UiT The Arctic University of Norway

Faculty of Science and Technology
Department of Computer Science

Enhancing Prediction of Blast-Induced Ground Vibrations through Machine Learning

Guro Jansrud

Master's thesis in Computer Science - INF-3981

“I particularly hope that you will conclude the merit of the ideas I present outweigh my defects as a writer.”
–Philip A. Fisher

“To start something is good, but to finish it is a miracle.”
–Richard Strawbridge

Abstract

This Master's thesis investigates the application of Machine Learning (ML) in predicting blast-induced ground vibrations in mining, with the aim of surpassing the precision of the current industry-standard model that utilizes an empirical, regression-based method. The study applied a Deep Neural Network (DNN) model, selected for its capability to consider a broader range of variables than the industry-standard model, leading to significantly enhanced predictive capabilities. The evaluation of these models was conducted using three statistical criteria: coefficient of correlation (R^2), mean square error (MSE), and mean absolute error (MAE).

The key finding is the DNN model's superior performance, achieving an R^2 of 0.94, an MSE of 0.94, and an MAE of 0.60, which represent a significant improvement and reduction over the industry-standard model's predictive results. Specifically, there is an 84% improvement in the R^2 value, an 87% decrease in MSE, and a 71% decrease in MAE compared to the industry-standard model's R^2 of 0.51, MSE of 7.41, and MAE of 2.04. This marked enhancement in predictive accuracy illustrates the model's ability to analyze multiple variables concurrently and highlights the potential of AI and ML to improve environmental safety and operational efficiency in the mining industry.

Acknowledgment

I want to thank my primary supervisor Anne Håkansson for all the help during this autumn, with her insights, structured approach, and invaluable advice. My gratitude also extends to my co-supervisor, Aya Saad, whose expertise and perspectives have significantly enriched this thesis. Their combined feedback and support have been instrumental for shaping this thesis.

Thank you to all my fellow students at UiT, and to my friends and family for always supporting me, and believing in me, which has been a constant source of strength.

Lastly, my deepest appreciation is reserved for my partner, for being my biggest supporter. During my toughest times, that must be have been the biggest of challenges. Your presence and support have been a guiding light throughout this journey, and I am immensely grateful for everything you've done. This achievement is as much yours as it is mine.

Contents

List of Figures

List of Tables

1

Introduction

Artificial Intelligence (AI) serves as a sub-field of Computer Science, concentrating on the development of algorithms and models that enables machines to execute tasks, traditionally requiring human behavior. One of AIs sub-disciplines, Machine Learning (ML), aims to design algorithms that enable machines to improve their performance through data extraction and adaption to the knowledge from this data [23]. AI and ML can provide environmental data insights, and autonomously take actions that promote sustainability, reducing company costs and free up time and resources for other priorities.

The mining industry stands as an important component of the global economic infrastructure, contributing substantially to both development, labor productivity and employment rates, by discovering and extracting materials and metals from the earth [58]. Among various operations in the mining industry, blasting has a significant role, as it is recognized as a cost-effective means of fracturing rock formations, thus enabling excavation [9, 8, 28, 27, 6, 49].

Blasting operations are initiated through the controlled detonation of explosives within blast holes, with the primary aim to fragment and displace rock masses to enable excavation. The objective is not only to achieve optimal rock fragmentation but also to minimize the environmental impacts, ensuring safety, environmental responsibility, and conformance to governing regulations. [18, 55].

When explosives are detonated in a mine, they release a tremendous amount of energy in the form of shock waves [18]. These shock waves propagate through the surrounding rock mass, causing the ground or rock at the blast site to vibrate and move, with the potential to cause structural or environmental damages [69, 24, 3, 54]. Due to environmental consequences, controlling blast-induced ground vibration is a crucial aspect of mining operations. To alleviate these consequences and to control blast-induced ground vibrations generated by blasting, it is essential to implement effective control predictive measurements and continuously evaluate blast-induced ground vibration [69, 24, 3, 54].

To achieve this level of control, it is imperative to identify the parameters that exert significant influence. The parameter used to predict and evaluate blast-induced ground vibrations is known as Peak Particle Velocity (PPV). PPV represents the highest velocity reached by individual particles in the ground as they respond to these seismic waves. In the context of mining operations, PPV serves as the key evaluation metric used to characterize the ground motion resulting from the detonation of explosives [69, 24, 3, 54].

In this complex interplay of human, machine, and the environment, technology has come to play an increasingly central role [58]. Technological advancements have enabled the mining industry to scale up production, enhance safety, and address environmental impacts more effectively. The integration of cutting-edge technologies, including AI and ML, holds the promise of tackling industry challenges by refining critical operations and potentially revolutionizing the sector [58].

Incorporating Machine Learning (ML) into the mining industry, aligns seamlessly with the industries technological endeavours. The implementation of a ML model in this context offers a sophisticated approach to addressing the challenge of predicting and therefore better controlling blast-induced ground vibrations [33, 34, 64]. Machine Learning models enhance the analysis of various parameters influencing blast-induced ground vibrations, offering a more comprehensive approach when compared to traditional methods. ML's ability to process and learn from a dataset enables a model to distinguish complex patterns and relationships between various blasting parameters and their impact on Peak Particle Velocity (PPV), which could reduce the limitations of current industry empirical formulas [65].

The implementation of a ML model in the mining industry represents a paradigm shift towards data-driven decision-making. This shift not only enhances the precision of PPV predictions but also contributes to reducing environmental impact and improving safety standards in blasting operations.

The application of ML in this field demonstrates a commitment to innovation and sustainability. It aligns with the industry's goal to optimize operations while minimizing adverse environmental effects. By improving the accuracy of PPV predictions, a ML model can be instrumental in ensuring that blasting vibration outcomes comply with regulatory standards, while also improving community relations. It achieves this by proactively mitigating the risks associated with blast-induced ground vibrations. [33, 34, 64, 65].

This thesis explores the application of Machine Learning (ML) within the mining sector, focusing specifically on enhancing the prediction of blast-induced ground vibrations. Advancements in this area align well with the industry's ongoing focus towards sustainable practices.

1.1 Background And Motivation

Accurately predicting, and therefore controlling blast-induced ground vibrations is essential in mining. It is vital for ensuring safety, environmental integrity, and compliance with legal and regulatory standards. In the context of evolving environmental concerns and stringent safety regulations, accurate vibration prediction becomes crucial for the protection of nearby structures, personnel safety, and ecological preservation. Despite its significance, the industry-standard method, established in the 1970s, often falls short in capturing the complexities of blast-induced vibrations due to it's limited ability to consider multiple variables simultaneously. This highlights a need for more advanced and comprehensive predictive models to be used in the industry.

The current industry-standard method for predicting blast-induced ground vibration employs a two-parameter approach for the computation of PPV, namely, explosive mass per delay, and distance. Distance, in this context, is specifically defined as the "minimum separation distance between a blast hole and a designated point of interest". Typically, this point of interest corresponds to a location dedicated to vibration monitoring, which plays a critical role in overseeing the vibration aspects of blast operations [46].

Another critical parameter is the determination of 'the maximum charge mass per delay,' commonly referred to as the Maximum Instantaneous Charge (MIC). MIC represents the cumulative charge mass of explosives that initiates simultaneously during a blasting event. These two parameters constitute fundamental metrics in the contemporary management of blasting-induced vibration, serving as the prevailing industry standards for PPV quantification. [69, 24, 3, 54].

Artificial Intelligence (AI) provides computers to simulate human intelligence, offering significant advantages in areas such as task automation, data-driven decision-making, and digital assistance [17]. By leveraging data, AI systematically extracts valuable insights that can guide informed decision-making processes. Machine Learning (ML), a specialized sub field of AI, utilizes sophisticated algorithms to analyze extensive datasets, and enables to uncover patterns within the data that are often challenging or time-consuming for humans to identify. As a result of this, ML is capable of facilitating more accurate and efficient decisions [17].

Within this framework, ML algorithms play a crucial role as they thoroughly explore the data, examining each variable to understand underlying patterns and relationships, which otherwise might be difficult to recognise. This analytical capability allows for the creation of mathematical models that reflect these identified patterns. In the context of the mining industry, related work on applying ML has shown promise in assisting professionals to make more accurate predictions of PPV, and this thesis aims to build upon those existing results[36].

1.2 Problem Definition

This study addresses the critical challenge of improving the accuracy of predictions for blast-induced ground vibrations in the mining sector. The sector currently relies on a predictive method developed in the 1970s, which is constrained to considering only two blast design variables. Given the capability of ML to analyze a multitude of variables comprehensively, this industry-standard method is likely producing sub-optimal results in comparison. Such limitations can lead to inadequate risk management, especially in areas critical to safety, environmental responsibility, and regulatory compliance. Persisting with this outdated approach risks neglecting more advanced models that could significantly improve predictive accuracy and, consequently, risk management practices.

1.3 Goal

The overarching goal of this thesis is to apply Machine Learning (ML) to develop a more advanced predictive model that offers improved accuracy in predicting blast-induced ground vibration outcomes, compared to the current industry-standard model.

This ML model aims to overcome the limitations of existing methods, which are restricted by their reliance on a limited range of variables. By harnessing the capabilities of ML, the study seeks to incorporate a broader and more comprehensive set of influential variables into the prediction process.

The objectives of this thesis are to explore, implement, evaluate, and identify a specific model that not only demonstrates superior predictive accuracy but also can be integrated into existing mining operations. This integration aims to foster advancements in blast design and risk management, ultimately enhancing overall industry practices.

1.4 Methodology

This thesis explores the contribution of Machine Learning in blast operations, and the prediction of blast-induced ground vibration using a multifaceted research approach by combining both quantitative and qualitative methodologies.

Qualitative research predominantly draws upon existing knowledge within the field to thoroughly examine the subject matter, enriching the development of theories, products, and innovations based on this knowledge. In contrast, quantitative research is oriented towards experimentation and empirical testing to establish theories and principles. Qualitative research primarily seeks to formulate hypotheses, while quantitative research aims to rigorously assess the validity of these hypotheses [11, 22].

The methodological choices are compatible with the overarching project objective, which focus on the development of a system prototype. A prototype is a sample implementation of a system, that provides limited and main functional capabilities of a proposed system. The development and has been guided by the principles of:

- The Design Science Method.
- Task Force Approach.
- Prototyping Methodology.

The utilization of a design science method entails a structured and systematic approach to the creation and evaluation of innovative systems. In the context of this project, this method serves as the foundational framework through which the system prototype is conceived, designed, implemented, and assessed. With an active engagement of a task force, the project benefits from a dedicated group of experts assembled to collaborate on various aspects of the research

[11, 22].

The task force plays a vital role in the research process by providing specialized knowledge, valuable insights, and evaluation of the prototype. The research approach employed in this project integrates quantitative and qualitative methods within the framework of the design science method. These methodological choices are carefully tailored to support the overarching project goal of developing an innovative system prototype [11, 22].

1.5 Contribution

The main contribution of this thesis is to research the possibilities for Machine Learning in blast operations. More specific the contributions include

- A discussion of the possibilities and limitations of Machine Learning being used with blast operations, and assumptions of beneficial use in relation to current industry method.
- A proposed system design, for enabling the use of all the features in a dataset and all the features noted around a blast operation.
- A prototype of a Machine Learning model, that takes into account the entire dataset with all of its features that may impact on the calculated output. The system provides an implementation of a Machine Learning model.

1.6 Stakeholders

In this research, understanding the roles and interests of the stakeholders is key to appreciating the context and impact of this study. Two primary stakeholders are central to this thesis: The Arctic University of Norway (UiT) and a mining consultancy company whose identity remains confidential as per their request.

The Arctic University of Norway (UiT), specifically the Faculty of Science and Technology, and Department of Computer Science played a critical role in this research, providing academic support, resources, and a platform for intellectual inquiry. Their interest lies in contributing to the advancement in predictive modeling techniques. The success of this study aligns with the university's objectives of academic excellence and innovation.

In line with the confidentiality agreement, the second key stakeholder is referred to as 'a Mining Consultancy Company.' Their anonymity is maintained throughout this thesis to respect their privacy and commercial interests. This company has been pivotal in shaping the research's direction, providing the initial topic, a comprehensive dataset, and practical insights into the industry's needs. Their expectation is to gain valuable knowledge from outcomes of this study, with the potential of re-shaping their approach to predictive modeling for future blasting projects.

The thesis follows certain requirements in agreement with the Mining Consultant Company.

Requirements

- **Data Quality and Quantity:** Access to high-quality dataset that covers relevant blasting parameters.
- **Model Performance:** The model should ideally provide greater predictive capabilities for Peak Particle Velocity (PPV) when compared with the current industry-standard model (USBM equation).
- **Security and Privacy:** The study should ensure confidentiality and integrity of shared data.
- **Evaluation and Reporting:** Clear metrics for evaluating the model's performance.

The partnership between the academic environment of the UiT and the practical, industry-focused approach of the mining consultancy company created a unique collaboration. This collaboration ensured that the research was grounded in real-world applicability while being anchored in academic methodology. The combination of these different stakeholders and their viewpoints is expected to make the study more valuable and useful for both the academic world and the mining industry.

1.7 Limitations

In this thesis, developing a Machine Learning (ML) model for the mining industry faced key limitations. A list of limitations of the thesis is provided below:

- **Scope of Expertise:** The task force's collective expertise might not fully cover all the specialized areas needed for ML development and mining industry applications. This may limit the depth of analysis or the sophistication of the ML model.

- **Time Constraints:** Master's thesis projects have strict deadlines. Collaborating with a task force required coordinating schedules, which was challenging through-out this study, and lead to some delays. These delays were predominately associated with receiving the compiled dataset required to advance the study.

- **Data Availability and Quality:** Access to high-quality, relevant data is crucial for ML. This was limiting as the mining industry has strict restrictions on sharing data due to confidentiality or proprietary concerns. The provided data set was restrictive in the amount of variables being included, due to confidentially reasons.

- **Generalizability of the Model:** The model developed may be tailored to specific conditions of the collaborating company or dataset, which may not be generalizable to other scenarios or mining operations.

- **Regulatory and Ethical Considerations:** In the mining industry, regulatory compliance is crucial [16]. The thesis might not fully address the complex regulatory landscape, especially regarding the use of AI and ML.

- **Intellectual Property and Publication Restrictions:** Due to the dataset being filtered for sensitive information prior to being shared, there is no restrictions applied to the publication of this study.

- **Expectation Management:** Aligning the academic goals of a thesis with the practical objectives of the industry partner can be challenging. There might be a gap between academic exploration and industry applicability.

In this thesis, the use of Machine Learning (ML) models are explored and developed in the context of the mining industry. While these models offer innovative solutions, they also present inherent limitations, which are critical to acknowledge for a comprehensive understanding of their capabilities and constraints, and is listed below:

- **Model Complexity and Interpretability:** ML models, especially deep learning networks, can become highly complex, making them difficult to interpret. This can lead to challenges in understanding how decisions are made, which is crucial in mining where safety and precision are paramount.

- **Data-Driven Nature of ML Models:** ML models heavily rely on the quality and quantity of data available. In mining, the variability and inconsistency of data can affect the model's accuracy and reliability.

- **Transferability and Scalability:** While ML models can be highly effective in specific scenarios, their ability to generalize across different mining environments is limited. A model trained on data from one mine may not perform well in another due to geological and operational differences, raising concerns about its scalability and transferability.

- **Real-World Application vs. Theoretical Modeling:** There often exists a gap between theoretical modeling and real-world applications. Models that perform well in simulated environments may not yield the same results under actual mining conditions, due to unforeseen variables and complex interactions not accounted for in the model.

- **Technological Constraints:** The implementation of ML models in mining is constrained by the available technology. Limitations in computational resources, data storage, and processing capabilities can restrict the complexity and efficiency of the models.

1.8 Outline

This thesis contributes to an improvement in prediction of blast-induced ground vibration by considering more than the two variables used to predict PPV in the industry today. It provides the design, implementation and evaluation of a deep learning model to predict blast-induced ground vibration.

The remainder of this thesis is structured as follows:

Chapter 2 provides an exploration of the technical background underlying this thesis. It encompasses essential blast-induced ground vibration fundamentals, gives an description of considerations related to designing a blast event, and goes on to describing Peak Particle Velocity and its application and implication. Lastly, the chapter looks at models currently used in the industry.

Chapter 3 provides descriptions of an Artificial Neural Network and Deep Neural Network and the steps of creating a Deep Neural Network model, involving the description of techniques for training and validating the model. This chapter provides a summary and in depth description of related work that has succesfully used Machine Learning and Artificial Neural Network in mining operations.

Chapter 4 outlines the methods and methodological framework of the research, detailing the chosen research methods, their theoretical foundations, the dataset utilized, and specifics of the implementation. It describes the chosen research approach, the research strategies, data collection methods, and the Machine Learning frameworks.

Chapter 5 presents the system's design, requirements and architectural blueprint, delineates the functional and non-functional requirements, and establishes the criteria for evaluating the system's performance and effectiveness, as relevant to the objectives of this thesis.

Chapter 6 details the implementation of the prototype of a Machine Learning model. It describes the steps of implementing a Deep Neural Network model utilizing a dataset compiled from multiple mining operations.

Chapter 7 provides a comprehensive account of the experiments conducted using the Machine Learning models. It details the experimental setup, the methodology followed for testing the models, and the specific metrics used for evaluation. This chapter also includes a thorough analysis of the experimental results, highlighting the model's performance in various scenarios.

Chapter 8 covers the validity and reliability of the research findings, as well as presenting performance metrics and compares results towards related work. The chapter presents interpretation of results, discusses strengths and weaknesses of the Deep Neural Network by presenting analysis of the model and patterns of error. This section reflects on the thesis objectives by demonstrating the Deep Neural Network's contribution to the field.

Chapter 9 reflects on the encountered challenges and obstacles throughout the research process. This chapter provides insights into the practical difficulties, theoretical limitations, and unexpected findings that emerged during the study.

Chapter 10 summarizes the thesis, recapping the key findings, contributions to the field of Machine Learning in mining, and the implications of this research. It also provides a critical reflection on the research objectives and the extent to which they were achieved. The chapter proposes directions for future research, building on the findings of this thesis. It suggests potential areas for further exploration and improvements, both in terms of the methodology and application of Machine Learning in the mining industry.

2

Background

This chapter details the mining sector, focusing on blasting operations and their environmental impact. It outlines the steps of a blast event and the crucial role of blasting engineers in managing both controllable factors and uncontrollable factors. It includes the critical aspects of designing a blast event, and the effects of the blast in blast-induced ground vibrations. It categorizes these blast factors into those that can be controlled and those that cannot, illustrating the complex nature of managing blast-induced vibrations.

