better models and improved predictive accuracy. Analysis on feature importance was done to determine most important features for predicting Cu metal concentration. Table 5 below shows the calculated feature weighting which served as criteria for determining irrelevant features not required for the predictions.

Table 5: Feature correlation matrix.

	Sample Number	Northing	Southing	Zn (ppm or g/t)	Cu (ppm or g/t)	Pb (ppm or g/t)
Sample Number	1.000000	0.000846	-0.014349	0.012728	0.020999	0.014300
Northing	0.000846	1.000000	-0.151469	-0.126888	0.076550	0.069118
Southing	-0.014349	-0.151469	1.000000	-0.294143	-0.049233	-0.042269
Zn (ppm or g/t)	0.012728	-0.126888	-0.294143	1.000000	0.279575	0.219536
Cu (ppm or g/t)	0.020999	0.076550	-0.049233	0.279575	1.000000	0.759836
Pb (ppm or g/t)	0.014300	0.069118	-0.042269	0.219536	0.759836	1.000000

The features below the weighting of 0.20 were considered non-important. In figure 11 below, the '.drop()' method was used to remove columns from the DataFrame (df). The first argument of the drop() function is a list of column names to be dropped. ['Sample Number', 'Northing', 'Southing', 'Grid Name', 'Type']: This is the list of column names that were removed from the DataFrame. 'axis=1' is an argument that indicates that its columns to be dropped (vertical scaling) and not rows. The value 1 signifies columns, while 0 would represent rows. The 'data =' assigns the modified DataFrame with the specified columns removed to the variable data and the original DataFrame df remains unchanged.

```
In [5]: # Remove columns not to use
        data = df.drop(['Sample Number', 'Northing', 'Southing', 'Grid Name', 'Type'], axis=1)
```

Figure 11: Dropping of columns not used for prediction.

4.3.3 Splitting data input feature and target variables

In supervised machine learning, dividing the dataset into input features (X) and target (y) is an essential step. Effective model construction and evaluation are made possible, and data leaking is prevented and code modularity is preserved. In this way, the potential of supervised learning algorithms is fully utilised to provide forecasts and derive significant understanding from the information. It is made apparent what the model is to predict (target) based on the information provided (features) by dividing data into features and target variables; in other words, the data is labeled with known input-output pairings. Furthermore, dividing the data into input and output variables aids in preventing data leakage during model training. When data from the target variable is unintentionally used during training, it is known as data leakage and causes an overestimation of the model's performance. Moreover, the code becomes modular and scalable when the features and target are separated. It is possible to modify the features or the target variable without changing the entire source. With distinct X and Y, it is possible to provide the right input features for prediction once a model has been trained and applied to fresh, untested data. Finally, when performing feature engineering or feature selection, having the features and target variable separated makes it easier to identify which features are most relevant for predicting the target.

The dataset was split into two separate data structures X (features) and y (target variable) with the code given in figure 12 below.

```
In [4]: # Split the dataset into features (Zn and Pb) and the target variable (Cu)
        X = df[['Zn (ppm or g/t)', 'Pb (ppm or g/t)']]
        y = df['Cu (ppm or g/t)']
```

Figure 12: Splitting of data into features and target variables.

X = df[['Zn (ppm or g/t)', 'Pb (ppm or g/t)']]: Here, X is assigned the value of a DataFrame containing the columns 'Zn (ppm or g/t)' and 'Pb (ppm or g/t)' from the original DataFrame (df). These columns represent the features or inputs of the dataset.

y = df['Cu (ppm or g/t)']: Here, y is assigned the value of a Series containing the column 'Cu (ppm or g/t)' from the original DataFrame (df). This column represents the target variable or the output to be predicted using the features.

After this splitting, X will contain the feature columns, and y will contain the target column.

The objective is to predict the copper concentration using zinc and lead concentrations as inputs.

4.3.4 Splitting into Training and Testing Datasets:

In evaluating the performance of machine learning models on new, unseen data, it is essential to partition the dataset into distinct subsets for training, validation, and testing. This study followed a standard practice, where the dataset of 3282 samples was initially divided into training and testing sets.

The training set, comprising 80% of the original data, was utilised to train the machine learning models. During this phase, the models learned patterns and relationships between input attributes (X) and the target variable (y).

Conversely, the remaining 20% of the dataset, designated as the testing set, was held out from the training process. This subset served as an independent benchmark to assess the models' performance on fresh, unseen data. The models were applied to this test set to generate predictions, and their generalisability was evaluated by comparing these predictions with the actual outcomes. In this study, the focus was primarily on the training and testing sets for model evaluation.

4.4 Data Visualisation

The graphical depiction of the data is known as data visualisation. It entails producing graphic components to illustrate patterns, trends, and insights in the data, such as graphs, charts, plots, and maps. The Matplotlib software was utilised in this study to visualise the data. Matplotlib is versatile, powerful and widely used Python library to provide a wide range of plotting functions to create various types of charts and graphs. For this study, visualisation was accomplished through graphs such as scatter plot, histogram plot, box plot, distribution plot; pairwise scatter

plot, and the correlation heatmap. Each of these visualisations serves different purposes and helps in gaining insights from data in various ways. The codes used for these plots generation are presented in Appendix A.

Relationship between Cu, Zn, and Pb concentrations - Scatter plot

A scatter plot is a two-dimensional data visualisation that represents individual data points as dots. It is useful for visualising the relationship between two variables whereby each dot on the scatter plot represents a single data point with its x and y values.

A scatter plot in figure 13 below was plotted to visualise the relationship between Cu, Zn, and Pb concentrations and help observe any potential correlations or patterns between these variables. The plot shows that most of the concentrations are below 500ppm. The scatter plot further depicts that for high Copper (Cu) concentrations the Lead (Pb) concentrations are in the same ranges whereas the Zinc (Zn) concentrations are much lower.

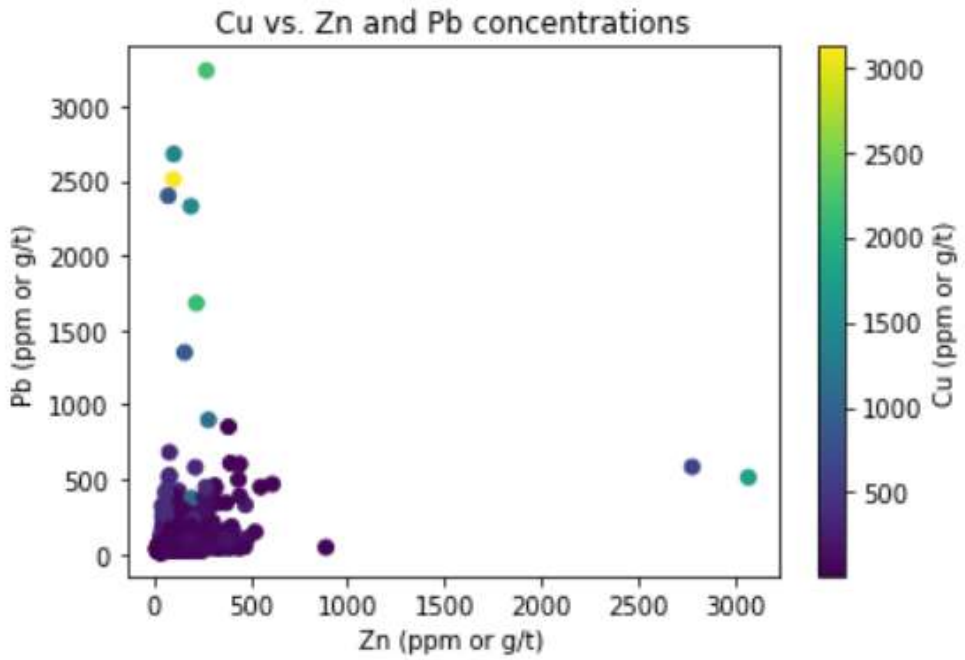


Figure 13: Scatter plot showing relationship between Cu, Zn and Pb concentration.

Representation of the distribution of variables - Histogram

A single variable's distribution is shown graphically via a histogram. It aids in determining the data's central tendency, dispersion, and skewness and shows the frequency or count of data points that fall into certain bins or intervals.

Figure 14 was plotted to visualise the distributions of Cu, Zn, and Pb concentrations individually and help with understanding the range and distribution of each variable. The histogram shows that within the limit of 500 ppm, most Pb and Cu concentrations are in the lower ranges whereas some of the Zn concentrations proceed further toward 500ppm.

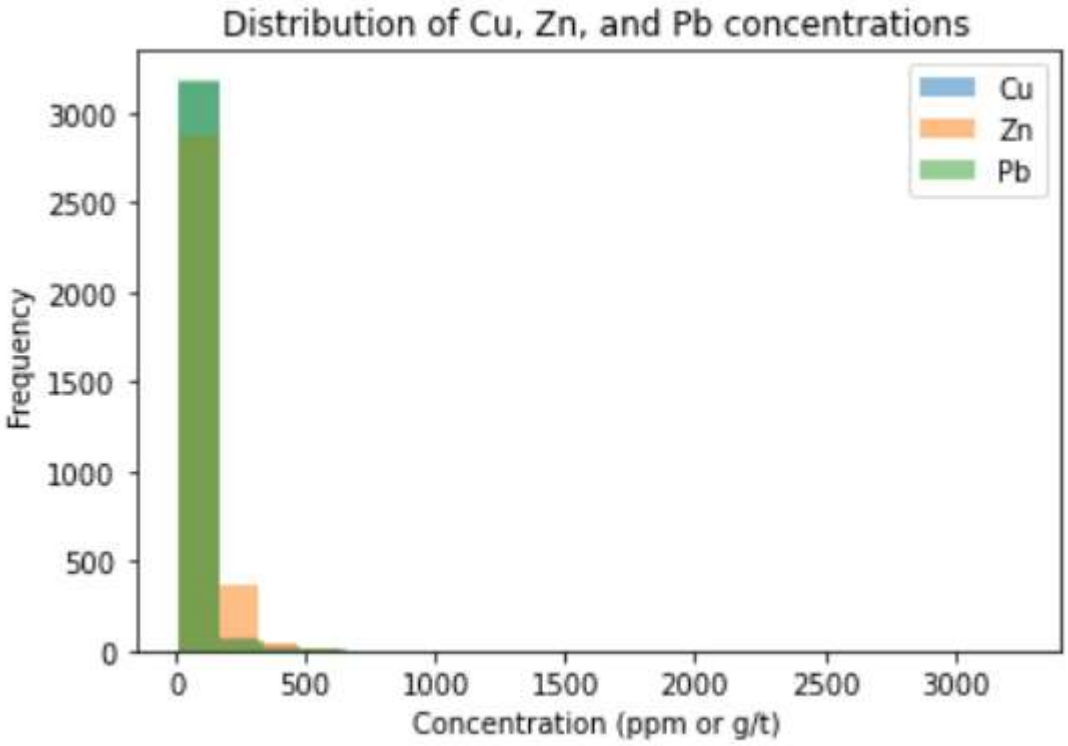


Figure 14: Histogram representation of the distribution of variables of the dataset.