This chapter introduces Peak Particle Velocity (PPV) as the primary metric for measuring blast-induced ground vibrations, and describes its relevance in industry standards and environmental assessments.

It then examines the traditional industry method for predicting PPV, which uses linear regression, and an empirical formula.

2.1 Blasting Operations

The operation of blasting is fundamental in the mining industry, serving as the primary method for rock fragmentation, which is crucial for enabling various subsequent activities like excavation and material removal in mining projects.

Blasting operations employ the controlled use of explosives to induce fractures in rock masses, thereby facilitating the desired level of fragmentation and displacement of materials. While effective, this process is not without its challenges [40].

The controlled detonation of explosives releases a significant amount of energy. While a large portion of this energy is used for the intended purpose of breaking the rock, there is often a residual release of energy that can manifest in various forms. One of the most significant environmental implication associated with blasting is referred to as blast-induced ground vibration.

Blast-induced ground vibrations transmit through the ground over distances, affecting not only the immediate blast site but also surrounding ecosystems, human settlements, and structures. Blast-induced ground vibrations could lead to structural damage in nearby buildings, and poorly controlled blasts may result in the unnecessary use of explosives, which further intensifies environmental issues and elevates operational costs [40].

2.1.1 Environmental Consequences

The environmental consequences that originates from blasting operations have received significant research attention. Research by Bakhtavar et al. (2021), Hosseini et al. (2021, 2022), Armaghani et al. (2018) and Nguyen et al. (2020) have extensively explored these implications [9, 28, 27, 6, 49, 8, 32]. They found that unintended blast-induced vibrations holds the potential to cause structural damage and to compromise the stability of mining pit walls, thereby presenting substantial risks to peoples safety, infrastructure, groundwater, railways, highways, heritage sites, and nearby communities [41, 32, 35, 18].

These findings emphasizes that the need for precise control and accurate prediction of blast outcomes is not only a matter of operational efficiency, but also of environmental protection and community safety. It is within this context that the potential of Machine Learning (ML) algorithms to improve the accuracy of blast-induced effects is explored.

2.1.2 Controllable And Uncontrollable Blasting Variables

Research has identified a number of factors influencing blast-induced ground vibration during blasting operations. These factors can be broadly categorized as controllable, falling within the domain of blast engineers, and uncontrollable, encompassing geological parameters beyond human control [45, 24].

Controllable factors encompass a range of blast design parameters and explosive characteristics, allowing engineers to tailor blast designs to specific requirements. In contrast, uncontrollable factors, such as site location, and geological features like rock strength, rock structure, ground water presence, pose inherent challenges in predicting and mitigating blast-induced ground vibration [5, 25].

In the mining industry, understanding controllable factors of blast design and uncontrollable factors such as rock strength, is crucial for optimizing blasting operations. Machine Learning can potentially analyze these variables to identify influential patterns, enhancing efficiency and safety. This would be particularly useful for risk mitigation and environmental compliance, as Machine Learning could predict the impacts of various blasting scenarios by adapting to the complex, variable conditions typical in mining, learning from high-dimensional data. A Machine Learning's predictive analytics and continuous learning capabilities is believed to be beneficial for site-specific customization and operational decision-making in mining.

2.1.3 Designing A Blast Event

A blasting engineer is one who is responsible for designing a blast, and must carefully consider the design variables which can influence the magnitude of blast-induced ground vibrations [38]. By having an improved understanding of how the design variables contribute to the magnitude of blast-induced ground vibrations, the blasting engineer is able to make more informed decisions to aid risk assessments. As well as producing blast designs that meet specific requirements, while minimising the environmental footprint of a blasting activity. Below is a more detailed description of some considerations a blasting engineers must consider at the design stage:

- **Safety:** Ensuring the safety of personnel, equipment, and nearby communities is the foremost concern in any blasting operation. By meticulously controlling factors such as blast design, charge size, and initiation sequence, blasting engineers can reduce the risk of accidents related blast-induced ground vibrations.

- **Structural and Environmental Protection:** Blasting near existing structures or environmentally sensitive areas demands precise control over blast parameters. Effective blast design helps safeguard these elements by limiting blast-induced ground vibrations, ground displacement, and other potential disturbances.

- **Design Optimization:** Blasting engineers are challenged in the need to balance multiple, often conflicting, objectives such as minimising total blast cost, maximising material recovery, minimising environmental impacts, and optimising rock fragmentation distribution.

- **Economic Efficiency:** Controlling factors in blast design contribute to cost-effectiveness in mining operations. By optimizing blast parameters, engineers can reduce drilling and blasting costs, minimize secondary breakage, and improve fragmentation. This leads to lower operational expenses and increased profitability.

- **Environmental Responsibility:** Modern mining operations are under increasing scrutiny regarding their environmental impact. By carefully considering specific design variables, blasting engineers can mitigate adverse effects related to environmental disturbance.

- **Regulatory Compliance:** Compliance with local, national, and international regulations is mandatory for mining operations. Ensuring blasts are designed to adhere to legal limits on blast-induced ground vibrations, airblast levels, and other environmental parameters.

- **Community Relations:** Maintaining a harmonious relationship with neighboring communities is vital for the mining industry's social license to operate. Well-designed blasts that minimize noise, dust, and vibrations reduce disturbances to nearby residents. This, in turn, helps build trust and goodwill within the community.

There are specific fundamental components of a blast event that are required. These components are listed below:

- **Drill Holes:** Precisely drilled into the rock, these holes serve as the containers for the explosives. The drill hole size, pattern spacing, are all important for the blast's efficiency, and impact the blast-induced ground vibration response.

- **Explosive Products:** Depending on the type of rock and the intended outcome, various explosive materials are used. These are carefully inserted into the drill holes. Different explosive types can influence the amount of energy transmitted to fracturing the surrounding rock mass.

- **Capping or Stemming:** To focus the explosive energy to fracturing the rock, the blast holes are typically sealed with stemming material, like sand or crushed aggregate rock after the explosives have been loaded into the hole.

- **Detonation:** A detonator placed within the explosive product of each blast hole initiates the explosion, causing a nearly simultaneous detonation of the explosive within the blast hole. The timing of each blast hole detonation is a design variable which is considered by the blasting engineer, and impacts on the propagation of the blast-induced ground vibration generated [7].

- **Blast Pattern:** The arrangement and sequencing of the multiple drill holes contribute to the direction and extent of rock fragmentation [26].

2.2 Blast-Induced Ground Vibration

Blast-induced ground vibration is considered one of the most undesirable effects of surface mining blasting operations. Blast-induced ground vibrations are especially relevant in mining and civil operations where blasting activities are common. These vibrations pose potential risks to sensitive locations, including buildings, natural heritage sites, and other structures. When explosives are detonated within a blasthole, they undergo a rapid chemical reaction that produces a tremendous release of energy in the form of a shockwave. This shockwave initiates the vibration waves, which then propagate outward from the blast site. A multitude of factors contribute to the characteristics and impact of blast-induced ground vibrations. These encompass blast design elements, the properties of the explosives used, the distance from the blast site, as well as the prevailing geological condition [67, 39, 59].

2.2.1 Peak Particle Velocity

The key parameter to evaluate the vibration impact in mining operations, and considered a measurement to both predict and control blast-induced ground vibrations, is known as Peak Particle Velocity (PPV) [13]. In the domain of mining, construction, and various technical engineering applications, PPV is commonly used to quantify the intensity of blast-induced ground vibrations caused by activities such as blasting.

PPV is expressed in terms of velocity (often in millimeters per second, mm/s) and represents the highest particle velocity reached by any particle in the ground during a vibration event. It is considered as a crucial metric because it is directly related to the potential for damage to structures and is often used in regulations and guidelines for blasting and other activities that produce blast-induced ground vibrations [56, 46].

2.2.2 Initial Predictor Of Peak Particle Velocity

The initial notable predictor of Peak Particle Velocity (PPV) was conceived through collaborative efforts of the United States Bureau of Mines (USBM), Duvall, and Fogleson, and is still used as the industry standard predictor for PPV.

PPV is a measurement of the velocity of the most forward particles in terms of transverse (T), vertical (V), and longitudinal (L) velocities [19]. The technique is mathematically expressed in the Equation 8.1, and can be viewed below.

$$\text{Duvall and Fogleson (USBM)} \quad PPV = K(\frac{D}{Q^{1/2}})^{-b} \tag{2.1}$$

$$\text{Scaled Distance} \quad SD = (\frac{D}{Q^{1/2}}) \tag{2.2}$$

Where:

- PPV is the Peak Particle Velocity, typically measured in millimeters per second (mm/s).
- K is the site constant value, representing the geological and environmental characteristics of the blast site. It is derived by linear regression.
- Q is the Maximum Instantaneous Charge (MIC), the maximum charge per delay, measured in kilograms (Kg).
- D represents the distance from the blast source to the monitoring point, measured in meters (m).
- SD is the Scaled Distance, measured in kg/m$^{0.5}$ (mm/s), is a ratio that standardizes the distance from a blast site relative to the amount of explosives used.
- b is the attenuation constant value, which represents how the wave's energy decreases with distance. It is derived by linear regression.

PPV is used extensively in mining operations for several reasons. These being:

- **Safety and Structural Integrity:** High levels of blast-induced ground vibration can cause damage to nearby structures, both above and underground. Predicting and controlling PPV helps in maintaining the safety of these structures.

- **Regulatory Compliance:** Many regions have legal limits on blast-induced ground vibration levels to protect surrounding communities and infrastructure. Accurate prediction of PPV can assist in maintaining compliance with these regulations.
- **Blast Optimization:** Understanding and controlling PPV helps in optimizing the blasting process. This optimization can lead to more efficient fragmentation, and reduce environmental impact.
- **Community Relations:** Excessive vibration can be a nuisance or cause concern for nearby communities. Managing PPV levels is essential for maintaining good community relations and minimizing complaints.
- **Environmental Conservation:** PPV is also important for the protection of sensitive environments. Improved prediction in PPV helps in assessing the potential impact of blasting activities on nearby wildlife and ecosystems [19].

2.2.3 Measurement And Representation

Mining engineers, geotechnical experts, and construction professionals rely on PPV measurements to determine the safety of nearby structures, evaluate the potential for structural damage, and formulate blast design strategies that adhere to permissible limits.

PPV is meticulously measured through the use of specialized instruments known as seismographs or blast-induced ground vibration monitors. These devices are strategically positioned within the surrounding area of the blasting activity to capture the precise ground motion data. The output from these instruments, typically in the form of time-series data, is subsequently analyzed to recognize the peak amplitude, which corresponds to the highest instantaneous particle velocity achieved during the blast event [56, 46].

Blast-induced ground vibration monitors use a transducer, also called a geophone, to measure vibrations. To properly measure these vibrations, the geophone is securely coupled to either the ground or a structure. This ensures that the geophone sense all the energy in the ground [25, 40].

2.2.4 Multidimensional Characterization

One distinctive aspect of PPV is its capacity to provide a multidimensional ground motion characterization. This metric accounts for particle velocity along three orthogonal axes: transverse (T), vertical (V), and longitudinal (L).

These components collectively offer a comprehensive description of how ground particles oscillate in space as they respond to the propagating wavefront, thereby enhancing the precision of assessment [19].

2.2.5 Regulatory And Industry Standards

Peak Particle Velocity (PPV) is a universal recognized benchmark metric used in the mining industry, featuring prominently in various regulatory frameworks and guidelines that govern blasting activities globally. These frameworks typically establish criteria for acceptable PPV levels. Monitoring and controlling PPV is often mandated to ensure that blasting activities fall within acceptable safety and environmental parameters [13].

2.3 Industry Standard Prediction Method

The USBM empirical equation, as outlined in 8.1, is the prevailing industry standard for predicting Peak Particle Velocity (PPV) resulting from blasting operations. This equation fundamentally incorporates two key variables: the Maximum Charge per delay (MIC) and the distance between the blast site and the measurement point. These two variables combined to quantify a third parameter being Scaled Distance (SD) which is described in equation 2.2.

The equation's efficacy depends on the accurate determination of the two constants, K and b, derived from the statistical analysis model linear regression. Linear regression analysis is used to predict the value of a variable based on the value of another variable.

Constant K refers to the site constant value that represents the geological and environmental characteristics of the blast site, and constant b refers to the attenuation constant value, representing how the wave's energy decreases with distance.

These constants are more than numerical placeholders; they embody the complex interplay of geological variability and the inherent uncertainties associated with design variables and operational conditions in blasting scenarios. These constants are derived through regression fitting process as previously mentioned, ensuring that they reflect the specific geological and operational context of each blasting site as accurately as possible [43].

In essence, K and b are not static values but are dynamically derived to adapt the USBM equation to vary geological settings and blasting conditions. This adaptability underscores the USBM equation's widespread acceptance and utility in the industry, providing a tailored approach to predicting PPV for diverse blasting scenarios.

2.3.1 Linear Equation

The essence of linear regression lies in establishing a linear equation of the form:

$$Y = a * X + b \tag{2.3}$$

In this equation:

- Y - Dependent Variable
- a - Slope
- X - Independent Variable
- b - Intercept

The coefficients a (slope) and b (intercept) are directly interpretable [70]. The slope a indicates the change in the dependent variable Y for a one-unit change in the independent variable X. The slope and intercept are derived based on minimizing the sum of the squared difference of distance between data points and the regression line, and the model gets the best regression fit line by finding the best a and b values.

2.3.2 Liner Regression

Linear regression is often used to model or analyze data for estimating the relationship between a set of independent and dependent variables [70, 31]. Some of the benefits of linear regressions to measure Peak Particle Velocity (PPV) in the field of mining is listed below:

- **Simple Mathematical Formulation:** Its formula is simple with one independent variable and an extension for multiple independent variables, making the linear equation easy to compute and interpret.
- **Minimal Assumptions:** Linear regression requires fewer assumptions compared to more complex models. The primary assumptions are linearity, independence, constant variance of errors, and normality of error terms.

- **Effectiveness in Establishing Relationships:** Linear regression is effective at identifying and quantifying the relationship between a value and various influencing factors. In regards to measuring PPV, this would involve the relationship between PPV and influencing factors such as the amount of explosive used, distance from the blast, and geological conditions.
- **Computational Efficiency:** Linear regression models can be fit quickly, even with large datasets, due to their mathematical simplicity. This efficiency is a contrast to more complex models, which often require significantly more computational resources and time.
- **Data Availability:** Mining operations often collect data that linearly correlates with PPV, such as blast design parameters and geological data. Linear regression can effectively utilize this available data to make reliable predictions.
- **Historical Precedence:** Linear regression has a long history of application in various fields, including mining. Its proven effectiveness over time has made it a go-to method for predictive modeling in this industry.
- **Ease of Interpretation:** The results of a linear regression model are easy to interpret, making them user-friendly for engineers and decision-makers who may not have extensive statistical training.

2.3.3 Linear Regression Analysis

Linear regression is an algorithm extensively applied to predict target outputs from input variables, assuming a linear relationship between them [44]. The regression line is defined by its slope and intercept, which are derived mathematically to minimize the sum of the squared differences between the observed values and the values predicted by the model.

A linear regression technique creates a link between independent and dependent variables by determining the most suitable line, often referred to as the line of best fit. This line illustrates how the dependent variable is expected to change in response to the independent variable.

3

Artificial Neural Network

This chapter explores the technical aspects of Machine Learning (ML) and Artificial Neural Networks (ANNs), focusing on the development and application of algorithms essential for predicting Peak Particle Velocity (PPV). It provides an insightful overview of the Machine Learning pipeline and references relevant studies to highlight challenges and variances in ML modeling approaches.

The chapter begins with an introduction to ML models, emphasizing their ability to learn from data and make predictions autonomously. It further explains how these models recognize patterns, adapt to changes, and enhance their performance with exposure to new data. The chapter emphasizes the significance of ML in predicting blast-induced ground vibrations, noting the models' capacity to autonomously learn from historical data and uncover underlying patterns crucial for accurate predictions.

ANNs are introduced as a subset of ML models, where Deep Neural Networks are an extension of ANNs. The chapter explains their structure, inspired by biological Neural Networks, comprising interconnected nodes organized into input, hidden, and output layers. The chapter explains activation, optimization and loss functions, and their relevance for Deep Neural Networks, as well as presenting their mathematical formulas. The chapter concludes with an overview of related work in the field, summarizing key findings, methodologies, and algorithms from various studies that have informed this thesis's approach to ML in mining operations.

3.1 Machine Learning Models

Machine Learning (ML) focuses on developing algorithms and models to be capable of learning from data and make decisions or predictions without being explicitly programmed. Machine Learning systems, also called Machine Learning models, can solve complex problems and extract valuable insights from the data it is given. The models are capable of recognizing patterns, adapt to changing conditions, and improve their performance over time as they are exposed to new and more data [44].

A Machine Learning model is a mathematical representation of a real-world process based on data, and can serve as a critical component in the pursuit of predicting blast-induced ground vibrations with accuracy and precision. These models possess the ability to autonomously learn from historical data, to capture underlying patterns or relationships, enabling the mapping of inputs to outputs in a manner that is essential to the research goal [44].

3.2 Artificial Neural Network

An Artificial Neural Network (ANN) is a subset of Machine Learning, inspired by the structure and function of biological neural networks. ANNs consists of interconnected neurons (nodes) which are simplified computational models that receive and process information. An ANN is structured into layers: an input layer, one or more hidden layers, and an output layer. The input layer receives the initial data, the hidden layers process the information, and the output layer produces the final result [20].

They layers are created and connected together. When the network is asked to solve a problem, it attempts to do so over and over, each time strengthening the connections that lead to success and diminishing those that lead to failure. An ANN is considered a tool for creating predictive modeling since it consist of interconnected nodes that collectively learn complex patterns from data.

3.3 Deep Neural Network

A Deep Neural Network (DNN) is an extension of an Artificial Neural Network (ANN) that includes more hidden layers between the input and output layers than an ANN. These additional layers enable the network to learn more complex and abstract features in the data [42].

DNNs are particularly useful for handling large and complex data sets, though they require more computational resources and are prone to overfitting if not properly regulated. Overfitting occurs when the model cannot generalize and fits too closely to the training dataset instead.

Each neuron in a DNN performs a linear transformation on its inputs. With an input vector x, each neuron has a weight vector q and a bias term b, the linear transformation is represented as:

$$z = \mathrm{w}^T \mathrm{x} + b \tag{3.1}$$

$$z = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b \tag{3.2}$$

- z represents the weighted sum of the inputs plus the bias.
- w_i represents the weight associated with the i-th input.
- x_i represents the i-th input from the previous layer.
- b represents the bias term.

The formula 3.1 uses vector notation and is more favored when dealing with high-dimensional data or the need to emphasize the operations in a generalized form. In this context, z denotes the calculated value for a neuron prior to the application of an activation function. This value is the sum of the products of each input x_i as the input vector, and its corresponding weight w_i that represents the invert of the weight vector, along with a bias term b. The operation w^T is the dot product between the two vectors, which is a compact way of expressing the weighted sum of the inputs.

Both formula achieve the same results, but formula 3.2 expands the vector notation into its full scalar form. It sums the product of each weight w_i with its corresponding input x_i and then adds the bias b. This linear transformation is performed across all neurons within a layer. In neural networks, vector notation is used since it aligns with how these computations are implemented in libraries like TensorFlow, which are optimized for vector and matrix operations.

3.3.1 Weights And Biases

In a Deep Neural Network, weights and biases are fundamental components that determine how the input data is transformed and processed through the network layers [62]. The components are described below:

- **Weights:** These are the parameters within the network that are adjusted during training. Each connection between neurons in different layers has a weight associated with it. In a Deep Neural Network, weights represent the strength or importance of the connection between neurons.

- **Biases:** A bias is an additional parameter in a Deep Neural Network which is used along with the weighted sum of inputs to a neuron. It is a constant value for each neuron that is added to the product of inputs and weights before the activation function 3.3.11.

Weights and biases are learnable parameters of a Deep Neural Network, and adjusted through the training process to minimize the difference between the actual output of the network and the desired output. Effectively 'learning' from the training data. The appropriate tuning of weights and biases allows the network to model complex relationships between inputs and outputs, making accurate predictions [62].

Updating weights and biases involves:

- **Forward Propagation:** For each input, the network computes the output by applying weighted sums and activation functions. During the forward pass, each neuron computes its output which is mathematically is represented here 3.6.

- **Loss Calculation:** The difference between the network output and the true value is calculated using a loss functions, the specific functions can be found in section 3.3.14.

- **Backpropagation:** Begins after the loss has been computed, and involves calculating the gradient, also known as the derivative, of the loss function with respect to each weight and bias. This is achieved by applying the chain rule of calculus, and its mathematical formula shown can be found here 3.3.8.

- **Gradient Descent:** The weights and biases are updated in the direction that reduces the loss. The optimization algorithm used is called Adam, with its mathematical formulas found here 3.11.

3.3.2 Generalization

Generalization refers to the ability of a Neural Network to perform well on new, unseen data, not just the data it was trained on. Generalization determines how well the network can apply its learned knowledge to different scenarios.

A model that generalizes well can accurately interpret and predict outcomes for data that it has not yet encountered, indicating robustness and reliability.

Poor generalization, often due to overfitting, means the model performs well on its training data but fails to predict accurately on new data. Achieving generalization is possible through regularization techniques known as L1- and L2 Regularization, dropout, and early stopping, to prevent overfitting. Generalization can be achieved with using diverse and representative training data, and with the use of cross-validation techniques to ensure the models performance is consistent across different subsets of the data [10].

3.3.3 Regularization

Regularization techniques in Deep Neural Networks are strategies used to prevent overfitting. Overfitting occurs when a model learns the training data by capturing the underlying patterns and the random fluctuations, and performs poorly on unseen data. The goal of regularization is to improve the model's generalization ability, ensuring it performs well on new, unseen data. Regularization is used to prevent overfitting by penalizing complex models and encouraging the learning of simpler models that generalize better. Is is used to improve generalization, by constraining the learning capacity of the model, regularization ensures that the model does not learn the noise and specifics of the training data but captures the underlying trends [63].

Common Regularization Techniques:

- **L1 Regularization (Weight Decay):** L1 regularization adds a penalty equal to the absolute value of the magnitude of coefficients. It can lead to sparse models where some weights become zero.

- **L2 Regularization (Weight Decay):** L2 regularization adds a penalty equal to the square of the magnitude of coefficients. It encourages weight values toward zero but not exactly zero.

3.3.4 Overfitting

In the context of Machine Learning, overfitting occurs when a model captures the underlying patterns in the training data, and the noise and random data fluctuations [44, 63]. This will result in a model that adapts to irrelevant details and random error in the training set, while also learning the useful information.

An overfitted model may show excellent evaluation metrics on the training data, but its evaluation results will significantly drop when it encounters real-world data. This is because the real-world data will likely not have the same noise characteristics as the training set. Overfitting can lead to misleading results when making predictions about data, since it might make decisions based on irrelevant features that happened to correlate with the target variable in the training set but do not have a causal relationship.

Overfitting can occur due to different reasons listed below:

- **Learning the Noise:** In a dataset, there are generally two components, the actual underlying pattern and random variations or errors (noise). A model that is overfitting learns to reproduce these random variations as if they were significant patterns.
- **Too Complex Models:** Overfitting is often a result of a model being too complex relative to the simplicity of the data. Such models have too many parameters and are capable of learning intricate details and patterns.
- **Loss of Generalization:** The primary goal of a Machine Learning model is to make accurate predictions on new, unseen data, known as generalization. When a model is overfitted, it performs well on the training data, because it has effectively memorized it, but fails to predict accurately on new data because the intricate details it learned are specific to the training set and don not apply to other data.

3.3.5 Dropout

Dropout is a regularization technique used in Neural Networks, particularly Deep Neural Networks to prevent overfitting [10] Dropout is applied during the training of a neural network and works by randomly "dropping out" or deactivating a subset of neurons in a layer during a forward propagation, effectively making the network smaller. This randomness forces the network to learn more robust features that are useful in conjunction with many different random subsets of the other neurons.

The importance of Dropout is:

- **Reduce Complexity:** Dropout reduces the complexity of the model by preventing units from collaborating. Since a neuron cannot rely on the presence of other neurons, it outputa features that are generally useful.
- **Learning Method:** It simulates a powerful ensemble learning method, similar to training a large number of neural networks with different architectures in parallel.
- **Scaled Down:** During testing and in actual use, which is after training,

dropout is not used; instead, the full network is utilized for performance. The weights of the neurons are typically scaled down by the dropout rate to balance the larger number of active neurons compared to the training phase.

3.3.6 Early Stopping

Early stopping is a regularization technique used in training Deep Neural Networks to prevent overfitting. Early Stopping works by stopping training as soon as the performance on a validation set starts to deterorate, rather than continuing to train until the iteration limit is reached. It involves monitoring the model's performance on a validation dataset during the training process. If the model's performance begins to reduce, which leads to the validation error starting to increase, it indicates that the model is starting to overfit to the training data [60].

The primary goal of early stopping is to stop the training process at the point where the model performs best on unseen data. By halting the training before the model becomes too specialized to the training data, early stopping ensures that the model maintains generalization capability, and making it capable of performing well on new, unseen data.

3.3.7 Feedforward Neural Network

A Feedforward Neural Network (FNN) is a basic form of neural network where connections between the nodes do not form a cycle [44]. It is comprised of an input layer, one or more hidden layers, and an output layer, where information moves in only one direction, forward, from the input nodes, through the hidden nodes, and to the output node. Feedforward Neural Networks are often used when the task involves a straightforward mapping of input to output, like in regression task.

3.3.8 Backpropagation

Backpropagation is a technique for training feedforward Neural Networks. The process begins by comparing the network's predicted output with the desired output, identifying inconsistency as an error rate. This error rate informs the adjustment of weights for each neuron. Backpropagation systematically works its way backward through the network, updating weights in each layer based on the calculated error [23]

Backpropagation computes the gradient of the loss function with respect to the output of the network. It then proceeds to calculate gradients with respect to the parameters (weights and biases) of each layer. Its mathematical formula is shown here:

$$\frac{d}{dx} f(g(x)) = f'(g(x)) \cdot g'(x) \tag{3.3}$$

The output of each neuron is the result of applying an activation function f to the linear combination of inputs $g(x)$, forming a composite function. The output of each layer is a function of both the linear transformation and the subsequent non-linear activation, so the gradient of the loss with respect to the weights and biases contains derivatives of both these components. The forward and backpropagation processes are repeated iteratively across multiple iterations to minimize the loss function.

3.3.9 Forward Propagation

Forward propagation is the mechanism where a neural network processes input data to generate predictions [23]. The input layer takes raw data, and each neuron from the layer corresponds to one input feature. These inputs are then transformed by a series of weighted sums and biases as they pass through the network's layers.

The general linear combination formula of inputs in forward propagation is:

$$g(x) = \sum_i (w_i \cdot x_i) + b \tag{3.4}$$

Here, w_i are the weights, x_i are the input values to the neuron, and b is the bias. This builds on the mathematical formula for backpropagation 3.3 with linear transformation that occurs at each neuron in a layer of a neural network.

$$f(g(x)) \tag{3.5}$$

This equation represents the application of an activation function f to the result of the linear transformation $g(x)$. The activation function named ReLU 3.8 is applied to introduce non-linearity, before the process continues across all layers until reaching the output, where the network reaches its final prediction. Forward propagation involves no learning, it is used only for inference learning as it occurs during the backpropagation phase, which adjusts the network based on prediction errors.

This equation is specially for a layer l in a neural network, and represents the combined input to the neurons in layer l after applying the weights and biases. Each layer l in a neural network a linear transformation is performed on its inputs, and is represented as:

$$z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)} \tag{3.6}$$

Where:

- $z^{(l)}$ is the input to the activation function in the l^{th} layer (weighted sum of activations from the previous layer plus bias).
- $W^{(l)}$ is the weight matrix associated with the l^{th} layer.
- $a^{(l-1)}$ is the activation vector from the $(l-1)^{th}$ layer (outputs of the neurons in the previous layer).
- $b^{(l)}$ is the bias vector for the l^{th} layer.

The results of the linear transformation $z^{(l)}$ is passed through a non-linear activation function to produce $a^{(l)}$, the output of layer l.

3.3.10 Supervised Learning

Supervised learning is a Machine Learning model approach, that requires labeled input and output data during the training phase of the Machine Learning model. The vast majority of available data is unlabelled and raw data. Supervised Machine Learning is used to classify unseen data into established categories and forecast trends and future change as a predictive model. A model developed through supervised Machine Learning will learn to recognize objects and the features that classify them. Supervised Machine Learning models can predict outcomes from new and unseen data by learning patterns between input and output data [4].

3.3.11 Activation Functions

The main purpose of an activation function is to introduce non-linearity into the output of a neuron. This non-linearity allows the network to model complex relationships in the data. Without non-linearity, a Neural Network, regardless of how many layers it has, would behave like a linear regression model, limiting its ability to capture complex patterns. Activation functions in Neural Networks are mathematical equations that determine the output of a Neural Network. The function is attached to each neuron in the network and determines if the neuron should be activated or not. This decision is based on the input of each neuron and if the input is relevant for the model's prediction [2].

The formula of the activation function is:

$$a = \sigma(z) \tag{3.7}$$

where σ is the activation function.

To introduce non-linearity, an activation function σ is applied to the linear transformation result z, of the linear transformation formula for Deep Neural Networks, found here 3.1.

ReLU, which stands for Rectified Linear Unit, is a type of activation function widely used in Deep Neural Networks. ReLU is computationally efficient because it involves simple thresholding at zero. The mathematical formula for ReLU is:

$$\text{ReLU}(fx) = max(0, x) \tag{3.8}$$

The gradient of ReLU is constant for positive inputs, and therefor allows deeper networks to be trained more effectively. The ReLU function and its derivative are monotonic, which involves the function to return 0 if it receives any negative input, and when it receives any positive value x, it returns that value. As a result, the output has a range of 0 to infinite. Since ReLu sets all negative values to zero, a certain proportion of the neuron outputs will be zero, naturally leading to sparsity in the hidden layers of the network [2].

3.3.12 Normalization

Normalization in Deep Neural Networks is a technique used to standardize the inputs of neurons within a network layer [30]. It helps in stabilizing and speeding up the training of Deep Neural networks. The most common form of normalization is Batch Normalization (BN), and its formula is:

$$\text{BN(x)} = \gamma \left(\frac{\text{x} - \mu}{\sqrt{\sigma^2 + \epsilon}} \right) + \beta \tag{3.9}$$

- x represents the input to the batch normalization layer.
- μ is the mean of the inputs in the batch.
- σ^2 is the variance of the inputs in the batch.
- ϵ is a small constant added for numerical stability (to avoid division by zero).

- γ and β are parameters that the model learns.

After normalizing the inputs, γ scales and β shifts them. The normalization could limit the network's ability to represent complex relationships, as it forces the inputs to take a standard distribution. γ and β restore this capability.

The impacts of normalization is described below:

- **Inputs of a layer:** Normalization adjusts the inputs of a layer so that they have a mean of zero and a standard deviation of one. This is achieved by subtracting the mean of the batch from each input and then dividing by the standard deviation of the batch.
- **Efficient training:** This process makes the training more efficient by reducing internal covariate shift, which involves the distribution of each layer's inputs change during training, slowing down the training process and making it harder to tune hyperparameters.
- **Batch Normalization:** It is typically applied after a convolutional or fully connected layer but before the activation function.
- **Initialization:** It helps in reducing the dependence on initialization, it acts as a form of regularization, and can sometimes eliminate the need for Dropout.
- **Learning rates:** It often allows for the use of higher learning rates, which can further speed up training [30].

3.3.13 Optimization Functions

Optimization functions, are algorithms used in the training of Deep Neural Networks to minimize or maximize, a given objective function, typically a loss function. The loss function measures the difference between the predicted output of the network and the actual output. The role of the optimizer is to adjust the weights and biases of the network to reduce this loss [61].

Optimization functions determine how quickly and effectively a Deep Neural Network learns from data, and can significantly speed up the training process and improve the performance of the model. They ensure that the training process converges, meaning that it reaches a point where additional training does not significantly improve the model. Some are better at avoiding local minima and plateaus, common challenges in training Deep Neural Networks.

Optimization functions influence the stability of the learning process and the ability of the model to generalize from the training data to unseen data, and help in avoiding overfitting, where the model performs well on training data but poorly on new, unseen data.

The optimization functions used in this thesis are:

- **Adam (Adaptive Moment Estimation):** which combines ideas from RMSprop and Momentum. It calculates adaptive learning rates for each parameter and is often effective in practice.
- **Stochastic Gradient Descent (SGD):** Finds the model parameters that correspond to the best fit between predicted and actual outputs. SGD randomly picks one data point from the whole data set at each iteration to reduce the computations enormously. Adam optimizer is the extended version of stochastic gradient descent.

In the Stochastic Gradient Descent (SGD) optimization function:

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i)}, y^{(i)}) \tag{3.10}$$

the terms represent the following:

- θ represents the parameters (weights) of the model.
- η is the learning rate, which determines the step size during the optimization process.
- $\nabla_\theta J(\theta; x^{(i)}, y^{(i)})$ is the gradient of the model's loss function J, evaluated at the current parameter θ for a single data sample $(x^{(i)}, y^{(i)})$. This gradient indicates the direction in which the model's weights need to be adjusted to minimize the loss.

Adam (Adaptive Moment Estimation), is an extension of SGD that combines the advantages of RMSprop and Adagrad, and its optimization algorithm can be viewed in 3.11

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \tag{3.11}$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \tag{3.12}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{3.13}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{3.14}$$

$$\theta = \theta - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t \tag{3.15}$$

the terms represent the following:

- θ_t and θ_{t+1} represent the parameter vectors at time steps t and $t + 1$, respectively.
- η is the learning rate, which determines the step size during the optimization process.

- m_t and v_t are estimates of the first moment (the mean) and the second moment (the uncentered variance) of the gradients, respectively.
- $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected versions of m_t and v_t.
- β_1 and β_2 are exponential decay rates for the moment estimates of m_t and v_t, respectively.
- g_t is the gradient at time step t.
- ϵ is a small scalar used to prevent division by zero.

This thesis uses Adam since it adjusts the learning rate for each parameter individually based on estimates of first and second moments of the gradients, which means it scales the learning rate for each parameter dynamically.

This can be effective for complex models and datasets, and makes Adam perform well with objectives that are noisy or change over time. Adam does not require a significant amount of memory, making it is advantageous when dealing with large models or datasets.

3.3.14 Loss Functions

A loss function, in the context of Deep Neural Networks, is a mathematical tool used to measure the performance of the model. It quantifies how well the model's predictions match the actual data by calculating the error or difference between the predicted outputs of the model, and the actual target.

The goal of training a Deep Neural Network model is to minimize this loss [61]. This thesis uses the Mean Squared Error (MSE) optimization function, commonly used for regression problems, and its calculation is represented below 3.16:

Mean Square Error (MSE):

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.16}$$

Where n is the total number of data points, y_i is the actual value for the i^{th} data point, in relation to this thesis it is measured as PPV. The $\widehat{y_i}$ is the model's prediction for the i^{th} data point, which is considered as the predicted PPV.

To achieve the best-fit regression line, the model aims to predict the target value such that the error difference between the predicted value and the true value is minimal, where the importance lies in updating the slope value a and intercept value b, to reach the best value that minimizes the error between the predicted value and the true y value. The MSE was chosen based on the goal of the thesis of creating a the Deep Neural Network, designed to perform a predictive modeling task. A single-task model with a clear objective in predicting PPV, benefits from the simplicity and direct focus of a single loss function.

Root Mean Square Error (RMSE)):

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.17}$$

Where n is the total number of data points, y_i is measured as PPV, $\widehat{y_i}$ is the predicted PPV.

RMSE is a measure of the differences between values predicted by a model and the values actually observed from the environment being modeled. RMSE is a standard way to measure the error of a model in predicting quantitative data. Lower RMSE values indicate better fit. However, it can be sensitive to outliers.

MSE is similar to RMSE , but is the average of the squares of the errors, which is the average squared difference between the estimated values and the actual value. MSE is a risk evaluation metric corresponding to the expected value of the squared error loss. The lower the MSE, the better the model's performance.

Mean Absolute Error (MAE):

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.18}$$

Where n is the total number of data points, y_i is measured as PPV, $\widehat{y_i}$ is the predicted PPV.

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It is the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight. MAE is considered less sensitive to outliers when compared to RMSE and MSE.

3.3.15 Coefficient of Determination (R-squared):

The Coefficient of Determination, also known R-squared or R^2 is considered a critical measurement when evaluating the performance of regression models, including those used in deep learning. It is not a direct output like training or validation loss, but provides a complementary perspective on model performance.

R-squared is the proportion of the variance in the dependent variable that is predictable from the independent variables. For Deep Neural Network models this addresses regression tasks, such as predicting Peak Particle Velocity (PPV) in mining operations.