Distribution of metal across the area of study - Box plot

A box plot uses quartiles to show how a dataset is distributed. The median is shown as a line inside the box, which symbolizes the interquartile range. The points extend from the edges of the box to show the range of the data within a certain distance from the quartiles. These plots are useful for identifying outliers and comparing the distribution of multiple groups or variables.

A box plot in figure 15 was plotted to visualise the distribution of Cu concentrations across different categories such as Grid Name. This help with identifying any variations or outliers in Cu concentrations based on these categories. The box plot below shows that high Cu concentrations are more common at Otasline, Otagross and Otainsel. Lower Cu concentrations are more common at Oaasisk, RL, Otaschn, Otastr and RP.

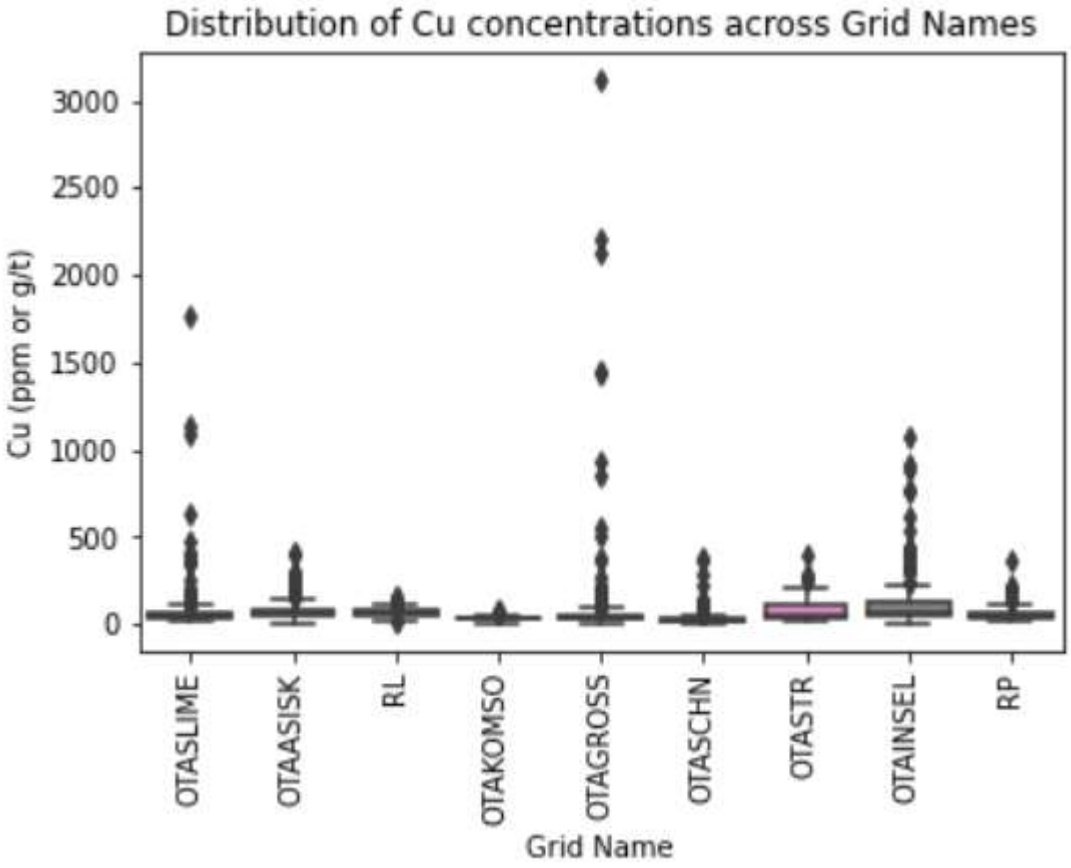


Figure 15: Box plot of the dataset showing distribution of metal across the area of study.

Distributions of Cu, Zn, and Pb concentrations -Distribution plot

A distribution plot provides a smooth estimate of the probability density function of variables. It represents overall distribution of the data as a continuous curve. The distribution plot in Figure 16 below creates histograms and density plots to visualise the distributions of Cu, Zn, and Pb concentrations individually to help understand the spread and shape of each variable. The

distribution plot below shows that the underlying probability densities of the data of the three metals are in line.

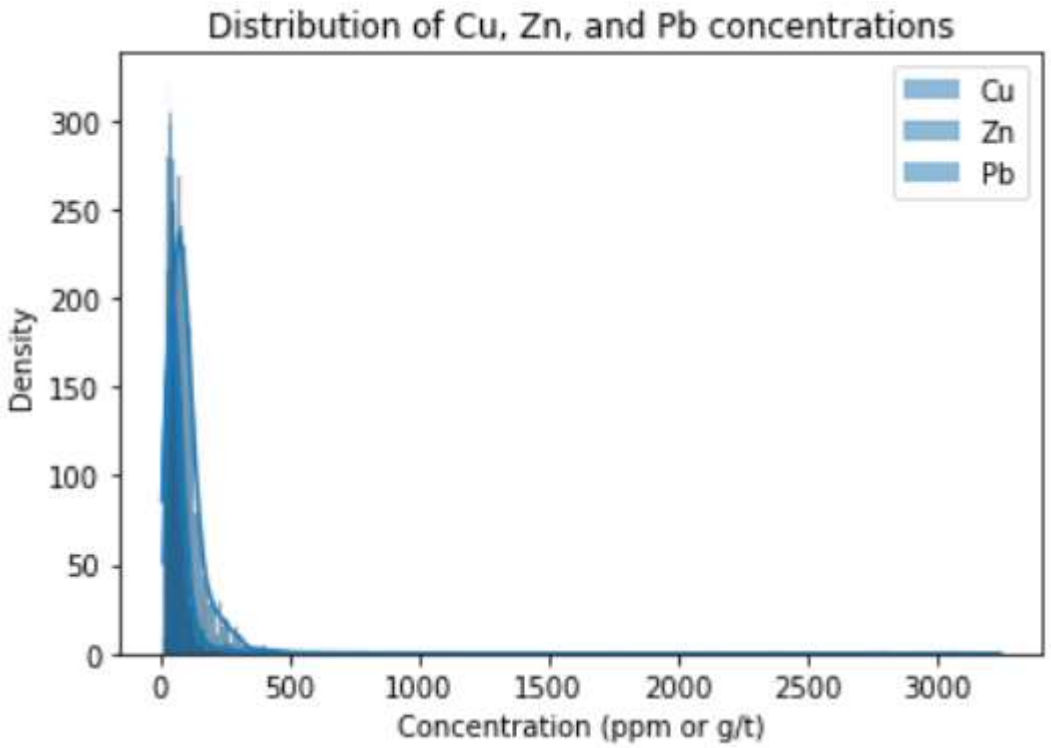


Figure 16: Histogram distributions of Cu, Zn and Pb concentrations of the dataset.

Pairwise relationships between concentrations - scatter plot

A grid of scatter plots, each of which shows the association between two variables from the dataset, is called a pairwise scatter plot. It makes it possible to concurrently visualise the pairwise relationships between several variables. Figure 17 is a pairwise scatter plot which creates a matrix of scatter plots to visualise the relationships between Cu, Zn, and Pb concentrations and help identification of any correlations or patterns among the three variables. The plot depicts an indirect proportional relationship between Cu and Pb, whereby for higher Cu concentrations, the corresponding Pb concentrations are lower. However, there is a direct proportional relationship between Cu and Zn as the values for the two metals increase proportionally.