R-squared is a statistical measure of how close the data are to the fitted regression line, and a higher R^2 value indicates that a larger proportion of variance in the dependent variable is explained by the model, which generally implies a better fit to the observed data.

R-squared R^2 serves as a measurement instrument of model accuracy, which is considered useful for comparing the predictive performance of different models or assessing the improvements in model performance after tuning parameters or adding complexity.

Mathematically, R-squared is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{3.19}$$

Where n is the total number of data points, y_i is measured as the actual observed values of PPV, $\widehat{y_i}$ is the predicted PPV values predicted by the model, and $\overline{y}$ is the mean of measured PPVs of the observed data.

3.4 Related Work

Numerous related work have been conducted in applying artificial Artificial Neural Networks (ANNs) and Machine Learning (ML) to mining operations, demonstrating promising outcomes. These studies collectively contribute to an evolving understanding of how ML techniques can optimize mining practices, particularly in blast-induced ground vibration prediction and operational strategy optimization.

This section provides an overview of various studies that have applied different ML methods to mining. These methods range from simple regression models to more complex like decision trees and ensemble methods.

A short description of the main findings and approaches, methodologies, algorithms and results of these studies are summarized in table 3.4. A list and description of the type of Machine Learning models and algorithms can be found in 3.4.2. Researchers have used specific evaluation metrics to measure performance, the amount of error in predictions, and how well a model explains the data it's been given. These metrics are listed in the summarized studies 3.4, but their mathematical formula with descriptions be found in section 3.3.14, or the description of R-squared, 3.19, or section of the metrics used in related work 3.4.1.

For readers seeking an in-depth analysis and a more thorough exploration of each study, a detailed summary can be found in related work 3.4.3. These summaries go into each study in more depth, describing their methods, models and algorithms, their results and how the research contributes to the field.

Summary of Related Work

Authors	Year	Study Method	Algorithms Used	Evaluation Metrics	Location	Results and Key Findings
Fissha, Ikeda, Toriya, Adachi, and Kawamura	2023	Comparative Evaluation	Bayesian Neural Network (BNN), Random Forest, Gradient Boosting, K-Neighbors, Decision Tree	Root Mean Square Error (RMSE), Root (R), Mean Square Error (MSE)	Mikurahana quarry, Japan	Bayesian Neural Network (BNN) a directed probabilistic graphical model, representing variables and conditional dependencies via a directed acyclic graph, here with 8 input parameters and 100 datasets from blasting outperformed traditional ML methods in predicting PPV. The nonlinear structure and adaptability of the BNN model enabled more precise estimation of PPV [19].
Monjezi, Ahmadi, Sheikhan, Bahrami and Salimi	2010	Predictive Analysis	Mulitlayer perception Neural Network (MLPNN), Radial Basis Function Neural Network (RBFNN), General Regression Neural Network (GRNN)	Root Mean Square Error (RMSE), R^2	Sarcheshmeh copper mine, Iran	Confirmed that Neural Network architectures to predict blast-induced ground vibrations had good results. Mulitlayer perception Neural Network (MLPNN) was superior in predicting blast-induced ground vibrations, with RMSE of 0.03, R^2 of 0.954. Sensitivity analysis showed the influence of distance from blast, number of holes per delay, max charge per delay was features that impacted blast-induced ground vibration [48].
Nguyen, Bui, Tran, Le, Do and Hoa	2018	Predictive Analysis	Artificial Neural Network (ANN)	Root Mean Square Error (RMSE), R-squared/R^2	Coal mine, Vietnam	ANN with specific hidden layer showed superior predictive capabilities, with RMSE of 0.738 and R^2 of 0.964. Found single-hidden-layer ANN models not suitable due to inaccuracies [52].
Hosseini, Pourmirzaee, Armaghani, and Sabri	2023	Predictive Analysis	Artificial Neural Network Ensemble, EXGBoost	R^2 , Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Variance Accounted For (VAF), Accuracy	Lead-zinc open-pit mine, Middle East	EXGBoosts exhibited superior performance in PPV prediction compared to individual models. EXGBoosts is the implementation of gradient-boosting algorithm to optimize ML models. The spacing parameter and the number of blast-holes were found to have the most and least significant influences on PPV, respectively [29].

Samareh, Khoshrou, Shahriar, Ebadzadeh and Eslami	2017	Predictive Analysis	Nonlinear Regression Analysis (NLRA) , Artificial Neural Network (ANN)	Correlation Coefficient (r)	Not specified	Optimized nonlinear regression analysis (NLRA) model outperformed ANN in terms of correlation coefficient (0.854 for NLRA vs 0.662 for ANN). The optimized model demonstrated a more favorable performance in PPV prediction [57].
Nguyen, Bui, and Topal	2023	Predictive Analysis	SONIA with Metaheuristic Algorithms	MAE, RMSE, MAPE, R^2	Open-pit coal mine, Vietnam	Manta Ray Foraging Optimization (MRFO) - self-organizing Neural Networks (SONIA) model showed most accurate predictions, lowest error rates and highest reliability. SONIA models with other algorithms had lower performance [50].
Guo, Zhao, and Li	2023	Predictive Analysis	PSO-LSSVM, GA-BP, LSSVM, BP	Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Correlation Coefficient (r)	Not specified	Introduces a novel hybrid intelligent model for predicting blast-induced ground vibrations with particle swarm algorithm (PSO) - least-squares support vector (LSSVM) model excelled in predicting vibrations with an RMSE of 1.954, MAE of 1.717, and r value of 0.965, demonstrated the control of blast-induced vibrations through two-objective optimization [21].
Nguyen, Bui, & Topal	2023	Predictive Analysis	SalSO-ELM, SpaSO-ELM, MFO-ELM	Accuracy	Coc Sau coal, Vietnam	Sparrow Search Optimization (SpaSO) - Extreme Learning Machine (ELM) model achieved the highest accuracy rate in PPV prediction (91.4%). Other hybrid models showed slightly lower performance [51].
Nguyen, Choi, Bui, and Nguyen-Thoi	2019	Predictive Analysis	PSO, GA, ICA, ABC optimized SVR	R-squared, Root Mean Square Error (RMSE), Mean Absolute Error (MAE)	Limestone quarry, Vietnam	Genetic Algorithm-Support Vector Regression-Radial Basis Function (GA-SVR-RBF) model identified as the optimal technique for PPV estimation, showing superior performance when combined with the SVR model [53].

3.4.1 Evaluation Metrics Used In Related Work

A variety of statistical evaluation metrics are used to evaluate the performance of the models from related work, listed in the table 3.4. These evaluation metrics, including R-squared 3.19 and the evaluation metrics for loss functions, Mean Square Error 3.16, Root Mean Square Error 3.4.3 and Mean Absolute Error 3.18, each offer unique insights into different aspects of model performance.

The evaluation metrics measures the error between predicted and actual values and the correlation and determination strength of models. These evaluation metrics is used for interpreting model results, guiding the selection of appropriate modeling techniques, and optimizing predictive accuracy. This section provides a concise overview of some of the key evaluation metrics employed in this modeling research.

Variance Accounted For (VAF): is an evaluation metric for assessing the performance of a model, by measuring the proportion of variance in the dependent variable that can be predicted from the independent variable(s). VAF is a measure of explanatory power, similar to R-squared.

$$\text{VAF} = 100 \times \left(1 - \frac{\text{Var}(y_i - \hat{y}_i)}{\text{Var}(y_i)}\right) \tag{3.20}$$

VAF indicates the proportion of variance in the observed data that is explained by the model. Higher values indicate better model performance, where a high VAF indicates that the model explains a large portion of the variance in the data. It is similar in interpretation to Coefficient of Determination (R^2).

VAF was used in the research by Hosseini et al. in 2023 on a Zinc open-pit mine in the Middle East. The research description and results be found in the Summary of Related Work 3.1, or in depth in the Related Work section 3.4.3.

Mean Absolute Percentage Error (MAPE): s not typically classified as a loss function, but as an accuracy measure, often used in time series analysis. MAPE represents the average of the absolute percentage errors between the predicted and actual values.

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{3.21}$$

MAPE measures the size of the error in percentage terms. It is calculated as the average of the absolute errors divided by the actual values, expressed as a percentage. MAPE is scale-independent and can be used to compare forecasts across different data scales. It has the disadvantage of being infinite or undefined if there are any zero values in the actual data.

MAPE was used in the research by Nguyen, Bui and Topal in 2023, in an open-pit coal mine in Vietnam. The research description and results be found in the Summary of Related Work 3.1, or in depth in the Related Work section 3.4.3.

Accuracy: Accuracy is defined as the ratio of correctly predicted instances to the total number of instances, and is a good measure when the classes are well-balanced. It can be misleading in the presence of imbalanced classes. In a case where 90% of the data belongs to one class, a model could achieve 90% accuracy by simply predicting that class for all instances, and would not lead to a informative model.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{3.22}$$

Accuracy is generally used to describe the closeness of a measurement to the true value. It is a measure of how well a model correctly identifies or excludes a condition. However, it can be misleading if the class distribution is imbalanced.

Research by Hosseini et al. in 2023, and Nguyen, Bui and Topal in 2023 used this evaluation metric, and the description can be found in the summarized table for Hosseini her 3.1 or in depth here 3.4.3, or summarized for Nguyen, Bui and Topal here 3.1, or more in depth in 3.4.3.

Correlation Coefficient (r): In Deep Neural Networks, the correlation coefficient is useful in regression tasks or to measure the strength of a linear relationship between predicted values and true values. It measures how well the variation in one variable predicts the variation in another. This evaluation metric is informative in cases where the model output is continuous.

$$r = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} \tag{3.23}$$

Correlation Coefficient measures the strength and direction of a linear relationship between two variables. Ranges from -1 to 1, where a value close to 1 implies a strong positive correlation, while a value close to -1 implies a strong negative correlation. A value around 0 implies little or no linear correlation.

Research by Samareh et al. in 2017 used this evaluation metric, and the description of methods, algorithms and results can be found summarized in the table here 3.1, or in depth here 3.4.3.

3.4.2 Algorithms Used In Related Work

Various algorithms and methods have been employed to analyze and predict data in related work. These algorithms contribute significantly to the accuracy and efficiency of the models. The summary below provides an overview of the key algorithms from the research 3.4 that has been the basis of this thesis.

The explanations in the table aims to describe the concepts and highlight the practical applications of each algorithm, in relation to related work.

Model	**Abbreviation**	**Description**
Bayesian Neural Network	BNN	Combines Neural Networks with Bayesian statistics. In BNN, the weights are assumed to be random variables with specific probability distributions, rather than fixed values. This allows for estimating uncertainty in predictions, providing a probabilistic interpretation of model outputs which is crucial in applications where understanding the confidence in predictions is as important as the predictions themselves [19].
Random Forest	RF	An ensemble learning method using multiple decision trees during training and outputs the mode of the mean prediction of the individual trees. It improves predictive accuracy and controls overfitting by averaging various trees [19].
Gradient Boosting	GD	Is a Machine Learning technique for regression and classification tasks, which constructs a predictive model in a stage-wise fashion as an ensemble of weak prediction models, typically decision trees. It optimizes a loss function by iteratively adding new models that adress the shortcomings of the existing model ensemble, effictively reducing erros through gradient descent [19].
Decision Tree	DT	Constructs a tree-like model, where each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label or value, depending on tasks being classification or regression [19].
k-Nearest Neighbors	k-NN	Is a simple, yet effective, Machine Learning method used for both classification and regression. It operates by identifying the k nearest data points to a given input point and makes predictions based on the majority label (for classification) or the average of the labels (for regression) of the neighboring points [19].

Multi-Layer Perception Neural Network	MLPNN	Is a type of feedforward Artificial Neural Network that consists of multiple layers, including input, hidden, and output layers. It works by processing inputs through these layers using weighted connections and non-linear activation functions, enabling it to learn complex patterns and relationships in data for tasks like classification and regression [48].
Radial Basis Function Neural Network	RBFNN	Is a type of Artificial Neural Network that uses radial basis functions as its activation functions, good for function approximation and time-series prediction by mapping inputs to higher-dimensional spaces, where it becomes easier to linearly separate data for analysis [48].
General Regression Neural Network	GRNN	Is a type of Neural Network that specializes in regression tasks, closely related to Radial Basis Function networks. GRNN can be used for regression, prediction, and classification. It operates by estimating continuous variables, making it highly effective for real-time prediction and learning tasks [48].
Artificial Neural Network Ensemble	ANN Ensemble	An ensemble can be considered a learning technique where many models are joined to solve a problem, because an ensemble tends to perform better than singles improving the generalization ability. ANN Ensemble combines multiple ANN models to enhance predictive performance by combining the predictions from multiple Neural Network models to reduce the variance of predictions and reduce generalization error [29].
Extreme Gradient Boosting	XGBoost	An optimized gradient boosting technique, efficient in handling large datasets. It works by sequentially adding predictors to an ensemble, each one correcting its predecessor, and employs sophisticated regularization techniques to control overfitting [29].
Nonlinear Regression Analysis	NLRA	A form of regression analysis where data is fit to a model and expressed as a mathematical function that uses a generated line to fit an equation to some data. The sum of squares is used to determine the fitness of a regression model, which is computed by calculating the difference between the mean and every point of data [57].
Self-Organizing Neural Network Intelligence Algorithm - Metaheuristic	SONIA with Metaheuristic Algorithms	This hybrid approach combines the adaptability of SONIA with the optimization power of metaheuristic algorithms. SONIA, as a Neural Network, learns from data by adjusting its structure and weights based on input patterns, enhancing its ability to recognize complex patterns and relationships. The integration with metaheuristic algorithms, aids in fine-tuning the Neural Network's parameters and structure for optimal performance. This combination is particularly effective in solving complex prediction and optimization problems where conventional methods might struggle, making it suitable for predictive modeling in various fields [50].
Particle Swarm Optimization - Least Squares Support Vector Machine	PSO-LSSVM	PSO is a metaheuristic optimization algorithm inspired by the social behavior of birds, used to optimize a problem by iteratively improving a solution based on a simple movement and velocity update rules. LSSVM is a variant of the Support Vector Machine, a ML method used for classification and regression tasks, which minimizes an objective function composed of a regularized least squares term. With PSO-LSSVM, the PSO optimizes the hyperparameters of the LSSVM, such as the penalty parameter and the kernel parameters, ensuring the model achieves the best possible performance [53].

Genetic Algorithm - Backpropagation	GA-BP	Merges Genetic Algorithms with Backpropagation to optimize NN training. Genetic Algorithms are optimization techniques inspired by the process of natural selection. In GA-BP, the GA is used to optimize the initial weights and structure of a NN, which is then further refined through the Backpropagation algorithm, a standard method for training NNs by adjusting weights in the network to minimize error [53].
Sparrow Search Optimization - Extreme Learning Machine	SpaSO-ELM	Integrates the SpaSO algorithm with an ELM framework, where SpaSO is inspired by the foraging behavior of sparrows, used to optimize complex problems by simulating their social behavior and communication strategies. It is employed to determine the optimal parameters and weights of the ELM, a type of feedforward NN known for its fast learning speed and simplicity, which does not require iterative tuning of the weights. This hybrid effectively enhances performance and accuracy [51].
Moth-Flame Optimization - Extreme Learning Machine	MFO-ELM	Uses Moth-Flame Optimization algorithm to fine-tune parameters and weights of the Extreme Learning Machine framework, a fast and efficient type of Neural Network. The MFO algorithm, inspired by the navigation method of moths in nature. This integration enhances the ELM's ability to rapidly train and make accurate predictions, making MFO-ELM suitable for tasks like pattern recognition, classification, and regression in various complex applications [51].
Particle Swarm Optimization	PSO	Is a computational method inspired by the social behavior of birds and fish, that optimizes problems by iteratively moving individual particles (potential solutions) within the search space towards the best found positions, combining personal bests with the group's overall best to find optimal solutions efficiently [53].
Genetic Algorithm	GA	Is an optimization technique based on the principles of natural selection and genetics, that works by creating a population of potential solutions, then iteratively applying operations like selection to evolve the solutions towards an optimum [53].
Imperialist Competitive Algorithm	ICA	A socio-politically inspired algorithm, where each potential solution is represented as an "empire", that compete for dominance, and over time, the algorithm iteratively evolves and reorganizes the empires to improve the quality of the solutions [53].
Artificial Bee Colony Optimized - Support Vector Regression	ABC-SVR	The ABC simulates the foraging behavior of honey bees. With a population of food positions, the artificial bees modify these positions over time by using computational agents called honeybees to find the optimal solution. The SVR works by finding a hyperplane that best fits the data while minimizing the margin of error. ABC helps fine-tune these hyperparameters to achieve the best regression results. The combination of ABC and SVR allows for the automatic tuning of SVR's hyperparameters [53].

3.4.3 Related Work Detailed

This section provides a more detailed understanding of the methodologies, statistical evaluations, and implications of the findings in the numerous studies that have been conducted in the realm of artificial Artificial Neural Networks and Machine Learning applied to mining operations.

Fissha, Ikeda, Toriya, Adachi, and Kawamura's (2023) study conducted a comprehensive evaluation of predictive models, and found that using Neural Network outperformed traditional methods [19]. Their research aimed to predict blast-induced ground vibration of the Mikurahana quarry located in Japan, by comparing Bayesian netural network (BNN) with four Machine Learning techniques, namely gradient boosting, k-neighbors, decision tree, and random forest. Their results revealed that the BNN model outperformed the traditional Machine Learning regression analyses.

Their BNN model had eight input parameters, one output, used with one hundred datasets from blasting, and assessing their performance using various evaluation metrics such as Root (R) 3.4.3, Root Mean Suare Error (RMSE) , and Mean Square Error (MSE) 3.18. Their BNN model's nonlinear structure and adaptability found to enable more precise estimation of PPV parameters compared to traditional methods [19].

Another research by Monjezi, Ahmadi, Sheikhan, Bahrami and Salimi (2010) confirmed that using Neural Network architectures to predict blast-induced ground vibrations had good results. Their research was carried out in Sarcheshmeh copper mine located in Iran, with different input parameters named "distance from the blasting location", "maximum charge per delay", "burden to spacing ratio", "number of holes per delay", for prediction PPV as output parameter. The performance of each Neural Network model was evaluated using coefficient of determination (R^2) and root-mean-square of errors (RMSE) 3.4.3. Their architecture consisted of multi layer perception Neural Network (MLPNN), radial basis function Neural Network (RBFNN) and general regression Neural Network (GRNN), and the MLPNN came out as the superior performer [48].

In support of this assertion, statistical evaluation metrics such as the root mean square error (RMSE) 3.4.3 and the coefficient of correlation (R^2) were determined to be 0.03 and 0.954, respectively. In general, an RMSE value closer to 0 indicates a better fit. However, the context is essential; the acceptability of this error margin heavily depends on the scale of data and the problem domain. In relation to blast vibrations predictions in mining engineering, a deviation of 0.03 is considered very precise, given that the measurements can vary widely and are influenced by numerous factors. A coefficient of correlation of 0.954 suggests that the model does an excellent job of predicting the real-world values. Additionally, through sensitivity analysis, it was established that factors influencing blast-induced ground vibration during blasting operations encompassed the distance from the blast, the number of holes per delay, and the maximum charge per delay [48].

Another study using Artificial Neural Network by Nguyen, Bui, Tran, Le, Do and Hoa (2018) looked into the predicament of PPV by employing a series of ANN models to forecast blast-induced PPV in an open-pit coal mine located in Vietnam [52]. The dataset comprised data from 68 blasting events, where the operations were recorded with three parameters known as "maximum explosive charge per delay", "monitoring distance", and blast-induced ground vibration (PPV), with five distinct ANN models were developed in this investigation. The performance assessment relied on evaluation metrics such as the root-mean-squared error (RMSE) 3.4.3 and determination coefficient (R^2). Remarkably, the ANN characterized by 10 neurons in the first hidden layer, 8 neurons in the second hidden layer, and 5 neurons in the third hidden layer, exhibited superior predictive capabilities.

The Root Mean Squared Error (RMSE) 3.4.3 is one of the two main performance indicators for a regression model, which measures the average difference between values predicted by a model and the actual values. RMSE provides an estimation of how well the model is able to predict the target value (accuracy) 3.22. This study resulted in an RMSE of 0.738, and R^2 had 0.964.

The conclusion research emphasizes that the number of parameters does not reflect all characteristics of the data, and suggest that further research should consider using Neural Network with many hidden layers. Nguyen et al. (2018) found that single-hidden-layer ANN models are not suitable for PPV prediction due to their failure to capture the data set's underlying characteristics, leading to forecasting inaccuracies [52].

Another research using artificial Neural Networks by Hosseini, Pourmirzaee, Armaghani, and Sabri (2023) harnessed Machine Learning ensemble techniques in their investigation conducted at one of the Middle East's largest lead-zinc open-pit mines to predict blast-induced ground vibration. [29] Their research centered on Peak Particle Velocity (PPV) prediction in surface mining operations, employing two ensemble systems: an ensemble of artificial Neural Networks (ANN) and an ensemble of extreme gradient boosting (EXGBoosts) for PPV prediction [29].

XGBoost, an artificial intelligence technique represents a method in the field of Machine Learning, that works by combining the predictions of multiple weak models, typically decision trees, into a single strong predictive model. It does this through a process of iterative model building and optimization, where each decision tree added to the ensemble corrects the errors of the previous ones, gradually improving the model's accuracy. By using a dataset of 162 blasting records and seven influential parameters, Hosseini et. al. (2023) enabled multiple ANNs and XGBoost base models to be developed, each characterized by distinct architectural configurations. Subsequently, performance evaluation relied on validation indices, including the coefficient of determination (R^2), root mean square error (RMSE) 3.4.3, mean absolute error (MAE) 3.18, variance accounted for (VAF) 3.20, and accuracy 3.22, applied to assess the base models' effectiveness.

The top five performing base models were selected to construct ensemble models for both ANN and XGBoost methods. These ensemble models were then integrated using the stacked generalization technique to generate a unified prediction [29]. The outcomes of the study underscore the efficacy of ensemble models in enhancing the accuracy of PPV prediction compared to individual models. Among the various methods explored, EXGBoosts exhibited superior performance in PPV prediction. Furthermore, a sensitivity analysis elucidated that the spacing parameter and the number of blast-holes exerted the most significant and least significant influences on PPV intensity, respectively [29].

In another research using an Artificial Neural Network by Samareh, Khoshrou, Shahriar, Ebadzadeh and Eslami (2017), an initial step involved the identification of four out of eleven blasting and geo-mechanical parameters of rock masses that exerted the most substantial influence on vibrational wave velocities [57]. This selection was accomplished through rigorous regression analysis. Subsequently, models were devised for PPV prediction using both nonlinear regression analysis (NLRA) and Artificial Neural Network (ANN) techniques, yielding correlation coefficients (r) 3.23 of 0.854 and 0.662, respectively [57].

The correlation coefficient (r) of 0.854 suggests a strong positive linear relationship between the predicted values and the actual values of PPV, and implies that the model predicts PPV with high reliability. The ANN model has a correlation coefficient (r) of 0.662, which indicates a moderate positive linear relationship.

Furthermore, the coefficients associated with the parameters in the NLRA model were fine-tuned through an optimization process employing the particle swarm-genetic algorithm. To assess the accuracy and performance of the developed models, PPV values were estimated for an additional dataset consisting of 18 test cases. The evaluation, based on statistical indices for the test data, revealed that the optimized model outperformed the other two models, offering more precise PPV predictions. Notably, the optimized nonlinear model demonstrated a more favorable performance, as evidenced by a correlation coefficient (r) of 0.75, in comparison to the other two models [57].

Another study done by Nguyen, Bui, and Topal (2023) aimed at presenting a method for predicting blast-induced ground vibration in open-pit mines, focusing on the application of self-organizing Neural Networks (SONIA) in conjunction with metaheuristic algorithms [50] To enhance the accuracy of the SONIA model, various metaheuristic algorithms, including Manta Ray Foraging Optimization (MRFO), Hunger Games Search (HGS), Aquila Optimization (AO), and Naked Mole-Rat Algorithm (NMRA), were employed. Additionally, the k-fold cross-validation technique was leveraged to identify optimal algorithm parameters, subsequently facilitating model retraining for the prediction of blast-induced ground vibration [50].

The study's effectiveness was assessed through a case study involving an open-pit coal mine in Vietnam, encompassing 288 blasting events. Results indicated that SONIA, owing to its self-organizing structure, was well-suited for predicting blast-induced ground vibration, even when dealing with a limited dataset featuring intricate relationships. However, the SONIA model's accuracy could be further improved through optimization with the selected metaheuristic algorithms. Among these, the MRFO-SONIA model emerged as the most reliable and accurate, exhibiting the lowest error rates (MAE = 0.379, RMSE = 0.453, MAPE = 0.08) and the highest reliability (R^2 = 0.896). These results indicates a high level of accuracy, and good predictions with a low value of MAE. MAPE 3.21 represents the average absolute percent error for each prediction, and with a value of 0.08 suggests that the model's predictions are off by 8 percent on average, indicating a reasonable result in a complex domain. The model seems to have a high R^2 , indicating strong predictive power.

In contrast, the HGS-SONIA, AO-SONIA, and NMRA-SONIA models demonstrated comparatively lower performance, with MAE values of 0.455, 0.500, and 0.492, RMSE values of 0.552, 0.603, and 0.580, MAPE values of 0.100, 0.112, and 0.111, and R^2 values of 0.845, 0.815, and 0.829, respectively. These findings underscore the potential of metaheuristic-based SONIA models in enhancing the prediction of blast-induced ground vibration in open-pit mines, with potential applications extending to various mining operations where the prediction of vibrations or other adverse effects resulting from specific mining activities is essential [50].

Research by Guo, Zhao, and Li (2023) underscore the significance of prediction and parameter optimization as effective tools for mine personnel to manage blast-induced ground vibrations. [21] However, the inherent complexity of open-pit blasting, characterized by numerous influencing factors and effects, poses a challenge to achieving accurate prediction and optimization. To address this challenge, the study by Guo, Zhao and Li introduces a novel hybrid intelligent model for predicting blast-induced ground vibrations. This model combines a least-squares support vector machine (LSSVM), optimized by using a particle swarm algorithm (PSO). Meanwhile, a multi-objective particle swarm optimization (MOPSO) approach is employed to optimize blast design parameters, taking into account specific site conditions and the desired bulk fragmentation rate [21].

To evaluate the predictive performance of the PSO-LSSVM model, a comparative analysis is conducted against alternative methods, including a genetic-algorithm-optimized BP Neural Network (GA-BP), an unoptimized LSSVM, and a conventional BP model, all utilizing the same dataset. Performance assessment relies on key evaluation metrics such as the root-mean-squared error (RMSE) 3.4.3, mean absolute error (MAE) 3.18, and correlation coefficient (r) 3.23. Moreover, the study verifies the optimization results for blast parameters through practical field tests [21].

The findings reveal that the PSO-LSSVM model excels in efficiently predicting vibrations, exhibiting an RMSE of 1.954, MAE of 1.717, and an r value of 0.965. Additionally, this study demonstrates the potential for controlling blast-induced vibrations by employing a two-objective optimization model to determine optimal blast parameters [21].

Nguyen, Bui, and Topal (2023) conducted a study with the objective of predicting blast-induced ground vibration intensity resulting from mine blasting, specifically focusing on Peak Particle Velocity (PPV). . To achieve this goal, they developed three innovative intelligent models utilizing a combination of metaheuristic algorithms and the Extreme Learning Machine (ELM) model, namely Salp Swarm Optimization-ELM (SalSO-ELM), Sparrow Search Optimization-ELM (SpaSO-ELM), and Moth-Flame Optimization-ELM (MFO-ELM). These models leveraged the distinct optimization mechanisms of SpaSO, SalSO, and MFO algorithms to refine the weights of the ELM for PPV prediction [51].

The study's performance assessment involved the utilization of 216 blasting records, with corresponding PPV measurements obtained from the Coc Sau open-pit coal mine located in North Vietnam. Differing activation functions for the ELM model were employed in configuring the algorithms' parameters. Additionally, to gauge the improvements introduced by SpaSO-ELM, SalSO-ELM, and MFO-ELM models, the researchers also examined and evaluated the standalone ELM alongside two empirical models (linear and nonlinear) [51].

The study's outcomes underscored the potential of nonlinear models for PPV prediction, with the ELM-based models demonstrating robust capabilities in modeling the dataset's nonlinear relationships. Practical engineering validation further reinforced these findings, revealing that the SpaSO-ELM model emerged as the most accurate intelligent model for PPV prediction, achieving an accuracy rate of 91.4%. The remaining hybrid models exhibited slightly lower performances, falling within the range of 89.8% to 90.5%. Despite the improved predictive performance of nonlinear empirical models compared to linear models, their accuracy still lagged significantly behind the proposed hybrid intelligent models [51] Consequently, the metaheuristic-based ELM models optimized in this study are considered highly reliable tools for predicting blast-induced ground vibration intensity in open-pit mines, thereby enhancing the safety of the surrounding environment [51].

In a study conducted by Nguyen, Choi, Bui, and Nguyen-Thoi (2019), blast-induced ground vibration(PPV), was quantified through the utilization of vibration sensors with the use of an empirical dataset of 125 blasting records collected and analyzed at a limestone quarry located in Vietnam [53]. Various evolutionary algorithms were systematically evaluated to predict PPV, encompassing the Particle Swarm Optimization (PSO) algorithm, Genetic Algorithm (GA), Imperialist Competitive Algorithm (ICA), and Artificial Bee Colony (ABC). These evolutionary algorithms were employed to optimize the Support Vector Regression (SVR) model, leading to the development of four hybrid models [53]. To assess and compare the performance of the developed models, various statistical evaluation metrics including R-squared, Root Mean Square Error (RMSE)3.4.3, and Mean Absolute Error (MAE) 3.18 were employed. The findings underscored the superior performance of the GA algorithm when combined with the SVR model for addressing the specific problem at hand. Furthermore, the Radial Basis Function (RBF) kernel function emerged as the most effective choice for the GA-SVR model. Consequently, the GA-SVR-RBF model was identified as the optimal technique for accurate PPV estimation [53].

4

Methods And Methodologies

The aim of this chapter is to offer an understanding of the chosen methods and methodologies chosen for this thesis. The methodology section will adhere to the framework described in Anne Håkansson's work, "The Portal of Research Methods and Methodologies" [22], which serves as a guiding framework for structuring and presenting the research methods utilized in this study.

The methodological framework presents the use of inductive reasoning to develop neural network models from data patterns, and deductive reasoning to test hypotheses on blast features and PPV. The chapter underscores the rigorous data collection from blasting events, ensuring data quality and privacy. The platforms and libraries used in creating the deep neural network is described, detailing the workflow from data handling to model training.

The chapter wraps up by emphasizing the iterative development process of the research, which includes problem definition, model design, empirical testing, and continuous refinement. This process culminates in a predictive model aimed at enhancing efficiency and safety in mining operations.

4.1 Methodological Framework

The methodological framework of this thesis is based on Anne Håkansson' work, "The Portal of Research Methods and Methodologies." [22] This work offers a complete guide to the selection of research methods, providing a structural navigation in methodological choices available.

The selection process of method involves exploring each layer of the portal, from philosophical assumptions to presentation methods, ensuring a theoretical understanding and practical application of at least one method from each layer before progressing to the next. The portal categorizes methods into quantitative and qualitative research, and offers a visual representation to assist the selection process, shown in Figure 4.1. This categorization has been significant in the current thesis's mixed-methods approach, allowing for a balanced integration of both quantitative data analysis and qualitative insights.

The portal acknowledges the dynamic nature of research by recognizing a spectrum of methods that lie between qualitative and quantitative approaches. The portal's framework supports the use of triangulation in the research of this thesis, by employing both qualitative and quantitative methods in a complementary way.

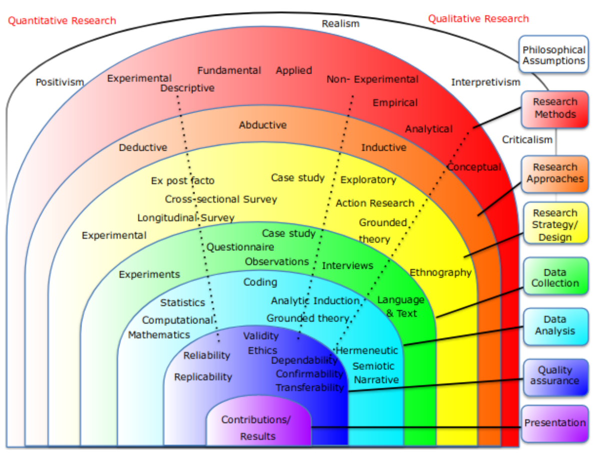

Figure 4.1: Portal of research methods and methodologies [22].

4.2 Research Methods

Research methods encompass the systematic approaches and techniques used to gather, analyze, and interpret data for the purpose of conducting research and answering research questions. This thesis uses a mixed method research which is a combination of qualitative and quantitative research methods. This combinations of methods is a comprehensive approach that integrates the strengths of both qualitative and quantitative methodologies to provide a more complete understanding of a research problem.

This integrative approach, known as triangulation, enhances the credibility and validity of the research findings. Typically, each research method is applied sequentially within the triangulation framework. Additionally, it recognizes a range of intermediate methods between these, allowing for flexibility in research selection [22].

4.2.1 Quantitative and Qualitative Research Methods

The left side of the portal includes quantitative research methodologies, which is objective and designed to facilitate empirical investigations through the systematic quantification of variables. Typically beginning with a theoretical framework, this approach investigates data to affirm or challenge the initial theory. Quantitative research involves collecting data through structured means, such as surveys and tests or leveraging existing data sets, where the analysis phase employs statistical, mathematical, or computational techniques. Findings from this approach are often presented as statistical models, graphical representations, and tables, providing a visual and numerical synthesis of the research outcomes [22].

The right side is dedicated to qualitative research methods. These are inherently subjective, exploratory, and descriptive, favoring an investigative stance that often deals with smaller, more focused data sets. These methods are designed for generating new theories or the crafting of artifacts. The essence of qualitative research is to explore underlying meanings, concepts, and detailed descriptions of the subject matter. Data is gathered via less structured means, including interviews, direct participation, or observation, allowing for a rich, in-depth perspective. Analysis within this realm is centered on selective emergent patterns and constructing narratives. Consequently, the results of qualitative research are viewed as conceptual frameworks, comprehensive theories, or detailed descriptive accounts [22].

This thesis involves testing a system with a larger data set, and to create an artifact, with the goal to impact the surrounding environment, which includes both qualitative and quantitative research methods. This requires a literature study and a planning of project to reach the desired outcome, and to achieve expected results.

In relation to this thesis, the essence of the quantitative approach is the predictive model itself. Since quantitative research is a systematic approach used to investigate and analyze phenomena through the collection and analysis of numerical data, the predictive model can be considered as a numerical tool that uses quantitative data to generate quantifiable predictions. The quantitative and qualitative research methods in relation to this thesis will be explained in detail in Research Approach.

4.3 Research Approach

Research methodologies are often defined as "the search of knowledge" or "systematic investigation to establish facts", and often employ various approaches to draw valid conclusions. In the realm of research and scientific inquiry, two fundamental methods of reasoning are often employed: inductive and deductive approaches. These methods represent different pathways of understanding and interpreting data and phenomena, with inductive and deductive reasoning being among the most frequent [22].

This thesis involve both inductive and deductive reasoning at different stages. The analyzing of data and identification of patterns relies on inductive reasoning, while the application of the model to make predictions and validate it against a test test, employed deductive reasoning.

4.3.1 Inductive Approach

Inductive reasoning is an approach that begins with specific observations or real instances and progresses towards broader generalizations and theories, with the aim to construct general theories based on specific observations. Typically employed with qualitative methods, an inductive approach collects and analyzes data to understand a phenomenon from multiple perspectives. Outcomes are rooted in behaviors, opinions, and experiences, and the data must be robust enough to elucidate the underlying patterns or requirements for a given artifact [22].

In the context of data analysis and machine learning in this thesis, inductive reasoning is central to the model development. Neural networks are considered inductive, as they learn from specific instances, known as data points, to generalize, identify patterns and make predictions about unseen data. The development of the deep learning model can be seen as inductive reasoning. Starting with specific observations with data collected from various blasting operations, and feeding this data into a neural network, the model is allowed to distinguish patterns and relationships between the variables. The model's ability to predict PPV for new sets of inputs is based on the generalizations it has formed during training, which is the essence of inductive reasoning.

4.3.2 Deductive Approach

The deductive approach is a top down approach that seeks to validate or invalidate predefined hypotheses. This is achieved through rigorous testing, usually relying on quantitative methods and extensive data sets. Deductive approach starts with a general statement or hypothesis, which must be operationally defined and measurable, specifying expected outcomes and the variables under research. The end result should offer generalizations grounded in collected data, and provide explanations for causal relationships between variables [22].