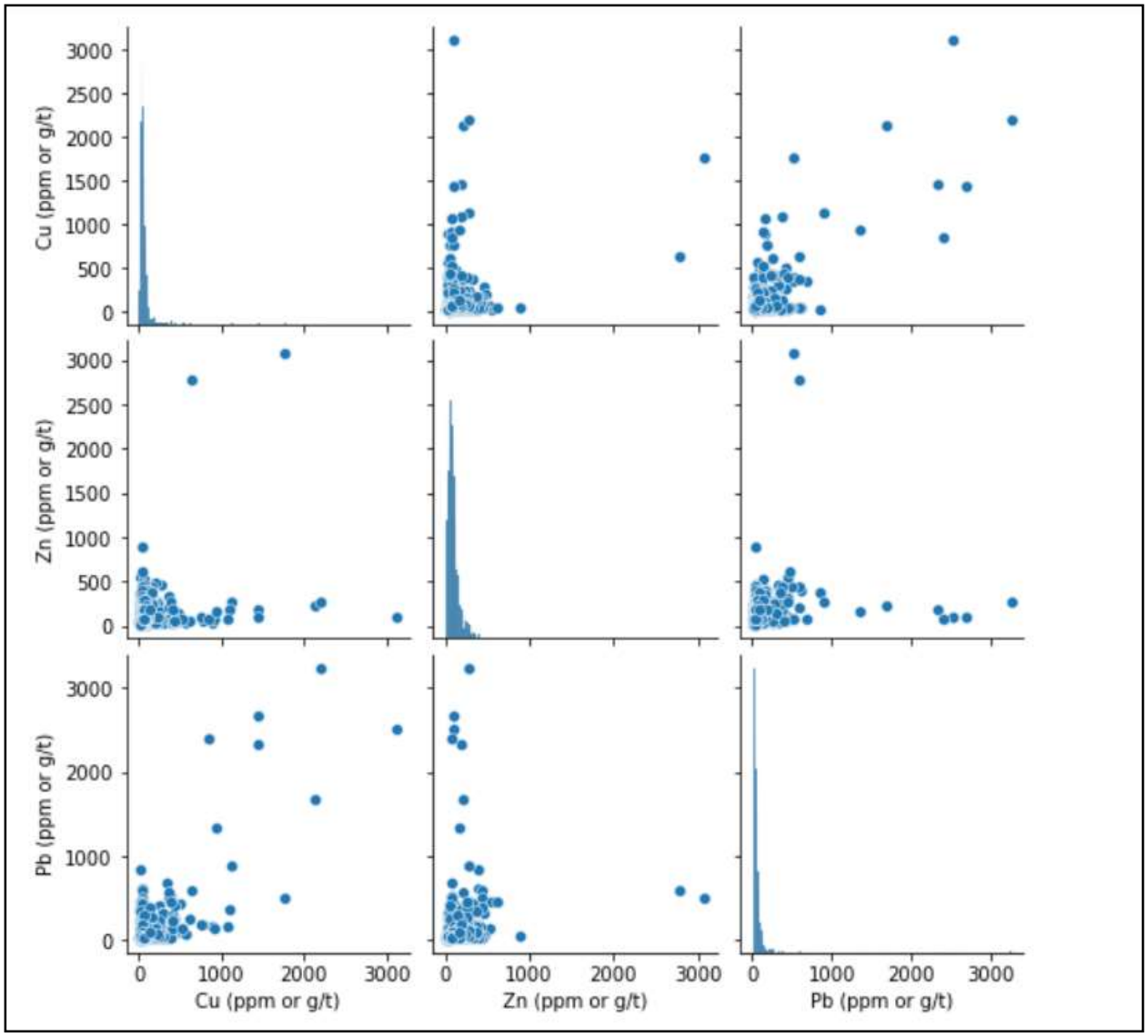


Figure 17: Pairwise relationships between concentrations of the dataset

Pairwise correlations between Cu, Zn, and Pb concentrations - Heatmap

The correlation heatmap presents a color-coded matrix that shows the relationships between several variables in a dataset. Magnitude and direction of the correlation of sets of variables are shown by colours in the plot. To spot multicollinearity and correlation patterns between variables, correlation heatmaps are helpful.

A correlation heatmap illustrating the pairwise correlations between Cu, Zn, and Pb concentrations is displayed in Figure 4.12 below. This figure aids in determining the direction and intensity of the interactions between the variables. Figure 18 further shows that there is a strong

correlation between Cu and Pb than that between Cu and Zn as presented by the values 0.76 for Cu and Pb and 0.28 for Cu and Zn. The plot further shows that Zn is more correlated to Cu than it is to Pb as presented by the numbers 0.28 and 0.22. These low values for Zn, also mean that Zn is poorly correlated to the other metals. Furthermore, for Pb the correlation is that it is strongly correlated to Cu than it is to Zn as shown by the values of 0.76 and 0.22 respectively.

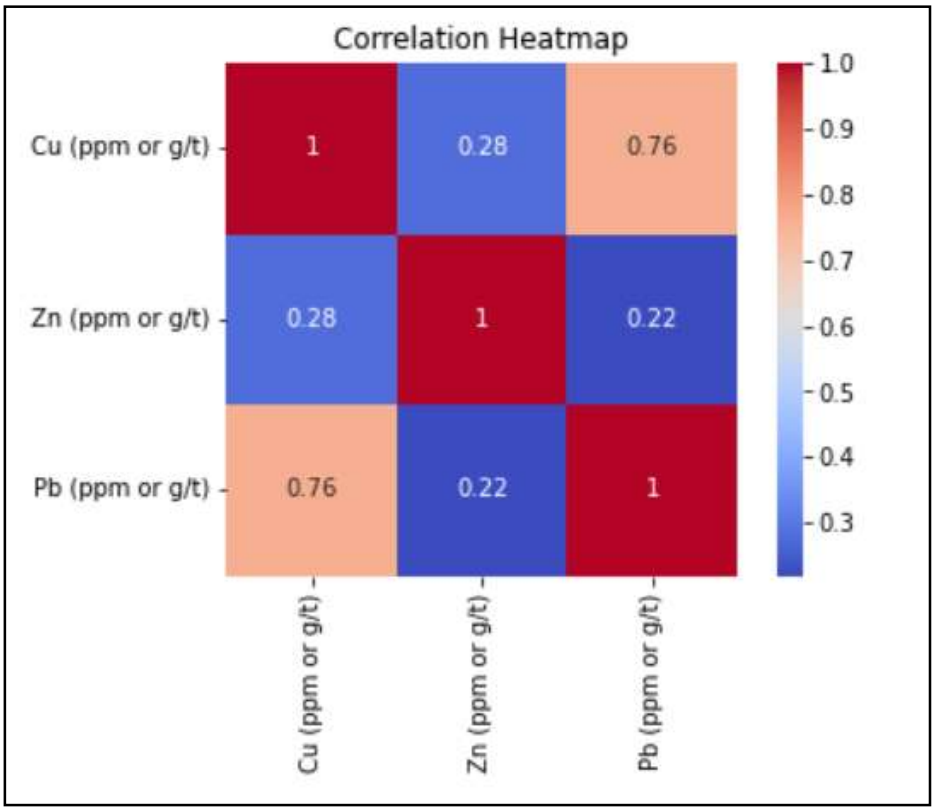


Figure 18: A heat map of pairwise correlations between Cu, Zn and Pb concentrations of the dataset.

4.5 Training, Testing and Validation of the Data

The development and assessment of machine learning models need completion of training, testing, and validation procedures. They are essential in making sure that the models work correctly in various contexts and that they generalise well to fresh, untested data.

A portion of labeled data is used during training phase to instruct the machine learning model how to create predictions. Through the use of optimisation techniques, the model adjusts its internal parameters as it discovers underlying patterns and relationships in training data. During validation phase, model's hyperparameters are adjusted and its performance is evaluated during training using a different subset of the dataset, referred to as the validation set. The learning

process is influenced by settings called hyperparameters (e.g., learning rate, regularisation intensity). Hyperparameters are adjusted to enhance the performance without utilising test set by assessing the model on validation set (preventing overfitting to the test set). After the model is trained and validated, it is critical to assess its performance with brand-new, untested data. During the testing step, the model is evaluated for generalisation ability using a test set that the model has never seen before. Testing helps identify problems like overfitting and gives an estimate of how well the model is expected to perform in comparison to other datasets.

Figure 19 below shows validation, testing, and training statement. The code is broken down step-by-step below:

```
In [5]: # Split the data into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figure 19: Data splitting, training, and testing code.

xVars, yVars, test_size=0.2, random_state=42) = train_test_split(xTrain, xValid, yTrain, yValid): This line divides the data into training and validation sets using train_test_split function from scikit-learn library.

The input features and target variable are passed as xVars and yVars, respectively.

test_size=0.2 designates that 80% of data will be utilised for training and 20% for validation.

random_state=42. To guarantee that the data split is repeatable, random_state=42 sets a precise random seed.

There are four data sets that the train_test_split method returns:

The training set of input features is represented by xTrain.

xValid: Represents the validation set of input characteristics.

yTrain: The target variable's training set.

yValid: The target variable's validation set.

With the use of these splits, models may be trained using the xTrain and yTrain data, and their performance can then be assessed using the xValid and yValid data. To ensure consistency in the

data split every time the code runs, the random_state parameter is set. This is helpful for comparing model performance and reproducibility.

4.6 Scaling of the features

The characteristics (Zn and Pb) were scaled to contain zero mean and unit variance using StandardScaler. This preprocessing stage aids in guaranteeing that each feature makes an equal contribution to the model. Using StandardScaler to scale feature variables enhances the stability and performance of machine learning algorithms. By guaranteeing that every feature is on the same scale, it avoids any unwarranted influence that would result from variations in feature ranges.

The scaler in Figure 4.15 below was initially made using the function "StandardScaler()," which produces an instance of the StandardScaler class. This instance was then used to scale the features.

The scaling transformation was applied to the training feature data (X_train) by using the formula "X_train = scaler.fit_transform(X_train)" to fit and transform training data. After determining average and standard deviation of each feature in training set, the fit_transform() method scales the features according to these figures. This guarantees that each feature in the scaled data has an average of 0 and a standard deviation of 1.

The code 'X_test = scaler.transform(X_test)' in Figure 20 below applied the same scaling transformation to the test feature data (X_test) in order to change the test data. However, it makes use of the values computed from the training data rather than creating new mean and standard deviation values. By doing this, the consistency of scaling between the test and training data is guaranteed.

```
In [6]: # Scale the numerical features
        scaler = StandardScaler()
        X_train_scaled = scaler.fit_transform(X_train)
        X_test_scaled = scaler.transform(X_test)
```

Figure 20: Transforming and scaling the data.

4.7 Models

The four machine learning techniques (KNN, SVM, DT, RF) that were covered in chapter three were tested following the completion of steps 4.2 through 4.6.

4.7.1 K-NN Model

For the K-NN model, the libraries in Appendix C were imported. The K-NN regressor model was created, training of the model and predicting of the target variables were done as per the labelled codes in the same Appendix.

The six model assessment metrics were created using the code in Figure 21 below. This was in order to assess the performance of the model.

```
In [10]: # Calculate evaluation metrics
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
rmse = mse ** 0.5
mae = mean_absolute_error(y_test, y_pred)
n = X_test.shape[0]  # Number of samples in the test set
p = X_test.shape[1]  # Number of independent variables (features)
adj_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
adj_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
explained_variance = explained_variance_score(y_test, y_pred)
```

```
In [11]: # Print the evaluation metrics
print('Mean Squared Error:', mse)
print('R-squared Score:', r2)
print('Root Mean Squared Error (RMSE):', rmse)
print('Mean Absolute Error (MAE):', mae)
print("Adjusted R-squared:", adj_r2)
print('Explained Variance Score:', explained_variance)
```

```
Mean Squared Error: 10866.860761035008
R-squared Score: 0.5692537942141789
Root Mean Squared Error (RMSE): 104.24423610461639
Mean Absolute Error (MAE): 32.82557077625571
Adjusted R-squared: 0.5679365275298185
Explained Variance Score: 0.5728390840225923
```

Figure 21: K-NN Model evaluation metrics.

The mean squared difference between the expected and actual numbers is measured by the Mean Squared Error, or MSE. Better performance is indicated by a lower MSE. The MSE score of 10866.860761035008 in this instance indicates that there is a significant squared difference between the expected and actual numbers on average.

R-squared Score: This figure indicates the percentage of target's variance that can be predicted from features. On a scale of 0 to 1, 1 denotes an ideal fit. The R-squared value of 0.5692537942141789 indicates that the features in the K-NN model account for roughly 56.93% of the variance in the target variable.

Root Mean Squared Error (RMSE): The statistic indicates mean magnitude of prediction error. Better performance is indicated by lower RMSE values. The RMSE value of 104.24423610461639 in this instance indicates that the average error of the forecasts is roughly 104.24 units.

Mean absolute difference between expected and actual values is measured by Mean Absolute Error (MAE). Without squaring the errors, it offers an interpretation that is comparable to that of the RMSE. The MAE number in this instance, 32.82557077625571, indicates that the average absolute error of the forecasts is roughly 32.83 units.

Adjusted R-squared: This statistic penalizes the inclusion of unimportant variables and takes the number of predictors in the model into consideration. When comparing models with varying numbers of predictors, it is helpful. Although the adjusted R-squared value of 0.5679365275298185 is lower than the R-squared score, it suggests that adding more characteristics to the model could not have a major positive impact.

Explained Variance Score: This score indicates the percentage of target variable's variance that the model can account for. On a scale of 0 to 1, 1 denotes an ideal fit. With an explained variance score of 0.5728390840225923, the K-NN model appears to account for roughly 57.28% of the variance in the target variable.

Plotting of predictions vs. actual

The bar graph in Figure 22 demonstrates a positive trend between the actual and expected values.

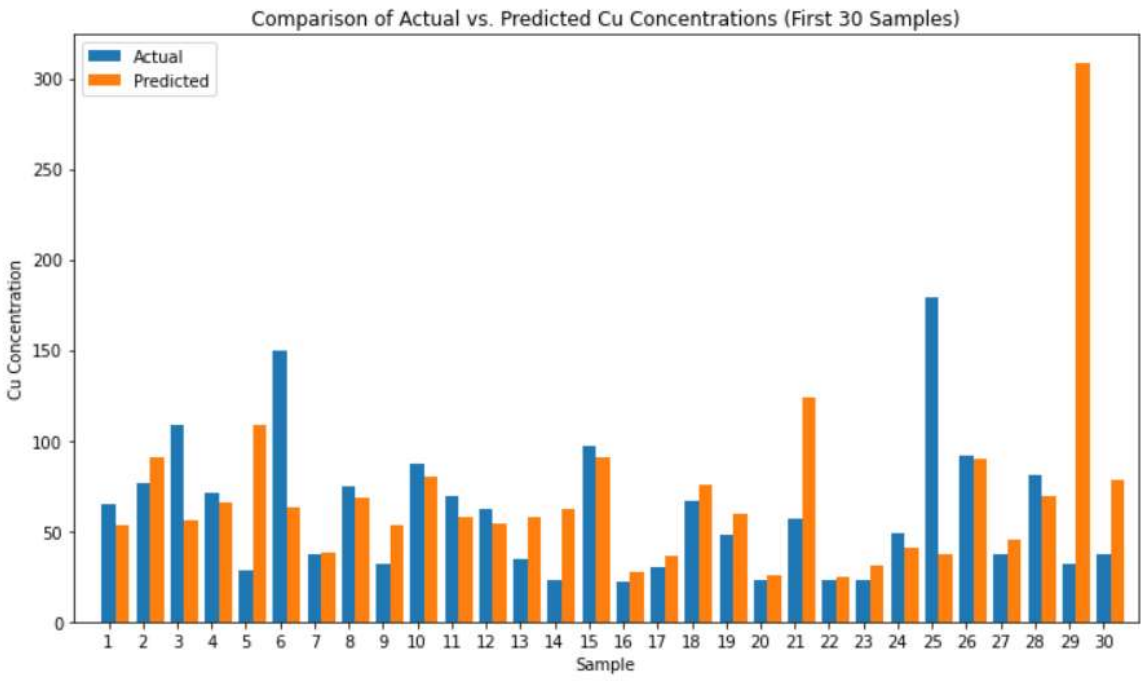


Figure 22: Comparison of the predicted and actual Cu concentration in the K-NN model.

The ideal prediction line, shown in Figure 23 below, is represented by the red line on the plot and indicates where the dots would fall if the predictions and actual values were perfectly equal. It is possible to observe how closely the projected values match the actual values by comparing the points to the perfect prediction line, or target. The map shows that there are a lot of points dispersed tightly about the ideal prediction line, indicating that the actual values and predictions coincide well. There are, however, additional points that are widely dispersed, indicating a greater variation between predicted and actual values.

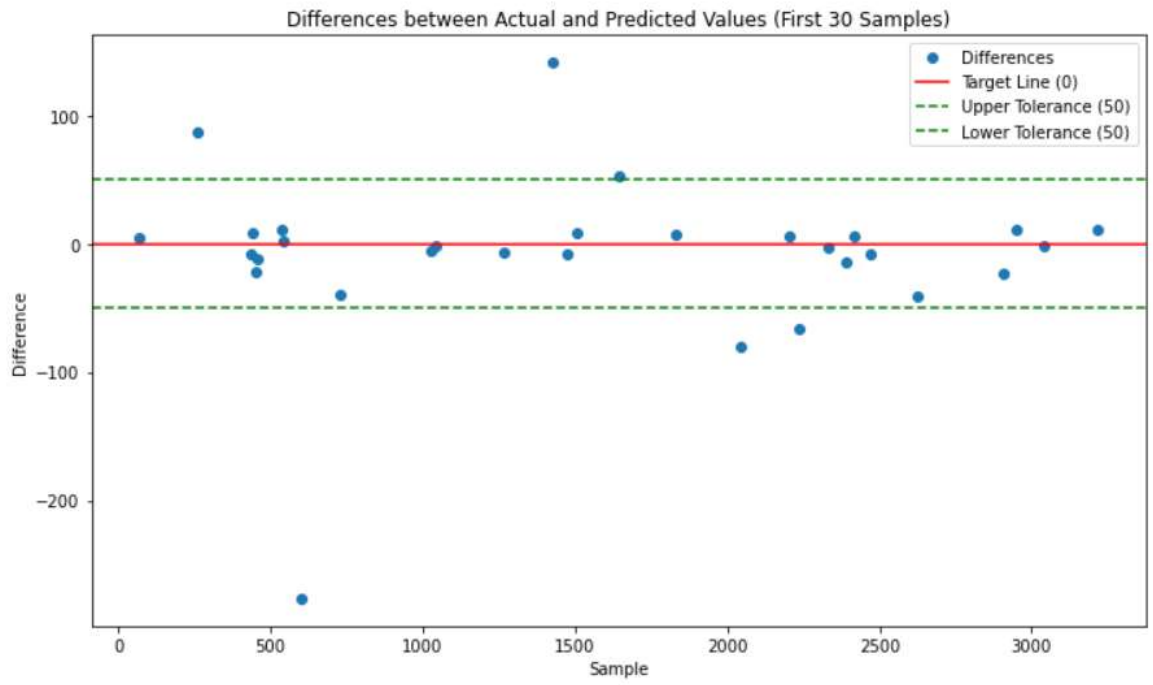


Figure 23: Variations between the expected and actual copper values.

4.7.2 Support Vector Machine (SVM)

For the SVM model, the libraries in Appendix D were imported. The SVM regressor model was created, training of the model and predicting of the target variables were done as per the labelled codes in the same Appendix.

To measure the KNN's model performance, the six model evaluation metrics and results were generated as per statement in Figure 24 below.

```
In [13]: # Calculate evaluation metrics
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
rmse = mse ** 0.5
mae = mean_absolute_error(y_test, y_pred)
n = X_test.shape[0]  # Number of samples in the test set
p = X_test.shape[1]  # Number of independent variables (features)
adj_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
adj_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
explained_variance = explained_variance_score(y_test, y_pred)
```

```
In [14]: # Print the evaluation metrics
print('Mean Squared Error:', mse)
print('R-squared Score:', r2)
print('Root Mean Squared Error (RMSE):', rmse)
print('Mean Absolute Error (MAE):', mae)
print("Adjusted R-squared:", adj_r2)
print('Explained Variance Score:', explained_variance)
```