In the context of this thesis, deductive reasoning was employed to test a hypothesis using a general model that describes the relationship between various features and Peak Particle Velocity (PPV). Before training the neural network, it was hypothesized that a machine learning model would outperform the current industry-standard model in predicting blast-induced ground vibration. This hypothesis was examined by training the model with the dataset to see if the predictions corresponded with the anticipated outcomes. Specific feature data inputs to the neural network facilitated the generation of PPV predictions. Additionally, the implementation of regularization techniques, guided by general principles of machine learning, was incorporated as part of the deductive reasoning process. The deductive nature of the research was highlighted when using the model's predictions to either confirm or disprove the initial theory.

4.3.3 Quantitative Methods

In relation to this thesis, quantitative research is evident and utilized through the use of numerical data and statistical methods, to analyze and comprehend the functional and interface-related aspects of mining engineering's drill and blast operations. The core of the analysis is a comprehensive dataset, spanning thousands of entries and multiple columns, each encoding specific variables integral to these operations.

The quantitative approach here is methodical, employing a suite of machine learning algorithms and data analysis techniques to construct a predictive model. These model is calibrated to forecast critical outcomes, with a primary focus on predicting the Peak Particle Velocity (PPV). The integrity and reliability of the quantitative methods are underscored by their ability to not only present an exposure of current conditions but also to offer robust predictions that can inform future operational decisions [22].

4.3.4 Qualitative Methods

Qualitative research, in relation to this thesis, aims at understanding and interpreting complex phenomena of the drill and blast environment, through non-numerical data collection and analysis. This method offers a narrative that encapsulates the experiential and observational angle of blast operations. This approach is essential in shaping an understanding of the contextual elements that applies influence over the numerical data. Through the use of qualitative methods, this thesis constructs a detailed understanding that incorporates the prevailing conditions at blast sites, including environmental and operational variables that could alter the data.

Using a qualitative research method seeks the underlying patterns and relationships, thereby facilitating the creation of an artifact in the form of a computer system that is not only informed by empirical data but also by qualitative insights. This ensures that the development of technological solutions is both data-driven and contextually grounded, enabling advancements that are technically sound and practically relevant [22].

4.4 Research Strategies

Research strategies within methods and methodologies refer to the approaches and plans of action designed to achieve the specific aims of a research project within the broader framework of the chosen research method and methodology.

4.4.1 Design Science Framework

Design science is an approach that focuses on creating innovative solutions to practical problems [14]. It typically involves iterative cycles of design, development, and evaluation. In the realm of data-driven problem-solving, the Design Science methodology emerges as a structured framework for the development and evaluation of machine learning models. Through predictive and data-driven approaches, this methodology serves as an an iterative process where artifacts are designed, implemented, evaluated, and refined in cycles until a satisfactory solution is achieved.

In the Design Science framework, the model's performance has been continuously evaluated against a set of predetermined criteria, which included accuracy, precision, recall, and score for classification tasks, or metrics used to evaluate regression tasks. These criteria were selected to tell the achieved accuracy and precision, and the amount of deviation from the actual values. The criteria also ensure that the model meets the practical requirements of predicting PPV in blasting operations effectively. A feedback loop was essential to the iterative process, where results from each model iteration were reviewed. Adjustments were made based on this feedback, such as tuning hyperparameters, altering model architecture, or revising the data preprocessing pipeline.

The steps of the Design Science framework are listed here:

- **Problem Identification and Definition:** The foundation of any Design Science inquiry lies in the identification and definition of the problem at hand. It mandates the establishment of a clear problem statement, where the research objective of this thesis is defined as the creation of a machine learning model to effectively address this identified problem.

- **Designing the Learning Model:** The development of the machine learning model is the primary artifact creation phase and includes selecting appropriate algorithms, data preproecssing techniques, and model architecture.

- **Iterative Development:** Includes using an iterative approach to design, implement, and refine the learning model by starting with a basic version and gradually improve it based on evaluation results and feedback.

- **Implementation:** The realization of the designed model occurs through conscientious implementation. Programming languages and frameworks, such as Python, TensorFlow, or scikit-learn, serve as the medium for the construction of the model. The model is trained using the training dataset, that undergoes iterative epochs to optimize its performance.

- **Empirical Testing:** Conduct conscientious testing and experimentation to evaluate the effectiveness and efficiency of the model by testing with different data sets, and benchmarking against existing methods.

4.4.2 Task Force

The process of developing a machine learning model requires the composition of a well-coordinated task force comprising interdisciplinary roles and responsibilities. This collaboration is integral to the systematic creation and evaluation of machine learning models, ensuring the successful and reliability of the research methodology.

- **Problem Identification and Definition:** Defining the problem statement is essential. In relation to this thesis, it involves improving the accuracy of predictions for blast-induced ground vibrations by considering multiple variables. This involves understanding the operational context of drill and blast engineering and the variables that might influence PPV.

- **Model Architecture:** Involves designing the machine learning model, by making critical decisions regarding the selection of algorithms, deciding on the number and types of layers in the neural network, and determining the best activation functions and optimizers, based on the characteristics of the drilling and blasting data.

- **Iterative Development:** Involves data preparation. This encompasses data cleaning, normalization, and partitioning the dataset into training, validation and testing sets. The iterative development process includes training the model on the dataset, adjusting hyperparameters, and refining the model architecture based on performance metrics.

- **Implementation:** The realization of the designed model occurs through implementation with programming languages and frameworks, such as Python, TensorFlow, or scikit-learn, serve as the medium for the model's construction. The model is trained using the training dataset, undergoing iterative epochs to optimize its performance.

- **Empirical Testing:** The model is subjected to continuously empirical testing to evaluate its predictive performance. This includes cross-validation, performance on unseen test data, and comparison against baseline models or industry standards. The task force analyzes the results, checking for accuracy, precision, recall, and other relevant metrics to assess the model's effectiveness in predicting PPV.

- **Model Evaluation:** Once the model has been refined through iterative development and feedback integration, it undergoes a final evaluation. This phase checks for generalizability, robustness, and how well the model performs in simulating real-world scenarios. The task force might also evaluate the model's interpretability, understanding how the input features affect the predicted PPV.

- **Documentation and Reporting:** The final step involves documenting the entire process, model specifications, performance metrics, and the conclusions drawn from the empirical tests. This documentation is essential for transparency, reproducibility, and for informing future work.

4.4.3 Prototyping

Prototyping is a valuable technique often employed within design science methodology. Prototyping is the process of creating preliminary versions or models of a proposed solutions, as prototypes are not final products but rather representations of ideas. They serve as a means to test and refine design concepts to gather feedback from task force and make iterative improvements to the prototypes. Prototyping is an integral part of the design science methodology, that allows to bridge the gap between theory and practice by creating tangible representations of solutions [12].

The iterative nature of prototyping allows for the gradual incorporation of insights gained from each prototype iteration, ensuring that the final solution is well-aligned with the specific requirements of mining operations and PPV prediction. Each iteration of the prototype offers an opportunity to identify and address potential issues, refine the system's architecture, and enhance its performance. By rapidly iterating on the prototype, it can adapt to new information, change requirements, or unforeseen challenges, ensuring that the final model is both innovative and practical.

Involvement of a task force or stakeholder group in the prototyping process is invaluable. Their feedback provides real-world perspectives and insights, ensuring that the prototype remains relevant and effective in practical scenarios. This collaborative approach improves the quality of the prototype, emphasizes flexibility, adaptability, and user feedback.

In relation to this thesis, prototyping is an approach that integrates theory with practice, enabling the creation of a deep neural network that is both technically and tailored to the specific needs of mining operations. Through iterative improvements and stakeholder collaboration, prototyping enhances the likelihood of developing a successful, well-validated solution for PPV prediction.

4.5 Data Collection Methods

The foundation of an empirical research relies on the quality and integrity of its data. In the domain of mining engineering, the collection of accurate and comprehensive data is crucial for developing predictive models that can enhance operational efficiency and safety. The dataset provided for this study was compiled by the external stakeholder, and serves as the base of this analysis.

The dataset was compiled from multiple blasting sites to capture a wide array of variables that are noted as being significant to blasting operations, and is stored in a comma-separated values (CSV). Each data point is said to be recorded with accuracy, ensuring that the dataset reflects the complexities of real-world blasting scenarios. The following is an overview of the variables collected, providing a snapshot of the multifaceted nature of the data that forms the basis for the subsequent analysis.

- **Site:** Refers to the specific location where the blasting has occurred. This variable is significant as it indirectly captures the unique geological and environmental characteristics that may influence the blast results.

- **BlastID:** This is a unique numerical identifier assigned to each blasting event to differentiate it from others. This ID can be used to track and reference specific blasts for analysis or follow-up studies.

- **Scaled Distance (SD):** Scaled Distance is a calculated value that normalizes the actual distance from the blast based on the amount of explosives used. It is used to compare the effects of blasts of different sizes and at varying distances. It's a crucial variable for standardizing measurements and making comparisons more meaningful.

- **Distance:** The actual physical distance from the point of the blast to the measurement or observation point. This could influence the intensity of the blast's effects, such as vibrations.

- **Maximum Instantaneous Charge (MIC):** MIC is a term in blasting operations that refers to the largest mass of explosive detonated within a specific, brief timeframe, usually to minimize environmental impacts like vibrations and airblast. The MIC is a controlled parameter to ensure compliance with regulatory limits and to protect nearby structures and sensitive areas. It is considered a critical factor in blast design, influencing the sequencing and timing of explosions to achieve optimal fragmentation while mitigating adverse effects.

- **Blast Direction:** The orientation or angle at which the blast is directed, in this thesis it has been categorised as integers for ease of analysis. This can affect the directional distribution of energy and potential vibrational waveform superpositioning at locations around the blast.

- **Timeframe:** A parameter used to quantify MIC, or the timing of the blast within a sequence of multiple blasts. This can help in understanding the temporal context of the blast events and any potential cumulative effects.

- **Ground Water:** Indicates whether groundwater was present (1) or not

(0) at the blast site. Groundwater can significantly affect the transmission and the stability of the blast area.

- **PPV:** PPV is the measure of the maximum svelocity at which particles move at a point as vibrational shock waves pass through the ground after a blast. It is a critical measure for assessing the potential for damage to structures and the environment and is the primary outcome variable that the study aims to predict through machine learning models.

4.5.1 Data Privacy and Confidentiality

Data privacy and data confidentiality are crucial for maintaining ethical standards, legal compliance, and the integrity and trustworthiness of research. Data privacy ensures the respectful and lawful use of sensitive data that might be part of the dataset or research process. Data confidentiality, on the other hand, safeguards the sensitive information related to mining operations, including blasting practices, geological data, and operational strategies, from unauthorized access.

Data confidentiality complements privacy by emphasizing the protection of sensitive operational data from unauthorized disclosure. This includes geological data, blasting practices, and strategic information crucial to a mining company's competitive edge. Maintaining confidentiality is not only a matter of protecting intellectual property but also a safeguard against potential security risks and legal repercussions that could arise from data breaches.

The ethical handling of data in this thesis involves careful measures to uphold both privacy and confidentiality. Data privacy relates to handling, processing, storage, and usage of sensitive data to ensure that sensitive details are not compromised. Data anonymization techniques were employed to uncertain site-specific identifiers, thereby maintaining the anonymity of the data sources. Additionally, strict data handling routines were followed to limit data access and prevent unauthorized data transfer or storage.

By prioritizing these principles, the project ensures the integrity of the research and fosters trust among stakeholders. It demonstrates a commitment to ethical research practices, aligning with legal standards and protecting the interests of the mining company. The careful consideration given to data privacy and confidentiality emphasizes the research's credibility.

4.5.2 Data Reliability

Data reliability refers to the degree to which data is considered accurate and trustworthy for use in analysis and decision-making. It is a critical aspect of data quality that affects the validity of research conclusions, the robustness of models, and the effectiveness of any insights derived from the data [15].

- **Repeatability:** Refers to if the same data collection processes are repeated under the same conditions, they should produce the same results.
- **Reproducibility:** Different mine sites or mining engineers using the same data collection methods and processes should obtain similar results, indicating that the data is reliable regardless of who collects it.
- **Accuracy and Precision:** Reliable data should not only reflect true values (accuracy) but also yield consistent results across repeated measurements (precision).
- **Robustness:** Reliable data should be resilient against small changes in the environment or methodology, meaning that the reliability is not significantly affected by changes in the process.
- **Validity:** Data is valid if it accurately represents the concept it is intended to measure, which is essential for its reliability.

In the development of a ML Model for predicting peak particle velocity (PPV) in mining operations, data reliability is critical. The accuracy of data to real-world conditions is important for training the neural network, as it directly impacts the model's learning and generalization capabilities. Consequently, this influences the predictive accuracy essential for diverse blasting situations under varying conditions.

The significance of data reliability lies in its influence on the trustworthiness of research outcomes and the efficacy of decisions derived from the model's predictions. Utilizing unreliable data can lead to insignificant decisions. Conversely, reliable data ensures that the model's predictions are stable and transferable across different scenarios and datasets, to strengthen stakeholder trust and facilitating decision-making.

As the dataset was compiled and cleaned by the external stakeholder, there was no significant data-cleaning required.

4.5.3 Data Quality

Data quality is data based on factors like accuracy, completeness, reliability, and relevance. It is considered an aspect of data management that ensures the data is suitable for its intended use in operations, decision-making, and planning [66].

- **Accuracy:** Refers to the degree to which the data accurately reflects the real-world conditions or objects they represent. Accurate data should be free from errors and depict the true values.
- **Completeness:** Is about the extent to which all the required data is available. Incomplete data can lead to biased analyses and decisions based on partial information.
- **Consistency:** Consistent data aligns with other data across the system or dataset. Inconsistencies can occur when there are discrepancies in how data is collected or when it's updated in some places but not others.
- **Reliability:** Reliable data can be used with confidence; it's dependable and reflects stable and consistent data collection processes over time.
- **Relevance:** Data is relevant if it's applicable and helpful for the purpose for which it's used. Irrelevant data can lead to wasted resources and misinformed decisions.
- **Timeliness:** This relates to data being up-to-date and available when needed. Outdated data can be as problematic as inaccurate or incomplete data.
- **Validity:** Valid data is in the correct format and within the range of allowable values for the respective data model and domain context.
- **Uniqueness:** Each data element should be unique and not duplicated. Redundant data can lead to confusion and impair data quality.
- **Integrity:** Refers to the maintenance of, and the assurance of the accuracy and consistency of data over its entire lifecycle. It implies that data across the ecosystem is linked and coherent.

In the context of developing a ML model for predicting peak particle velocity (PPV) from blasting operations, data quality is crucial for employing advanced statistical and machine learning techniques. The importance of data quality also reflects the predictive model's output, which depends heavily on the quality of the input data. High-quality data must be accurate, complete, and consistent, which ensures that the underlying patterns learned by the neural network are representative of real-world phenomena.

In mining operations, outliers in the data may represent significant deviations due to rare but possible events, like unusual geological conditions or blasting practices that deviate from the norm. High data quality involves correctly identifying and understanding these outliers to ensure that the model is trained on accurate and comprehensive data, including these edge cases.

For the ML model to be effective, it must be trained on data that is both consistent and complete. Inconsistencies in data can lead to a model that does not perform well or that learns incorrect relationships. High data quality means that the data used for training and validation is free from such issues, leading to a more accurate and generalizable model.

4.5.4 TensorFlow

TensorFlow is an open-source framework for machine learning, known for its utility in constructing and training complex models capable of conducting evaluations on extensive data sets [1]. The core computational framework of TensorFlow revolves around tensors, which constitute multidimensional data arrays central to the computational operations performed by neural networks. Tensors capture a broad array of data representations, from simple scalars to complex vectors and multi-dimensional matrices, effectively facilitating the handling of various data types and intricacies inherent in machine learning tasks [1].

In TensorFlow's tensor hierarchy, a scalar is represent the fundamental unit, signaling a singular numerical value characterized by its magnitude. Vectors, comprising both row and column vectors, constitute 1D arrays of numerical elements of magnitude and direction. Each element within a vector can be uniquely addressed through a single index. Matrices, on the other hand, are 2D arrays consisting of numerical values requiring dual indices for precise referencing. Tensors, while surpassing the two-dimensional constraints of matrices, extend their utility to encompass scalars, vectors, and matrices, constituting arrays with more than two axes [1].

Every element within a tensor adheres to a single data type, ensuring consistency across the structure. Three core attributes characterize a tensor: a unique identifier, the dimensions or shape of the data it represents, and the kind of data it holds. Together, these attributes support a tensor's flexibility and utility in TensorFlow, establishing it as a critical component for a broad spectrum of machine learning operations and analyses [1].

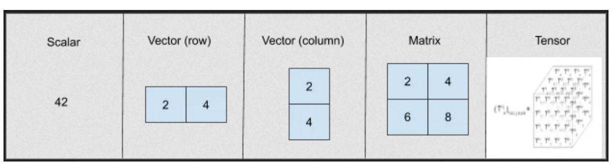

Figure 4.2: Representation of a tensor [1].

A TensorFlow project workflow typically follows these steps:

- Collecting data.
- Creating a model.

- Adding Layers to the model.
- Compiling the model.
- Training the model.
- Using the model.

TensorFlow operates on a graph-based model where mathematical operations form the nodes and tensors, being multi-dimensional arrays of data, act as the connecting edges. This structure enables machine learning algorithms to be mapped as a series of connected operations. A standard workflow in TensorFlow includes preparing the data, building the model, and then training it for prediction purposes. There are two main ways to feed data into the system: one is by defining a computational graph that outlines the entire data flow for the model's training, and the other is through eager execution, which processes operations immediately as they're called, following a more straightforward, line-by-line approach to programming [1].

In relation to this thesis, TensorFlow uses a computational graph to define and organize a series of operations that are executed together. This graph consists of a network of nodes, where each node is an operation and the edges represent the tensors that flow between these operations. The graph facilitates the management of dependencies and the optimization of computations, which is particularly beneficial for training complex models like neural networks. It allows TensorFlow to automatically compute gradients, which are essential for the backpropagation algorithm used in training.

Eager execution has been utilized for immediate evaluation of operations, allowing direct interaction with the execution of the program, without having to compute TensorFlow graphs. Eager execution is an imperative programming environment that evaluates operations immediately without building graphs. An approach that is more intuitive and easier for debugging since it executes operations, in regards to this thesis, standard Python code.

4.5.5 Keras

Keras, an open-source library, operates on top of TensorFlow, providing a high-level API that streamlines the development and testing of Deep Neural Networks. Its design emphasizes user-friendliness, modularity, and extensibility, making it highly suitable for Deep Learning applications.