```
Mean Squared Error: 13922.842551702664
R-squared Score: 0.44811921908456465
Root Mean Squared Error (RMSE): 117.99509545613607
Mean Absolute Error (MAE): 33.68681779825506
Adjusted R-squared: 0.44643151027442574
Explained Variance Score: 0.4581023044387621
```

Figure 24: The SVM model evaluation metrics.

Discrepancy between expected and actual values is, on average, quite large in this instance, according to the Mean Squared Error (MSE) value of 13922.842551702664.

R-squared Score: The R-squared score of 0.44811921908456465 indicates that the features in the SVM model account for around 44.81% of variance in the target variable.

Root Mean Squared Error (RMSE) in this instance is 117.99509545613607, indicating an average error of roughly 117.99 units in the forecasts.

Mean Absolute Error (MAE): In this instance, the MAE value of 33.68681779825506 indicates that the absolute error of the forecasts is roughly 33.69 units on average.

Adjusted R-squared: The model would not gain much from the addition of extra characteristics, as indicated by adjusted R-squared value of 0.44643151027442574, which is rather less than the R-squared score.

Explained volatility Score: The SVM model appears to account for 45.81% of the volatility in the target variable, according to the explained variance score of 0.4581023044387621.

These measurements seem to indicate a moderate level of performance for the SVM model. Nonetheless, as the comparatively high values of MSE, RMSE, and MAE show, performance can

still be improved. The model explains a moderate amount of the variance in the target variable, according to the R-squared and explained variance scores.

Plotting of predictions vs. actual

The bar graph in Figure 25 illustrates the poor trend between the actual and anticipated values.

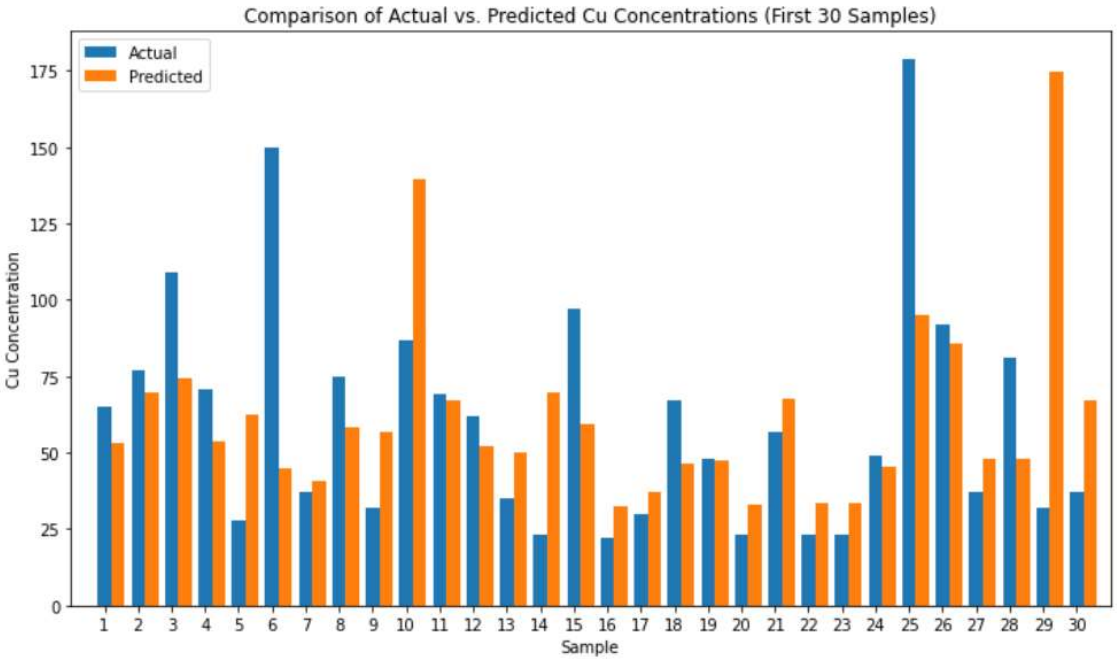


Figure 25: Correlation between actual and predicted values for the SVM model.

Most points in Figure 26 below closely plot around the perfect prediction line, suggesting that predicted and the actual values are generally in good agreement because differences are closer to zero.

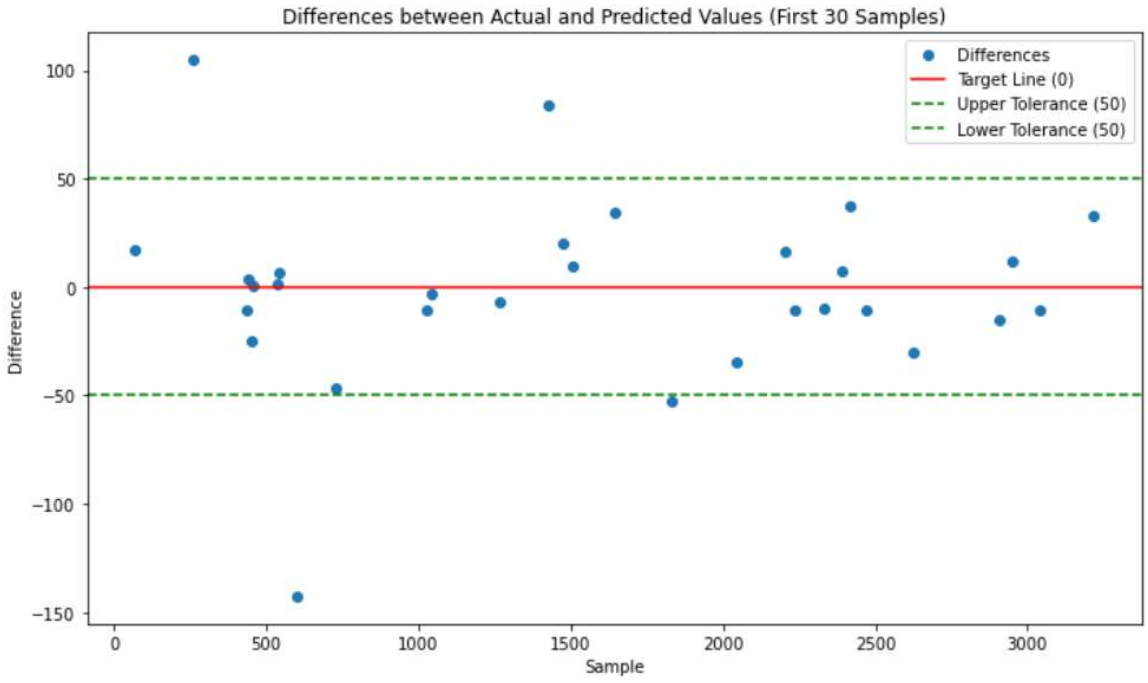


Figure 26: Variations between the SVM model's actual and predicted values.

4.7.3 Decision Trees

For the Decision Tree model, the libraries in Appendix E were imported. The Decision Tree model was created, training of the model and predicting of the target variables were done as per the labelled codes in the same Appendix.

Similarly, the six model evaluation measures that are shown in Figure 27 below were created to evaluate the model's performance.

```
In [10]: # Calculate evaluation metrics
         mse = mean_squared_error(y_test, y_pred)
         r2 = r2_score(y_test, y_pred)
         rmse = mse ** 0.5
         mae = mean_absolute_error(y_test, y_pred)
         n = X_test.shape[0]  # Number of samples in the test set
         p = X_test.shape[1]  # Number of independent variables (features)
         adj_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
         adj_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
         explained_variance = explained_variance_score(y_test, y_pred)

In [11]: # Print the evaluation metrics
         print('Mean Squared Error:', mse)
         print('R-squared Score:', r2)
         print('Root Mean Squared Error (RMSE):', rmse)
         print('Mean Absolute Error (MAE):', mae)
         print("Adjusted R-squared:", adj_r2)
         print('Explained Variance Score:', explained_variance)

         Mean Squared Error: 12905.736787586673
         R-squared Score: 0.48843577953473494
         Root Mean Squared Error (RMSE): 113.60341890800062
         Mean Absolute Error (MAE): 38.35920852359209
         Adjusted R-squared: 0.4868713629583886
         Explained Variance Score: 0.48942242325019447
```

Figure 27: Metrics to evaluate performance of the decision tree model.

Mean Squared Error (MSE): The MSE value of 12905.736787586673 in this instance indicates that the squared difference between the actual and projected numbers is, on average, somewhat large.

R-squared Score: The Decision Tree model's characteristics account for roughly 48.84% of the difference in the target variable, as indicated by the R-squared score of 0.48843577953473494.

Root Mean Squared Error (RMSE): In this instance, RMSE value of 113.60341890800062 indicates that there is an average error of roughly 113.60 units in the forecasts.

Mean Absolute Error (MAE): In this instance, the MAE result of 38.35920852359209 indicates that the absolute error of the forecasts is about 38.36 units on average.

Adjusted R-squared: This figure, which is 0.4868713629583886, is marginally less than the R-squared score, suggesting that adding more characteristics to the model could not have a major positive impact.

Explained Variance: The Decision Tree model accounts for roughly 48.94% of the volatility in the target variable, according to the explained variance score of 0.48942242325019447.

The Decision Tree model performs at a moderate level according to these measures. Nonetheless, similar to other models, there exists potential for enhancement, as demonstrated by the comparatively elevated values of MSE, RMSE, and MAE. The model explains a moderate amount of variance in the target variable, according to the R-squared and explained variance scores.

Ploting of the actuals vs predictions

The graph in Figure 28 below shows a comparison of actual and predicted values. Although there are some differences between the actual and expected numbers on the graph, overall the trend is well maintained.

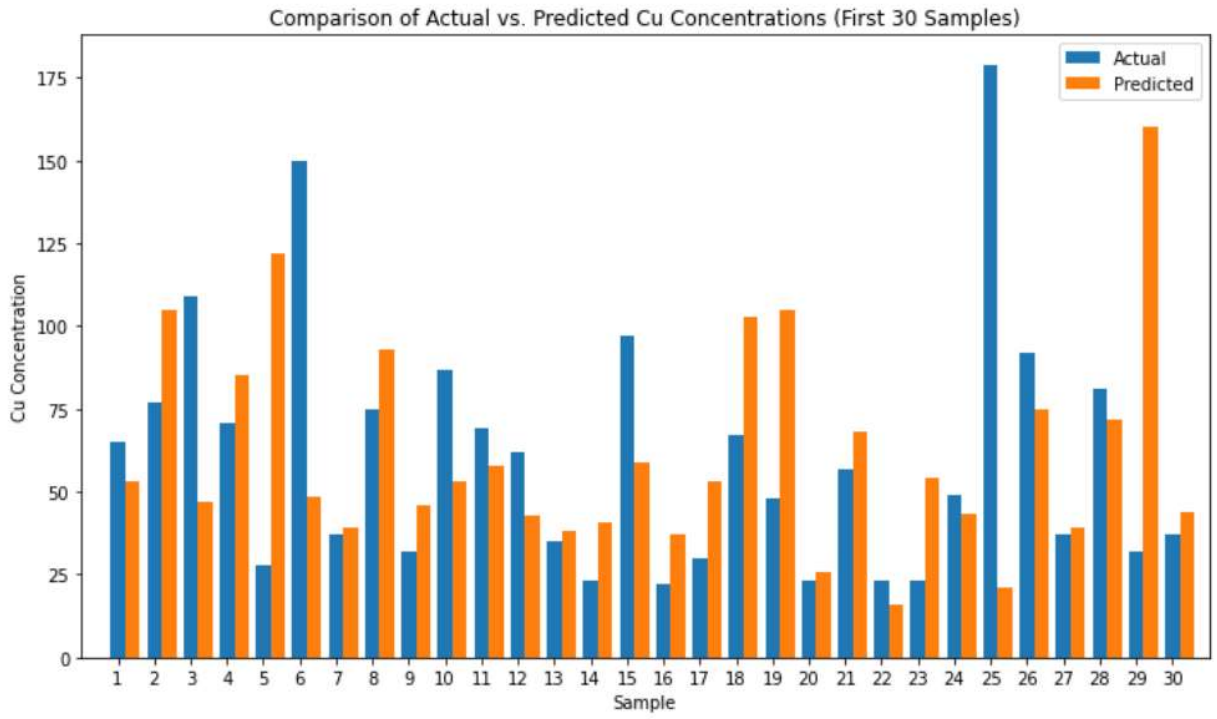


Figure 28: Comparison of the actual versus predicted Copper concentrations.

In addition, a line graph with tolerance limitations of 50 ppm was used to further project the disparities between actual and predicted values. The plot in Figure 30 below generated by the code preceeding it in Figure 29 shows most values are plotting closer to the target line (0), meaning the predictions are accpetable and there is no much differences except for the few outliers plotting outside the tolerance limit lines.