Using Keras for Deep Neural Network construction allows for swift prototyping and experimentation with various model architectures, hyperparameters, and data preprocessing techniques. This flexibility helps in exploring numerous potential solutions efficiently. Keras' modular design simplifies the integration of neural layers, optimizers, and activation functions, and accelerates the development and fine-tuning of complex models, significantly reducing the time and effort required to go from conceptual design to functional implementation.

As a high-level interface for TensorFlow, Keras inherits TensorFlow's advanced capabilities, including scalability and support for distributed training, while offering a more intuitive and accessible user experience. This combination of TensorFlow's powerful backend and Keras' user-friendly frontend makes it an good alternative to work with Machine Learning for both beginners and experienced practitioners in the field of machine learning.

Keras also features comprehensive support for convolutional and recurrent neural networks, essential for tasks like image and sequence processing, respectively. Its compatibility with both CPUs and GPUs allows for versatile deployment and accelerates computational processes. Furthermore, Keras' extensive community support and continual updates ensure access to the latest advancements in deep learning, alongside a wealth of resources and documentation to aid developers in their projects. Keras, an open-source library, built on top of TensorFlow and offer a high-level API that simplifies the construction and experimentation of deep neural networks. Its design revolves around user-friendliness, modularity, and extensibility, making it an ideal choice in the field of deep neural network [37].

5

Requirements And Design

This chapter introduces the development of a Deep Neural Network (DNN) model designed to advance the accuracy of Peak Particle Velocity (PPV) predictions in drill and blast operations. By combining observational data provided with a DNN model, the primary objective is to improve the ability to predict blast-induced ground vibration, therefore prioritising safety and operational efficiency in the mining sector.

The design details the systematic progression from initial design to functional code, encompassing the iterative processes of model development, evaluation, and enhancement. The end product is a DNN prototype designed to predict a more accurate Peak Particle Velocity.

The chapter mentions the Machine Learning algorithm used, the type of optimization and regularization techniques used to ensure the correct results from training and validation of the Machine Learning model, and to preserve the model's ability of generalization.

5.1 Requirements

The formulation of a Deep Neural Network (DNN) for a mining application requires a extensive understanding of stakeholder needs. This emphasizes the research and development approach, as previously outlined in the thesis introduction 9.2. The requirements articulation statement contrains:

- **Data Quality and Quantity:** The model relies on a robust and high-quality dataset that contains blasting parameters relevant for predicting Peak Particle Velocity. The dataset should be large enough to train a Machine Learning model, with enough diversity among the data to enable an accurate and generalizable model. The data must be in a format ready for analysis or easily preprocessed, since these steps are necessary when feeding the data into a Deep Neural Network.

- **Model Performance:** The model should provide accuracy, by enable prediction of PPV within an acceptable error margin. The model should be reliable with consistent performance under varying operational conditions.

- **Security and Privacy:** The research should ensure confidentiality and integrity of sensitive data, with a clear understanding and agreement of the ownership and use of the development model and associated data.

- **Evaluation and Reporting:** Establishing clear evaluation metrics for evaluating the model's performance is essential for informed decision-making, and model comparison.

5.2 Architecture

The core of the system is a Deep Neural Network architecture, capable of capturing complex, nonlinear relationships within the data. The design is facilitated by the use of TensorFlow and Keras, offering a balance between ease-of-use and flexibility. This allows for the design of neural network layers and a seamless integration of activation functions, optimizers, and regularization methods to enhance model performance and generalization.

The Deep Neural Network (DNN) architecture consists of an input dataset, a trained output model, and multiple hidden Deep Neural Network layers of interconnected nodes. Each layer is designed to perform specific transformations on the input data. The architecture of the DNN can viewed in the figure below 5.1.

The number of neurons in the input layer of a neural network typically corresponds to the number of features in a dataset that are used to predict the output. The figure below 5.1 represents the Machine Learning model consisting of 4 layers and number of neurons. The first hidden layer consists of 256 neurons, the second hidden layer of 512 neurons, the fourth hidden layer of 128 neurons. The output layer has 1 neuron since the target is the Peak Particle Velocity (PPV).

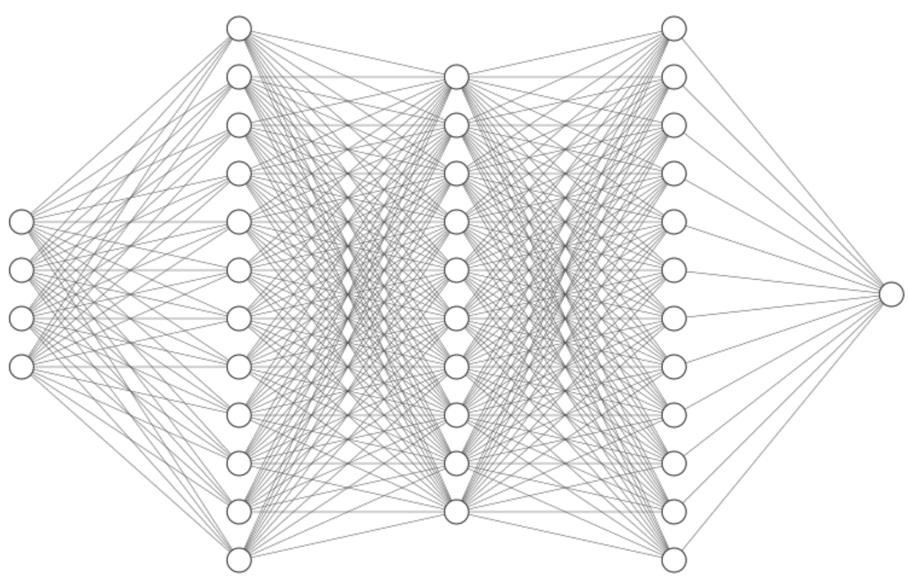

Figure 5.1: Example of a Deep Neural Network architecture.

5.2.1 Deep Neural Network Layers

In a Deep Neural Network (DNN), each layer performs a specific function in the process of transforming input data into meaningful output. As data is fed into the network, the first layer processes the raw input, and begins to recognize patterns within the extracted features, identifying relationships between features and passes the transformed information to the next layer.

Each subsequent layer builds upon the previous one, gradually abstracting and refining the data. This hierarchical process enables the DNN model to handle complex tasks like distinguishing between different features. The final layers integrates these insights to make accurate PPV predictions for each blast of a specific site.

The steps of this process is described here:

- **Input Layer:** This is the first layer that receives the raw input data. Its neurons pass the input data directly to the next layer, often without performing any calculations. The number of neurons in this layer corresponds to the number of input features.

- **Hidden Layers** These layers come between the input and output layers and are responsible for learning and extracting features from the input data. Each neuron in a hidden layer takes a weighted sum of the inputs from the previous layer and applies an activation function to produce an output.

- **Activation Functions:** Activation functions introduce non-linearity into the model, and are applied to the output of each neuron in the hidden layers. It enables neural networks to develop complex representations and functions based on the inputs. This thesis uses the activation function named ReLU (rectified linear activation unit).

- **Output Layer:** The final layer produces the network's output, which in this thesis is the prediction of Peak Particle Velocity (PPV).

- **Loss Function:** The loss function measures the error or the difference between the predicted PPV values and the actual PPV values in their training data. The loss

- **Optimization Function** Optimization algorithms are used to update the weights and biases of the network during training to minimize the loss function, and this thesis uses the optimization function named Adam.

- **Regularization Technique:** Regularization techniques involves dropout and regularization refereed to as L1 or L2 regularization to be applied to the hidden layers.

- **Hyperparameters:** The architecture includes hyperparameters such as the learning rate, batch size, and the number of epochs, which are the training iterations. Tuning these hyperparameters, which involves changing and testing these parameters for optimal results, is essential for achieving the best model performance.

- **Data Flow** The data flows through the network in a forward pass during training and inference. During training, backpropagation is used to calculate gradients and update the network's weights and biases.

- **Evaluation:** After training, the model is evaluated on a separate validation or test dataset to assess its performance using metrics like mean squared error or R-squared.

5.2.2 Machine Learning model

The Deep Neural Network (DNN) is given real world historical data, and in relation to the type of data and the size of the dataset, the creation of a Machine Learning model follows the approach described below:

- **Target Function:** The target function is conceptualized to estimate PPV from blast events, using historical data from controlled blasts. The function encapsulates the relationship between input features and the resulting PPV.
- **Specify Function:** The model is designed to learn complex functions that maps blast parameters to PPV. This involves selecting the most predictive features and defining the output variable structure to accurately reflect the PPV as a continuous value.
- **Deciding Representation:** The data is represented in a format valuable to neural network processing. This includes structuring input data as tensors within TensorFlow, handling various data types, and ensuring that the network architecture is appropriate for capturing the nuances of the problem.
- **Choice of Algorithm:** The algorithm selected is a deep learning approach with an ability to model complex, nonlinear relationships inherent in the prediction of PPV from blasting parameters., specifically a neural network algorithm facilitated by TensorFlow and Keras.

5.2.3 Data Collection

The dataset comes in a Comma-separated values (CSV) format, and consists of multiple blast operations from a mine site as mentioned in chapter 4, section 4.5. The dataset is tabular with 9724 rows of data points and the following 9 columns of blasting operation features:

- **Site:** A value representing the specific geographical location where the blasting event occurred. represented as a categorical and discrete numerical variable.

- **BlastID:** A unique numerical identifier assigned to each blasting event to differentiate it from others, represented as a categorical and discrete numerical variable.

- **Scaled Distance (SD):** A calculated value that normalizes the actual distance from the blast based on the amount of explosives used, represented by a numerical value.

- **Distance:** Physical distance from the point of the blast to the measurement or observation point, represented as a numerical value.

- **Maximum Instantaneous Charge (MIC):** A term in blasting operations that refers to the largest mass of explosive detonated within a specific, brief time of window.

- **Blast Direction:** The orientation category which the blast is directed, represented as a numerical variable.

- **Timeframe:** A parameter used to quantify MIC, or the timing of the blast within a sequence of multiple blasts, represented as a numerical variable.

- **Ground Water:** Represents a binary variable, indicating whether groundwater was present, (1) for, or not (0) at the blast site.

- **PPV:** PPV is the measure of the maximum speed at which particles move at a point as shock waves pass through after a blast. This column represents the target value aimed for prediction. It is a numerical value typically measured in mm/s.

5.3 Data Preprocessing

Data preprocessing in the development of a Machine Learning model affects model performance and effectiveness by improving the data quality. Preprocessing involves various techniques to transform raw data into clean data in a format , suitable for modeling.

Data preprocessing identifies numerical and categorical features, ensures that the dataset is free from missing values and normalizes numerical data by applying normalization.

5.3.1 Normalization

Normalization adjusts the scale of features so that the model is not influenced by the scale of measurement. Normalization does this by scaling individual samples to have unit norm, by adjusting the scale of features in the dataset to a uniform range. This ensures that all inputs are treated equally by the model and prevents features with larger scales from disproportionately influencing the training process.

5.3.2 Converts categorical variables

Neural network requires input to be numeric, and data preprocessing achieves this by converting categorical data into a numerical format. Categorical variables are transformed into a format that can be provided to Machine Learning algorithms to better predict the Peak Particle Velocity.

5.3.3 Feature scaling

Feature scaling is employed to standardize the independent variables of a dataset within a specific range, ensuring that all features contribute equally to the result. This standardization helps the machine learning algorithm converge more quickly, as it prevents features with larger numerical ranges from disproportionately influencing the learning process.

5.3.4 Data augmentation

Data augmentation increases the diversity of data available for training models without actually collecting new data. By adding noise to the data, so it learns the underlying patterns in the data, data augmentation ensures that the model become robust to slight variations. This also results in the model generalizing better to new, unseen data.

5.3.5 Supervised Learning

The goal of the Machine Learning model is to learn the mapping from the input data to the output data. The model is not using the labels, known as the PPV values during the actual prediction phase, but still requires them during the training phase to learn the relationship between the input features and the output. The model uses the original PPV values to adjust its weights and biases through a process called backpropagation to minimize the difference between its predictions and the actual values. Once trained, the model attempts to predict the PPV for new, unseen data based on the features provided.

5.3.6 Backpropagation

As the network makes its predictions, it receives feedback, often in the form of a loss function that measures the accuracy of its output against known data. This feedback helps the network adjust its weights and biases, the parameters that determine the strength of connections between neurons, and enables the network to continuously refine its internal parameters for better performance. Adjusting these parameters is a process known as backpropagation, where the network learns from its errors, tweaking itself to improve future predictions.

This is achieved by specifying initial values and conditions for the learning of the Deep Neural Network. By selecting particular initializers and regularizers, rules and constraints guides the DNN's learning process. Initializers, determine how the network's weights start out before learning begins. Regularizers, on the other hand, help prevent overfitting by penalizing the network for overly complex solutions that might not generalize well to new data.

5.3.7 Generalization

Generalization refers to the model's ability to adapt properly to new, previously unseen data. This ability is the difference between the training and test error, named generalization error. This is achieved through partitioning the data into three subsets, a training set, a validation set and a test set. The training set consists of the values the network is trained on. The validation set is used to tune hyperparameters, and the test set is used to measure the generalization performance. A separate training and test set ensures correctly generalizing by knowing that the model is not simply memorizing the training examples. If the algorithm works well on the traning set but fails to generalize, we say it is *overfitting*.

5.3.8 Overfitting

When the model's performance on the validation data begins to degrade, it is called overfitting. To prevent overfitting in Deep Neural Networks, weight regularization and dropout is added. With the use of TensorFlow and Keras, weight regularization is added by passing weight regularizer instances to layers as keyword arguments. Dropout is applied to a layer, and consists of randomly dropping out a number of output features of the layer during training, which involves setting numbers to zero.

The layer's output values are scaled down by a factor equal to the dropout rate, to balance that several units are active than at training time. Noise is introduced in the output values of a layer to break up the coincidence pattern that insignificant, as the network will memorize the patterns if no noise is present.

5.3.9 Machine Learning Algorithm

The architecture of the Deep Neural Network is constructed with fully connected layers, each employing the ReLU activation function explained in chapter 3 section 3.8 to introduce non-linearity, enabling the learning of complex patterns within the data.

Optimization during the training process is handled by the Adam optimizer explained in 3.11, a sophisticated algorithm that leverages first-order gradient-based optimization explained in 3.10, and adapts the learning rate for each weight of the model, which helps in navigating the stochastic nature of the learning process.

5.3.10 Model Training And Validation

During model training and validation, the preprocessed data is fed into the model in an iterative process known as training. During each iteration, also known as epoch, the model makes predictions based on the input data and then adjusts its internal parameters, named weights and biases, based on the difference between its predictions and the actual outcomes. The goal is for the model to learn the underlying patterns in the data so it can make accurate predictions on new, unseen data.

The validation step enables tuning the model's hyperparameters and for checking how well the model generalizes to new data. A separate validation dataset, unseen by the model during its training, is used to evaluate model performance. This helps in detecting issues like overfitting, where the model performs well on the training data but poorly on new data. The feedback from validation is used to adjust the model architecture or training process before the final evaluation.

5.3.11 Hyperparameter Tuning

Hyperparameter tuning in Deep Neural Networks (DNN) involves systematically experimenting with various combinations of hyperparameters to optimize the model's performance. Key hyperparameters in the context of the developed DNN model for Peak Particle Velocity (PPV) prediction include learning rate, batch size, number of epochs, and the architecture specifics like the number of layers and neurons in each layer.

The tuning process considers the balance between model complexity and computational efficiency, ensuring the model is sufficiently detailed without being computationally prohibitive.

In this thesis, hyperparameter tuning is focused on achieving the best possible predictive accuracy while maintaining a balance between underfitting and overfitting. The process includes fine-tuning regularization techniques, such as dropout rates and L1/L2 regularization strengths, to enhance the model's generalization capabilities.

5.3.12 Model Visualization And Evaluation

Model visualization and evaluation are taking use of analytical tools to interpret the model's predictions in relation to Peak Particle Velocity measurements. These tools are helping in understanding the model's decision-making process and ensuring technical validation by providing insights into the model's strengths and limitations, aiding in further refinement.

The evaluation phase involves testing of the model using a separate test dataset. This stage is for understanding how the model performs on data it has never encountered, which is a key indicator of its practical applicability. Various performance metrics, such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared value, are computed to quantify the model's prediction accuracy. Vsualization tools like correlation heatmap, plot graphs, and feature importance plots are employed to offer a more intuitive understanding of the model's performance. These tools facilitate the identification of areas where the model excels or needs improvement, particularly in terms of its predictive capabilities concerning different aspects of PPV in mining operations.

6

Implementation

This chapter provides an in-depth look at the development of a Deep Neaural Network (DNN) model, specifically designed for enhancing the predictive capabilities of blast-induced ground vibration. We present a prototype model that serves as a test case to explore the feasibility and advantages of integrating Machine Learning into mining practices. Key components of this chapter include a discussion on the choice of development tools, highlighting their suitability and effectiveness for the project.

The chapter also details the dataset employed, emphasizing the steps undertaken for effective data preprocessing, which is crucial for the accuracy and reliability of any ML model. The objective of this chapter is to offer a detailed narrative of the entire implementation process, spanning from the initial selection of tools to the concluding phases of model training and performance assessment.

6.1 Requirements

- **Data Quality and Quantity:** Have access to a high-quality dataset that contains blasting parameters relevant for predicting Peak Particle Velocity. The dataset should be large enough to train a Machine Learning model, with enough diversity among the data to enable a more ccurate and generalizable model.

The data should be in a format ready for analysis or easily preprocessed, since these steps are necessary when feeding the data into a Deep Neural Network.

- **Model Performance:** The model should provide accuracy, by enable prediction of PPV within an acceptable error margin. The model should be reliable with consistent performance under varying operational conditions, and provide fast computation to enable near real-time predictions if required.

- **Security and Privacy:** The research should ensure confidentiality and integrity of sensitive data, with a clear understanding and agreement of the ownership and use of the development model and associated data.

- **Evaluation and Reporting:** There should be clear evaluation metrics for evaluating the model's performance, with tools for generating reports that can assist in decision-making. This could include accuracy, precision and computational efficiency.

6.2 Tools For Development

The project was carried out using a combination of software and hardware that were vital at different stages such as preparing the data, building the model, training, and evaluating it. The software provided the necessary libraries to develop a Deep Learning Model, while the hardware offered the computing power needed for complex calculations throughout the thesis.

6.2.1 Software

The thesis involved using several tools to build and train the Deep Learning Model. The tools were important for different stages of the development process, including data preprocessing, model architecture design, training and evaluation. A Python virtual environment offered an isolated workspace for the Python project [47]. The isolation maintained project independence as well as protecting the integrity of the system. Other benefits of using a virtual environment were:

- **Dependency Isolation** Each project can have its own dependencies, irrespective of what dependencies of another project, and avoid interfering with the system-wide Python installation.