```
In [28]: plt.figure(figsize=(10, 6))

         # Calculate the differences between actual and predicted values for the first 30 samples
         differences = y_test[:30] - y_pred[:30]

         # Plotting the differences with tolerance bands
         plt.plot(differences, marker='o', linestyle='', label='Differences')
         plt.axhline(y=0, color='red', linestyle='-', label='Target Line (0)')
         plt.axhline(y=50, color='green', linestyle='--', label='Upper Tolerance (50)')
         plt.axhline(y=-50, color='green', linestyle='--', label='Lower Tolerance (50)')

         plt.xlabel('Sample')
         plt.ylabel('Difference')
         plt.title('Differences between Actual and Predicted Values (First 30 Samples)')
         plt.legend()
         plt.tight_layout()

         plt.show()
```

Figure 29: Code to generate the graph of differences between actual versus predicted values.

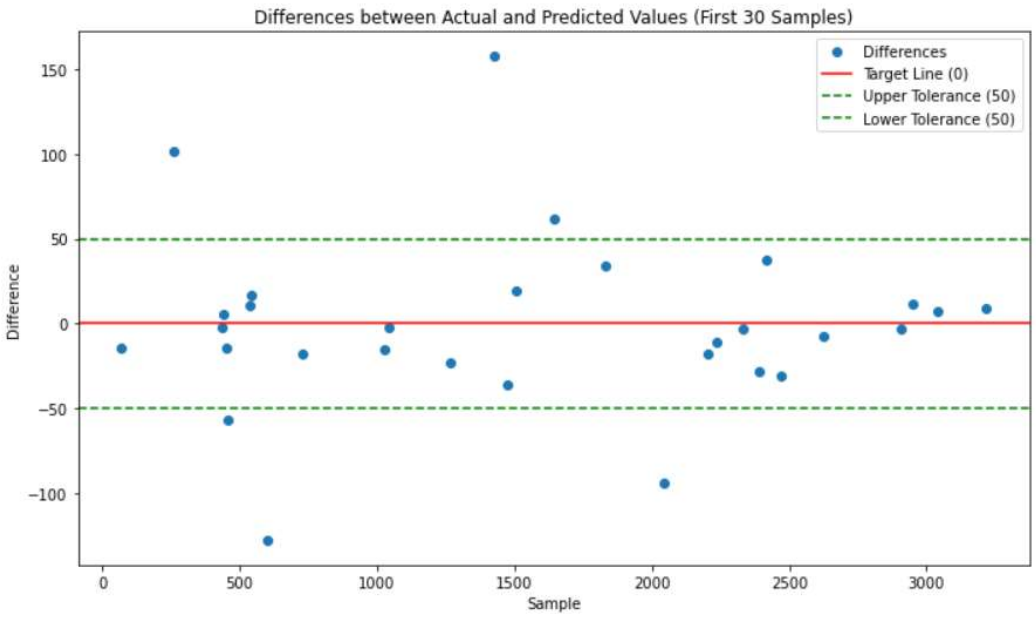


Figure 30: Discrepancies between the decision tree model's projected and actual values.

4.7.4 Random Forest

For the Random Forest model, the libraries in Appendix F were imported. The model was created, training of the model and predicting of the target variables were done as per the labelled codes in the same Appendix. The RandomForestRegressor class from scikit-learn library is used to generate a Random Forest Regressor model in the Python code snippet given below. The RandomForestRegressor was initialized with the following parameters:

The number of decision trees (estimators) to be generated in the random forest is indicated by the variable n_estimators. One hundred decision trees were made in this instance.

random_state: For repeatability, it establishes a random seed. The same random state will be utilised every time the code is run if it is set to 42, guaranteeing consistent outcomes.

Once the RandomForestRegressor object is created, it can be used to fit the model as per the snippet code in Appendix F, to the training data and make predictions.

Plotting the predictions

The following Figure 31, "Actual vs. Predicted Cu Concentration," is a bar graph with the actual Cu predicted and actual metal concentrations. Although there are some good corellation for a number of samples, there are outliers with large differences between the actual and expected numbers on the graph.

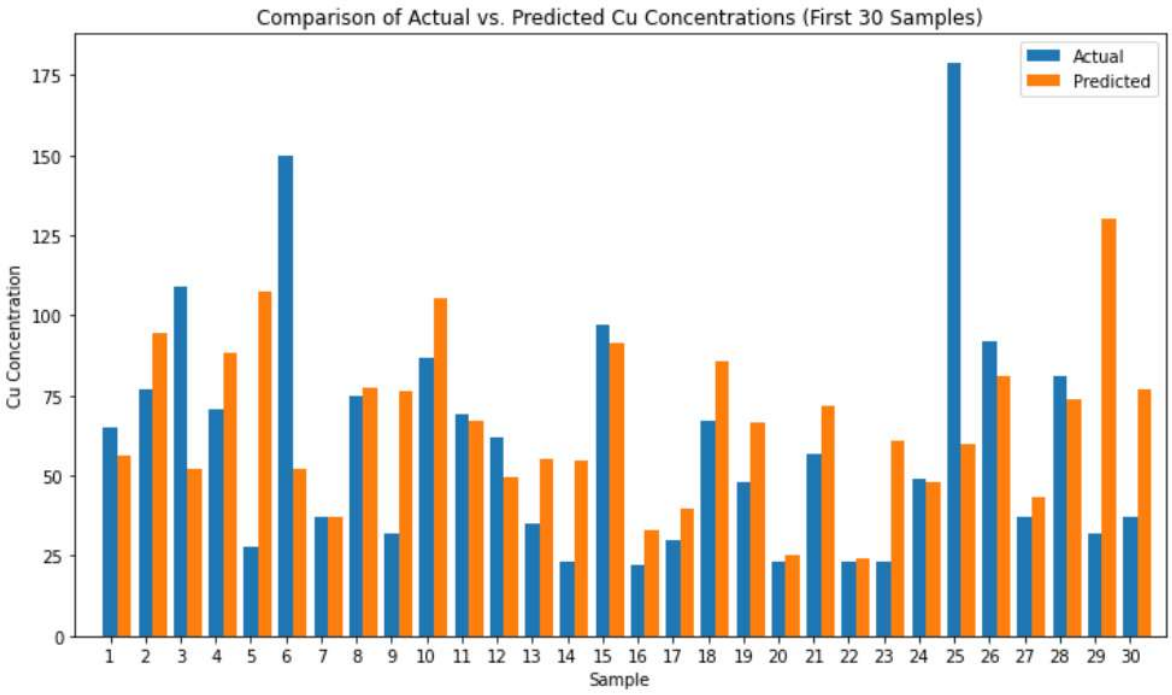


Figure 31: Comparison of actual versus predicted Cu concentrations in the random forest model.

The next Figure 32, titled "Residual Plot," displays a scatter plot with the predicted Cu metal concentrations (y_pred) on the x-axis and the residuals on the y-axis. Prediction mistakes of the model are shown by the residuals. When the residuals at y=0 are dispersed randomly

along the horizontal line, it indicates that the model's predictions are impartial. Potential problems with the model's performance may be indicated by patterns or trends in the residuals.

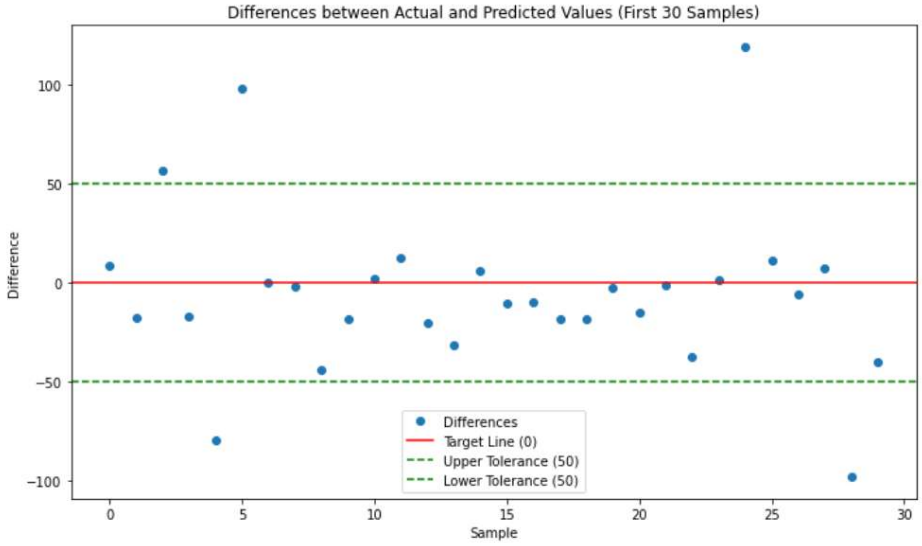


Figure 32: Differences between the actual and predicted values in the random forest model.

Again, to measure RF model's performance, the six model evaluation metrics were generated in figure 33 below.

```
In [20]: # Calculate evaluation metrics
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
rmse = mse ** 0.5
mae = mean_absolute_error(y_test, y_pred)
n = X_test.shape[0]  # Number of samples in the test set
p = X_test.shape[1]  # Number of independent variables (features)
adj_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
adj_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
explained_variance = explained_variance_score(y_test, y_pred)

In [21]: # Print the evaluation metrics
print('R-squared Score:', r2)
print('Root Mean Squared Error (RMSE):', rmse)
print('Mean Absolute Error (MAE):', mae)
print("Adjusted R-squared:", adj_r2)
print('Explained Variance Score:', explained_variance)

R-squared Score: 0.5469744364140721
Root Mean Squared Error (RMSE): 106.90614653005476
Mean Absolute Error (MAE): 32.202909839864176
Adjusted R-squared: 0.5455890371370509
Explained Variance Score: 0.5486224092775687
```

Figure 33: Evaluation metrics to measure performance of the random forest model.

Mean squared difference between expected and actual Cu metal concentrations in the test data is represented by the Mean Squared Error (MSE) value of 11428.924165905539.

With an R-squared of 0.5469744364140721, the independent variables in the Random Forest model (Zn and Pb) can account for roughly 54.70% of the variability in the Cu metal concentrations.

Taking into account the quantity of features and samples, an adjusted R-squared score of 0.5455890371370509 suggests that independent variables (Zn and Pb) in Random Forest model can account for roughly 54.56% of the variability in the Cu metal concentrations. Taking into account the quantity of features and samples utilised, the modified R-squared value of 0.5456 in this instance indicates that the model can explain a moderate amount of the variability in the Cu metal concentrations.

The predicted values have an average error of about 106.91 units, according to the model, which produced an RMSE value of 106.90614653005476.

The model's MAE was 32.202909839864176, indicating an average absolute inaccuracy of about 32.20 units in the predictions.

With an explained variance score of 0.5486224092775687, the Random Forest model is able to explain roughly 54.86% of the volatility in the target variable.

With an MSE value of 11428.924165905539, the model indicates that the squared difference between the actual and predicted values is generally quite high.

These measurements seem to indicate a moderate level of performance for the model. A moderate amount of variance in the target variable is explained by the model, according to the R-squared, modified R-squared, and explained variance scores. When compared to some of the other models covered above, the model's average prediction errors appear to be rather low, based on the RMSE and MAE figures.

4.8 Evaluation summary

The table 6 below is the summary of different model performances on predicting Copper metal contents from the Zinc and Lead metal contents in samples as presented by different metrices in section 4.7.

Table 6: Summary of the model performances based on the metrics evaluated.

Metrics	SVM Model	KNN Model	DT Model	RF Model
Mean Squared Error:	13922.84	10866.86	12905.74	11428.92
R-squared Score:	0.45	0.57	0.49	0.55
Root Mean Squared Error (RMSE):	118.00	104.24	113.60	106.91
Mean Absolute Error (MAE):	33.69	32.83	38.36	32.20
Adjusted R-squared:	0.45	0.57	0.49	0.55
Explained Variance Score:	0.46	0.57	0.49	0.55

A higher number in the R-squared score denotes a better fit. It quantifies percentage of variance in the target variable that the model explains. It is imperative that choosing a model shouldn't be based just on one metric because different models can perform better in different contexts. Instead, it's crucial to take into account a variety of metrics, including explained variance score, RMSE, MAE, adjusted R-squared, and explained variance.

By comparing these measures, the R-squared score, RMSE, MAE, Adjusted R-squared, and explained variance score show that the KNN model performs better than other three models. It obtain the highest R-squared score of 0.57, meaning that the features account for around 0.57% of variance in the target variable.

Furthermore, in comparison to the other models, the KNN model has the lowest RMSE (104.24) and MAE (32.83), indicating that, on average, the predictions have less mistakes. Additionally, the KNN model shows greater explained variance and modified R-squared scores, demonstrating improved model fitting and the capacity to account for variance in the target variable.With these things taken into account, the KNN model seems to be the most suitable option out of the four models for predicting the dataset.

4.9 Fine-tuning of the leading model (K-NN)

The K-NN model performed well in terms of prediction capacity for the data that was used. The purpose of this section was to further optimise the K-NN model's parameters and determine whether the model will perform any better using the same dataset. Figure 34 represents the python code for the fine tuned parameters in the K-NN model and thereof the improved predictions shown by the R^2 metric of 70%. The two parameters fine tuned were the testing size and the nearest neighbour estimator. The combination that yielded an improved prediction accuracy (R^2) was when the testing dataset was set to 10% (0.1) and the nearest estimators set to 4.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error, explained_variance_score

df = pd.read_csv("C:/Users/LYDIA/Downloads/Analytical_Dataset.csv")

# Drop rows with missing values
df.dropna(inplace=True)

# Split the dataset into features (Zn and Pb) and the target variable (Cu)
X = df[['Zn (ppm or g/t)', 'Pb (ppm or g/t)']]
y = df['Cu (ppm or g/t)']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)

# Scale the numerical features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Create a K-NN regressor object
knn = KNeighborsRegressor(n_neighbors=4)

# Train the model
knn.fit(X_train_scaled, y_train)

# Predict the target variable for the test set
y_pred = knn.predict(X_test_scaled)

# Calculate evaluation metrics
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
rmse = mse ** 0.5
mae = mean_absolute_error(y_test, y_pred)
n = X_test.shape[0]  # Number of samples in the test set
p = X_test.shape[1]  # Number of independent variables (features)
adj_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
adj_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
explained_variance = explained_variance_score(y_test, y_pred)

# Print the evaluation metrics
print('Mean Squared Error:', mse)
print('R-squared Score:', r2)
print('Root Mean Squared Error (RMSE):', rmse)
print('Mean Absolute Error (MAE):', mae)
print("Adjusted R-squared:", adj_r2)
print('Explained Variance Score:', explained_variance)

Mean Squared Error: 12747.825607902736
R-squared Score: 0.7039309065485759
Root Mean Squared Error (RMSE): 112.90626912577855
Mean Absolute Error (MAE): 34.49696048632219
Adjusted R-squared: 0.7021145317421256
Explained Variance Score: 0.7052188515950202
```

Figure 34: The code for fine-tuning the K-NN model.

Table 7 represents the summary of the six metrics performances at different parameter combinations for the K-NN model. The combination of test data at 20% (0.2) and 5 n estimators is what was initially used. To further improve the model, these two parameters were further fine tuned. The best combination was when the test dataset was set to 10% (0.1) and n estimators set to 4. This resulted in an R^2 of 70% (0.70).

Table 7: Summary of K-NN model metrics at different parameter combinations.

Test data, n estimators parameters Combination	**R^2**	**MSE**	**RMSE**	**MAE**	**Adjusted R^2**	**Explained Variance**
test=0.1, n=2	0.54	19640	140	39	0.54	0.54
test=0.1, n=3	0.67	14235	119	35	0.67	0.67
test=0.1, n=4	**0.70**	**12747**	**113**	**34**	**0.70**	**0.71**
test=0.1, n=5	0.68	13733	117	34	0.68	0.68
test=0.1, n=6	0.63	16104	127	35	0.62	0.63
test=0.2, n=2	0.52	12219	111	35	0.51	0.52
test=0.2, n=3	0.60	10135	101	33	0.60	0.60
test=0.2, n=4	0.64	9055	95	32	0.64	0.64
test=0.2, n=5	**0.57**	**10867**	**104**	**33**	**0.57**	**0.57**
test=0.2, n=6	0.54	11573	108	32	0.54	0.55
test=0.3, n=2	0.54	8466	92	34	0.54	0.54
test=0.3, n=3	0.54	8440	92	33	0.54	0.54
test=0.3, n=4	0.55	8278	91	31	0.55	0.55
test=0.3, n=5	0.53	8576	93	31	0.53	0.53
test=0.3, n=6	0.50	9229	96	30	0.49	0.50

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

The results from this study are aggregated in this Chapter, which summarises the goals of the reasearch, identifies shortcomings, and recommends possible directions for future work.

5.1 Revisiting the research objectives and questions

As stated in Chapter 1, the study's key objective was to assess the effectiveness of machine learning techniques in predicting metal concentrations in copper deposits and to investigate machine learning on geochemical data. There were three sub-objectives for this:

The first sub-objective was to explore the metal composition patterns in copper deposits. The observation is that, 99% of the Zn, Cu, and Pb metal concentrations are below 500ppm. The scatter plot further shows that for high Cu concentrations the Pb concentrations are proportionately high, whereas the Zn concentrations are much lower. Furthermore, analysis shows that higher Cu concentrations are more common at Otasline, Otagross and Otainsel while the lower Cu concentrations were more common at Oaasisk, RL, Otaschn, Otastr and RP locations.

The second sub-objective: to evaluate the performance of the four machine learning techniques (RF, DT, KNN and SVM) on the geochemistry data was achieved under section 5.7 where six different metrics were evaluated for all the four algorithims. This was presented in section 5.8. Sub-question ii) on how well the different four machine learning models perform, and was based on the R-squared score.

Plotting actual versus anticipated values for each model and comparing the R-squared score of the primary research metric allowed for the achievement of the third sub-objective, which was to identify the best approach based on performance metrics. The regression line (R-squared) for the K-NN model showed the best plot, with a score of 0.57. The RF model was second with a score of 0.55, and the SVM model had the lowest score of 0.45. By comparing these measures,

the R-squared score, RMSE, MAE, Adjusted R-squared, and explained variance score show that the KNN model outperforms the other three models.

The target variable's greatest R-squared score of 0.57 means that the features account for around 0.57% of the variation in the variable. Furthermore, compared to other models, the KNN model has the lowest RMSE (104.24) and MAE (32.83), indicating that, on average, the predictions have less mistakes. Additionally, the KNN model shows greater explained variance and modified R-squared scores, demonstrating improved model fitting and the capacity to account for variance in the target variable.

With all of these variables taken into account, the K-NN model seems to be the most suitable option out of the four models for predicting the copper deposits in the Kombat region. The parameters of the n-estimator and test size were adjusted further to enhance prediction accuracy of the leading model (K-NN). An improved R-squared score of 70% (0.70), was achieved with the optimal combination of test dataset set to 10% (0.1) and n estimators set to 4.

5.2 Recommendations and future work

The dataset for this study contained few features that were strongly correlated (i.e. concentrations for three metals, namely: Zn, Pb, Cu). The study focused on using concentrations of Zn and Pb to predict concentration of Cu. Future studies can consider collecting samples with additional features such as lithologies, mineral associations in those samples, sample colour as these could be utilised as input features in models to predict presence of Cu and other metals in these deposits.

The K-NN model created can be used to predict metal concentrations in copper deposits in the Kombat area. The data used for the research was a combination of different deposits in oder to have an adequate dataset to work with. Checking performances of the models on individual sites was not ideal as data from individual sites was minimal. To improve on the predictions, using more data from specific sites would improve the results. Additionally, to improve the

performance of these new models, future experimentation should consider the following aspects:

- Using grid search techniques and parameters like gamma, regularisation parameter, and kernel type to influence the model's performance, further optimisation of the hyper-parameters of the models is carried out.

- If at all possible, expand the training dataset from a single deposit site; more data can enhance model performance and improve generalisation.

- Assessing the effectiveness of other machine learning methods like Artificial Neural Networks, and investigating possibilities of integrating machine learning with other methods of data analysis, such as geostatistics, for the purpose of predicting metal concentrations.

This study demonstrates the potential of machine learning to predict metal concentrations, given some geochemical data, which could save exploration costs, boost productivity, and enhance metal concentration estimation accuracy. The R2 value of 0.70 is promising, but is not sufficient for an ML model to be deployed for the metal prediction task. Other variables like the textures, colour, lithologies of samples could be captured to improve the predictive power of the model. Using grid search techniques, gamma, kernel type, regularisation parameters could influence the model’s performance. For future work, further hyperparameter tuning could be done to improve the models. In addition, assessing the effectiveness of other machine learning methods such as Artificial Neural Networks can also be carried out. Furthermore, having seen the promising results of ML on geochemical data, investigating the integration of ML-based ore grade estimates into automated mineral extraction process optimisation would be the next step.

REFERENCES

ALS, G. (2023). *Geochemistry*. Retrieved May 10, 2023, from https://www..alsglobal.com/en/services-and-products/geochemistry

Antoine, A., & Miranda, E. (2017). Musical Acoustics, Timbre, and Computer - Aided Orchestration Challenges.

Arslan, H., Aslan, N., & Demirci, S. (2021). Mineral prospectivity mapping using machine learning techniques in the Murgul area, northeastern Turkey. 557-577.

Bansal M, Goyal A, Choundhary A. (2022). A comparative analysis of K-nearest neighbour, genetic support vector machine, decision tree and long short term memory algorithms in machine learning. *Decision Analytics Journal* .

Boateng E. Y., Otoo J. & Abaye D. A. (2020). Basics tenets of classification algorithims K-nearest neighbour, support vector machine, random forest and neural network. 341-357.

Bortey-Sam, N., Nakayama, S. M., Ikenaya, Y., Akoto, O., Baidoo, E., Mizukawa, H., *et al.* (2018). *Human health risk assessment of heavy metals in soil and food crops from farms in Ghana receiving irrigation with polluted urban wastewater.*

Breiman, L. (2001). Random forests. Machine Learning. *Adventure Works Monthly* , 5-32.

Cate, A., Perozzi, L., Gloguen, E., & Blouin, M. (2017). Machine learning as a tool for geologists. *The Leading Edge* , 36.

Charbuty B., & Abdulazeez A. . (2021). Classification based on decision tree algorithim for machine learning. *Journal of Applied Science and Technology Trends* , 20-28.

Cunningham P. & Delany S. J. (2021). K-Nearest neighbour classifiers- A tutorial. *ACM computing surveys* , 1-25.

Dalmia, A., & Nayak, A. (2019). Comparison of digestion methods for ICP-MS based trace element analysis of soil samples. *MethodsX* , 1539-1549.

Dumakor-Dupley, N. K., & Ayra, S. (2021). Machine Learning - A review of Applications in Mineral Resource Estimation. *Energies* , 14.

Glasgow, R. (2013). What does it mean to be pragmatic? Pragmatic methods, measures, and models to facilitate research translation. . *Health Education & Behavior* , 20.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer.

Hilson, G., & McQuilken, J. (2014). Four decades of support for artisanal and small-scale mining in sub-Saharan Africa. *Resource Policy , 41*, 52-59.

Johnson, A., & Brown, C. (2019). Importance of laboratory analysis for copper deposits. *Geology and Mineral Resources* , 123-136.

Liu, X., Ma, Y., Ma, X., & Zhou, C. (2019). The application of a convolutional neural network to the mapping of mineral distributions using remote sensing data. *International Journal of Remote Sensing* , 157-173.

Navada A., Ansari A., Patil S. & Sonkambe B. A. (2011). Overview of use of decision tree algorithms in machine learning. *IEEE* , 37-42.

Probst P., Wright M N., & Boulesteix A. L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews* , 1-13.

Raschka, S., & Mirjalili, V. (2017). *Python Machine Learning.* Packt Publishing.

Saunders, M., Lewis, P., & Thornhill, A. (2017). *Research methods.* England: Pearson Education Limited.

Sharol, S. M., & Tiko, I. (2011). A Guide to Selecting Theory to underpin Information Systems Studies. 1.

Smith, J. (2018). Mineralisation and the formation of copper deposits. *Earth Science Review* , 78-91.

Somvanshi M., Chavan P., Tambade S., & Shinde S. V. (2016). A review of machine learning techniques using decision tree and support vector machine. *IEEE* , 1-7.

Tanveer M, Rajani T, Rastogi R, Shao Y H, Ganaie M A. (2022). Comprehensive review on twin support vector machines. *Annals of Operations Research* , 1-46.

Wang, H., Sun, W., & Cheng, Q. (2020). 2020. *Ore Geology Review* , 122.

Wenau, S., Spiess, V., Pape, T., & Fekete, N. (2015). Cold seeps at the salt front in the lower Congo Basin II: The impact of spatial and temporal evolution of salt-tectonics pn hydrocarbon seepage.

Zaki, M. M., Chen, S., Zhang, J., Feng, F., Khoreshok, A., Mahdy, M. A., *et al.* (2022). A novel approach for Resource Estimation of highly skewed Gold using Machine Learning Algorithms. *Minerals* , 12.

APPENDICES

A. Certificate of Presentation at the IEOM Society Conference for publication

IEOM Society International

5th African International Conference on
Industrial Engineering and Operations Management
Pretoria, South Africa, Venue: Venue: CSIR ICC, Hosts: UNISA and SOMA

Certificate of Presentation

This is to certify that

Lydia Joel and Richard Maliwatu,
Faculty of Computing and Informatics, Department of Informatics,
Namibia University of Science and Technology (NUST), Windhoek, Namibia

Delivered an Oral Presentation Entitled "ID 204: Exploring Machine Learning on Geochemistry Data for Estimating Metal Concentrations in Copper Deposits."
Presented at the Fifth IEOM South Africa Conference.

Dr. Anthea Amadi-Echendu
Conference Chair
Senior Lecturer
Operations Management
University of South Africa (UNISA)
Pretoria, South Africa

Eldon G. Caldwell Marin, Ph.D., Sc.D., Dr.Ed.
Professor, Department of Industrial Engineering
University of Costa Rica
President, IEOM Society

Dr Ahad Ali
Conference Co-Chair
Executive Director – IEOM Society
Associate Professor and Director of Industrial Engineering Program
Lawrence Tech University, MI, USA

Professor Don Reimer
Chief Operating Officer
IEOM Society International
Adjunct Faculty, Lawrence Technological University,
Southfield, Michigan, USA

Sponsors and Partners

IEOM Society International, 21411 Civic Center Drive, Suite # 205, Southfield, Michigan 48076, USA, www.ieomsociety.org

B. Data shape code

```
In [10]: df.shape
Out[10]: (3282, 8)
```

C. K-NN Model

```
In [1]: import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import StandardScaler
        from sklearn.neighbors import KNeighborsRegressor
        from sklearn.metrics import mean_squared_error
```

```
In [7]: # Create a K-NN regressor object
        knn = KNeighborsRegressor(n_neighbors=5)
```

```
In [8]: # Train the model
        knn.fit(X_train_scaled, y_train)
Out[8]: KNeighborsRegressor()
```

```
In [9]: # Predict the target variable for the test set
        y_pred = knn.predict(X_test_scaled)
```

D. SVM Model

```
In [1]: import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import StandardScaler
        from sklearn.svm import SVR
        from sklearn.metrics import mean_squared_error, r2_score
```

```
In [10]: # Create an SVM regressor
         svr = SVR(kernel='linear')
```

```
In [11]: # Train the model
         svr.fit(X_train, y_train)
Out[11]: SVR(kernel='linear')
```

```
In [12]: # Predict the Cu metal concentrations for the test set
         y_pred = svr.predict(X_test)
```

E. Decision Tree Model

```
In [1]: import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.tree import DecisionTreeRegressor
        from sklearn.metrics import mean_squared_error, r2_score
```

```
In [8]: model = DecisionTreeRegressor(random_state=42)

In [9]: model.fit(X_train, y_train)
Out[9]: DecisionTreeRegressor(random_state=42)
```

```
In [10]: y_pred = model.predict(X_test)
```

F. Random Forest

```
In [1]: import pandas as pd
        from sklearn.ensemble import RandomForestRegressor
        from sklearn.model_selection import train_test_split
        from sklearn.metrics import mean_squared_error
```

```
In [14]: # Train the Random Forest model
         rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
         rf_model.fit(X_train, y_train)
Out[14]: RandomForestRegressor(random_state=42)
```

G. Codes for plots generation

APPENDIX H: LANGUAGE EDITING CERTIFICATE

Dr.GM

The Rev. Dr. Greenfield Mwakipesile

ThD, MBA, HBS | mwakipg@outlook.com

CONTACT

PO Box 99539,
UNAM,
Windhoek,
Namibia

LANGUAGE & COPY-EDITING CERTIFICATE

24th August, 2024

RE: LANGUAGE, COPYEDITING AND PROOFREADING OF LYDIA JOEL'S THESIS FOR THE MASTER OF DATA SCIENCE DEGREE IN THE DEPARTMENT OF INFORMATICS AT THE NAMIBIA UNIVERSITY OF SCIENCE AND TECHNOLOGY (NUST)

This certificate serves to confirm that I copyedited and proofread **LYDIA JOEL'S** Thesis for the **MASTER OF DATA SCIENCE DEGREE** entitled: **EXPLORING MACHINE LEARNING ON GEOCHEMISTRY DATA FOR EFFICIENT PREDICTION OF METAL CONCENTRATIONS IN COPPER DEPOSITS**

I declare that I professionally copyedited and proofread the thesis and removed mistakes and errors in spelling, grammar, and punctuation. In some cases, I improved sentence construction without changing the content provided by the student. I also removed some typographical errors from the thesis and formatted the thesis so that it complies with the Namibia University of Science and Technology's guidelines.

I am a trained language and copy editor and have edited many Postgraduate Diploma, Masters' Thesis, Dissertations and Doctoral Dissertations for students studying with universities in Namibia, Zimbabwe, Eswatini, South Africa and abroad. I have also copy-edited company documents for companies in the region and abroad.

Please feel free to contact me should the need arise.

Yours Sincerely,

The Rev. Dr. Greenfield Mwakipesile

greenfield.mwakipesile

@mwakipg

+264813901701

Dr. Greenfield Mwakipesile