- **Version Control** Virtual environments facilitate the management of package versions. This is crucial for maintaining long-term projects that may not support newer package versions.
- **Reduced Risk** Isolating project environments helps to mitigate the risk of system-wide package conflicts. This is especially beneficial when different projects require different versions of the same package.
- **Reproducibility** Dependencies can be saved and easily shared, allowing other developers or systems to replicate the exact environment.

6.2.2 Hardware

The deep learning models in this project were trained using the following hardware specifications: Intel Core i7-4550U CPU, with a base frequency of 1.50GHz, with 2 cores and 4 threads, allowing for parallel processing, beneficial for training the Machine Learning model efficiently. The memory of the computer has a 4 MiB L3 cache, along with 512 KiB L2 and 64 KiB L1 caches per core, aids in fast data retrieval during intensive computational tasks like model training. With VT-x virtualization technology, the CPU supports virtual environments, useful for creating a isolated environment for the Machine Learning project. The 64-bit architecture of the CPU is ideal for handling large datasets, and the presence of advanced vector extensions (AVX) can enhance the performance of certain machine learning operations

6.2.3 TensorFlow And Keras

TensorFlow [1] and Keras [37] were employed as the core deep learning frameworks. TensorFlow, an open-source library, offered a comprehensive set of tools and functionalities for building and training Deep Learning Models. Keras, built on top of TensorFlow, provided an easy way of constructing neural networks, simplifying the implementation process and allowing for faster prototyping. TensorFlow and Keras allows for focusing on the architecture of the model rather than specialization in deep learning.

TenorFlow enables the balance between providing high-level ease of use through Keras and the low-level control necessary for customizing and optimizing models to meet specific research requirements. For training large neural networks, TensorFlow's ability to efficiently scale across multiple CPUs and GPUs is crucial. This scalability was benefitial for handling the development of a complex model and testing datasets with in variable size. Additionally, TensorFlow includes TensorBoard, a tool for visualization the steps of model development, from training to analysis.

6.2.4 Visualisation Tool

TensorBoard provided a dynamic interface to visualize various training metrics, including losses and accuracy's, as well as the evolution of the weights and biases for the Machine Learning model [68]. It helped understanding the complex Machine Learning architecture as Tensorflow allows for the visual representation of computational graphs with its underlying dataflow graphs. A key feature of TensorBoard is its ability to track training metrics like loss and accuracy in real-time, aiding in early detection of training issues such as overfitting or underfitting. TensorBoard enabled to compare metrics across different training runs, it helps identify the most effective hyperparameters, optimizing model performance.

6.3 Deep Learning Model Architecture

The architecture of the thesis is designed to form an end-to-end Machine Learning pipeline, utilizing the Python programming language and leveraging libraries like TensorFlow, Keras, scikit-learn, NumPy, pandas and Matplotlib.

Since our dataset has a uniform data type, it was possible to use pandas DataFrame anywhere it was possible to use a NumPY array. This works because the pandas DataFrame supports the array protocol, and TensorFlow accepts objects that support the protocol.

The data frame can be converted to a NumPy array, and then to a tensor. Converting an object to a tensor enables the object to be passed anywhere it is possible to pass a tensor. A Data Frame interpret as a single tensor, can be used directly as an argument to the model fit method.

The models objective is to predict an accurate PPV value with high accuracy by learning from the features provided from the given dataset.

6.3.1 Data Preprocessing And Data Handling

The DataHandler class is a critical component, designed to streamline the process of preparing raw data for input into the Deep Neural Network. The class is composed of several integral methods that facilitate data loading, preprocessing, splitting of data, and encapsulating the logic for loading the data required for training a Deep Neural Network.

The key functionalities of this class included:

- **Data Loading:** Data is loaded from a specified file path, containing various features such as ’Site’, ’Scaled Distance’, with the primary target variable being ’Peak Particle Velocity (PPV)’. During the preprocessing the data, The dataset is collected in a CSV file format, and the target variable "PPV." The data undergoes preprocessing, during which the ’PPV’ value is separated from the other features. This step essentially removes ’PPV’ from the list of features and retrieves the last feature in the list by default.

- **Data Splitting:** The dataset is split into 70% training, 15% testing and 15% validation to evaluate the model’s performance on unseen data.

- **Feature Engineering:** It involves creating new features or modifying existing ones to make them more useful for predictive modeling. The goal is to transform raw data into features that better represent the underlying problem to the predictive models, thereby improving model accuracy and performance.

  - **Polynomial Features:** Creates new feature variables out of the existing ones, representing their various combinations and interactions. This technique involves raising existing features to various powers (squares, cubes, etc.) and creating interaction terms (like the product of two features). This process is beneficial in modeling non-linear relationships within the data, which might not be captured by the original features. When using polynomial features in a Deep Neural Network, it adds complexity to the model, allowing it to learn more intricate patterns, but it also increases the risk of overfitting.

  - **One-hot encoding:** Converts categorical variables into a form that can be provided to algorithms. One-hot encoding is applied to categorical variables, by creating a new binary column for each category in the original data. The method does this by converting categorical variables into a binary (0 or 1) format. For each unique category in a feature, one-hot encoding creates a new binary feature.

- **Convert Numerical Columns:** Processes numerical data to be suitable for model training. This includes normalization or standardization, where data is scaled to a specific range or distribution. This process is crucial because it helps in dealing with variables that have different scales and distributions, ensuring that each feature contributes equally to the model’s learning process.

- **Data Integrity Checks:** Ensures that the data is accurate and consistent throughout the dataset, which is crucial for making reliable predictions. The data preprocessing steps includes validation to ensure all numerical columns are converted to float32 for consistency.
  - **Error Identification:** These checks help identify and correct errors or anomalies in the data, such as missing values, duplicates, or incorrect entries, which could otherwise lead to inaccurate model predictions.
  - **Data Loading Validation:** During the loading of the data a data integrity checks if the loaded data is empty, ensuring the dataset is valid before proceeding.
  - **Type Checking Before Preprocessing:** Before one-hot encoding and other preprocessing steps, the script checks the data type of training data, ensuring that the data types are consistent and suitable for the intended transformations.

Shuffling Data

After preprocessing and one-hot encoding the data, the data is shuffled to reduce variance and to ensure that the training of the model is not biased towards any particular order or pattern that might be present in the dataset. The shuffling of data presents the data in random order, which makes it more likely for the model to learn general patterns instead of memorizing specific sequences of data, which enhances the ability to generalize from training data to unseen data.

Overfitting occurs when a model learns noise and fluctuations in the training data. Shuffling reduces the risk of overfitting by making sure that patterns related to the order of data don't influence the learning process.

6.3.2 Dividing The Dataset

The dataset is divided into three sets, a training set, a testing set and a validation set. The training data set serves as the primary component during the models training phase, and it consists of 70 percent of the dataset. The testing data set plays a crucial role in the iterative refinement of the model.

The more diverse the training data is, the better the Deep Neural Network (DNN) model will perform. A diverse training dataset ensures that the model receives more discriminative information during training. Data augmentation is used to increase the diversity of the training dataset by expanding the size of the training dataset. Each time a training sample is exposed to the model, random noise is added to the input variables which makes them different every time they are exposed to the model.

A manual split of the data is performed, allocating 15 percent for testing set and 15 percent for validation set. The partitioning facilitates practical model training, testing and evaluation, and is described below:

- **Training Dataset Features:** This is a set of features representing the training data set, and is named 'training dataset'. It contains the input data that the Machine Learning model will use to learn patterns and make predictions. It is loaded with the features of the training dataset after preprocessing, such as one-hot encoding and shuffling.

- **Training Data Label:** This variable represent the training data, named 'training data label PPV'. It contains the actual measured PPV value from the original data set that the model will try to predict during training. This variable is loaded after preprocessing and aligning with the training features.

- **Testing Data Set:** This set represents the unseen testing data features after the model is trained on the original data set and the target variable, and is named 'testing dataset'. This is a new set of features that the model has not seen before. Similar to 'training dataset', it contains the input data, but is used to evaluate the model's performance after training, and therefore loaded with the features of the testing dataset after preprocessing. The testing set assess the performance of the model after training and provides an unbiased evaluation of the final model fit.

- **Testing Data Label:** This variable represents the true data, and is named 'testingdata label PPV'. It contains the corresponding target values for the testing dataset, which the model tries to predict during evaluation.

- **Validation Data Set:** This set represents the validation data, and is called 'validation dataset'. The validation set provides an unbiased evaluation of the model on the training data set while tuning the model's hyperparameters.

- **Validation Data Label:** This variable represents the validation data labels, and contains the corresponding target value (PPV), and is named 'validationdata label PPV'.

6.3.3 Internal Parameters

To enhance the model's learning capability, its internal parameters commonly known as weights and biases is adjusted:

Weights
Initially, the weights in the Deep Neural Network is set to small random values, to break symmetry and ensure that the learning process proceeds. During training of the data, the network adjusts the weights based on the errors in its predictions. When an input data is passed through a neuron, it is multiplied by the weight of the connection.

During training, input data is passed through the network with forward propagation. Each neuron computes a weighted sum of its inputs, adds a bias, and applies the ReLU activation function. The final output of the network is compared to the desired output, and the difference is measured using a loss function.

The goal of learning is to minimize this loss. Backpropagation is used to calculate the gradient of the loss function with respect to each weight in the network. This involves applying the chain rule from calculus 3.3 to compute the contribution of each weight to the error. Once the gradients are computed, an optimization algorithm named Adam is used to adjust the weight to modify each weight in the direction that reduces the loss.

This process of forward propagation, backpropagation, and weight adjustment is repeated over many iterations (or epochs) across the entire training dataset. The goal is for the loss function to converge to a minimum value, indicating that the network's predictions closely match the actual outputs. The weights at this point are considered to have been 'learned'.

Biases
Biases are added to the weighted sum before passing it through the activation function. They enable the network to represent patterns that do not necessarily pass through the origin if plotted on a graph. In essence, biases allow the network to shift the activation function to the left or right, which is crucial for fitting the data correctly. This allows the neurons to activate even when the weighted sum of inputs is not sufficient. Biases represent unique values for each neuron that is added to the product of inputs and weights before the ReLu activation function 3.3.11 is applied.

6.3.4 Layers

The layers are of the Deep Neural Network model is distributed with dropout layers, which randomly deactivates a proportion of neurons during training to avoid overfitting. Dense layers are fully connected, meaning each neuron in one layer is connected to all neurons in the next layer, facilitating a comprehensive flow of information. The number of neurons per layer is calibrated to balance the model's capacity to learn complex relationships without overfitting to the training data.

The activation function called ReLU for intermediate layers was chosen to introduce non-linearities into the model, enabling the model to learn complex functions.

ReLU, which stands for Rectified Linear Unit is defined as:

$$f(x) = \max(0, x)$$

ReLU handles input and outputs by if the input is positive, ReLU will make the output the same as the input, but if the input is negative, the output will be zero. This linear function introduces non-linearity into the Machine Learning model, allowing it to learn complex patterns, as well as contributing to the model's ability to capture the nonlinear relationship between the input features and the output PPV.

6.3.5 Data Augmentation

Data augmentation is used to increase the diversity of the training dataset, by applying noise. Random Gaussian noise is added to expand the size of the training dataset, also known as the input layer of the Deep Neural Network. Each time a training sample is exposed to the model, random noise is added to the input variables which makes them different every time they are exposed to the model.

A Gaussian distribution of the same size as the data, the first value of the distribution is added to the first value in the data, then the second value in the distribution is added to the second value in the data, and this keeps on until the end of the training dataset.

6.3.6 Data Normalization

Data Normalization transforms the set of data to be on a similar scale. For Machine Learning models, the goal is usually to recenter and rescale the data, which is accomplished by calculating the mean and the standard deviation on the set of data and transform each sample by subtracting the mean and dividing by the standard deviation. This method standardizes the data and achieves a standard normal distribution.

A layer called batch normalization layer, placed after the dense layers, standardizes the activations from the previous layer, to accelerate training and improve performance. Normalization ensures that each input feature contributes equally to the ability of the model to learn, preventing features with initially large ranges from outweighing those with smaller ranges. This is particularly important when input parameters like charge weight or distance to the monitoring point vary widely. The set of features is normalized before being passed through the layers.

The Machine Learning model is manually build before adapting the normalization layer to make the data preprocessing steps explicit, which enables efficiency, instead of using a callback to adapt it during the first epoch. This solution enables compatibility since building the model manually will make it less likely to cause compatibility issues with other callbacks or model configurations added in the future. It also provodes less Overhead with not using a custom callback for something as fundamental as normalization, since that can add unnecessary complexity and overhead to the training process.

6.3.7 Optimizer and Loss Function

The model employs the Adam optimizer and uses a combination of the loss functions, Mean Absolute Error (MAE) and Mean Squared Error (MSE).

Mean Absolute Error 3.18 measures the average magnitude of errors in the set of predictions, without considering their direction, and is less sensitive to outliers than MSE as it does not square the error values. MAE contributes robustness to the model, particularly important in geophysical applications where data can be noisy or have outliers due to measurement errors or environmental factors.

Mean Square Error 3.16 calculates the average squared difference between the estimated values and the actual value, which is useful when large errors are undesirable. Squaring the errors leads to a loss surface that is generally convex which simplifies finding the global minimum during the training process.

The combination of MAE and MSE ensures a comprehensive error representation for PPV predictions that require high accuracy due to their implications in safety and environmental impact, and may lead to better generalization of unseen data. The combination of MSE and MAE captures the average error magnitude (MAE) and giving more weight to larger errors that could have significant impacts (MSE).

Computing the loss gets the gradients to model weights and updates those weights accordingly through backpropagation. The loss is calculated and the Deep Neural Network is updated after every iteration until model updates do not reach higher improvements in the desired evaluation metric.

The model propagates the total loss back into the Deep Neural Network to know how much of the loss every neuron is responsible for, and subsequently updates the weights to minimizes the loss by giving the neurons with higher error rates lower weights, and the nodes with lower error rates higher weights.

Backropagation fine-tunes mathematical weight functions and improves the accuracy of an the Deep Neural Networks output. After each forward propagation passes through a network, the backward propagation adjusts the models parameters based on weights and biases.

6.3.8 Forward Propagation

The forward propagation computes the output from the input data, by starting at the input layer and move through each hidden layer sequentially, ending at the output layer. In each layer, the input data is transformed using a combination of weights, biases, and activation functions. This involves multiplying the input data by the weights, adding the biases, and then applying the activation function. This process is repeated for each layer until the final output is produced. The forward propagation is essential for both making predictions and initiating the backpropagation process during training.

6.3.9 Backpropagation

The backpropagation algorithm works by computing the gradient of the loss function with respect to each weight of the neurons, computing the gradient layer by layer, iterating backward from the last layer to the input layer to avoid redundant computation of intermediate terms.

Backpropagation calculates the gradient of the loss function relative to each neuron's weights and biases. This is achieved by multiplying the derivatives across each layer and modifying the partial products to update the weights. Backpropagation works by propagating errors backward, starting from the output nodes towards the input nodes. This process identifies where the model is making errors and adjusts accordingly to improve accuracy.

6.4 Regularization

Regularization parameters and learning rates are configurable, making the Machine Learning model flexible for optimization challenges. Regularization Techniques are employed on all dense layers to penalize large weights, thus discouraging overcomplex models that could overfit the training data. The degree of regularization is controlled by the regularization factor parameter, allowing for fine-tuning based on the specific dataset and training dynamics. The initialization of each layer is dynamically adapted to accept a different index from the regularization factor list, enabling different values for each layer in the Deep Neural model.

To prevent overfitting, where the model learns the training data too well and fails to generalize to new data, the training process is augmented with an early stopping mechanism, which involves applying regularization techniques. Layers introduce dropout, which randomly 'drops out' a proportion of layer outputs during training, forcing the network to learn redundant representations and therefore be more robust. The final layer of the neural network is a single neuron, designed to output the predicted PPV value.

6.4.1 Training Process

During training, the model uses the Adam optimizer to iteratively adjust the weights of the network to minimize the Mean Squared Error. This involves feeding the normalized features through the network, applying the ReLU activation, and then using the optimizer to update the weights based on the calculated loss. This process is repeated for a number of epochs until the loss on the validation set no longer improves, indicating that the model has learned to predict PPV as accurately as possible, given the architecture and data.

In parallel, the model checkpoint strategy is deployed. This involves saving the state of the model at its peak performance on the validation data. By doing so, it ensures that the most effective version of the model is preserved, regardless of subsequent fluctuations in performance as training progresses.

6.4.2 Hyperparameter Tuning

Hyperparameter tuning is managed through a separate script that allows for the tuning of various parameters listed below. The table 6.1 provides an overview of the hyperparameters of the model, and the description of them.

Hyperparameter	**Value**
Custom Input Shape	13
Learning Rate	0.001
Dropout Rate	0.1
Regularization Factor	0.001
Kernel Initializer	glorot uniform
Bias Initializer	zeros
Activity Regularizer	l1, l2
Kernel Constraint	MaxNorm
Bias Constraint	NonNeg
Noise Level	0.01
Epochs	50

Table 6.1: Hyperparameters and their values.

- **Custom Input Shape:** Number of features in the dataset.
- **Learning Rate:** Typical starting value. If the model learns too slowly, this should be increased, or decreased in case of the opposite.
- **Dropout Rate:** This helps prevent overfitting and is adjusted based on the complexity of the model and the amount of training data.
- **Regularization Factor:** Originated with small values for every layer. Regularization applies penalties on layer parameters or layer activity, effectively simplifying the model.
- **Kernel Initializer:** 'glorot uniform' is a default choice for ReLU activations. Initializes the weights of a layer uniformly at random, but within a range that depends on the number of input and output neurons of the layer.
- **Bias Initializer:** Currently set to be 'zeros'. Can be changed to 'non-zeros' to break symmetry in learning.
- **Activity Regularizer:** The model uses 'l1' or 'l2' regularizers as it can help make the model's output more infrequent.
- **Kernel Constraint:** In case of a reason to restrict the range of weights or biases, currently set to 'MaxNorm'.
- **Bias Constraint:** In case of a reason to restrict the range of weights or biases, currently set to 'NonNeg'.
- **Noise Level:** Helps in making the model robust to variations in the input data.
- **Epochs:** An epoch is simply one stream of the entire dataset.

7
Experiment

This chapter presents the experimental process used to develop and evaluate the effectiveness of Machine Learning (ML) algorithms in improving Peak Particle Velocity (PPV) predictions for blasting events in the mining industry. The primary focus is on the implementation of ML models and comparing their predictive capabilities against the industry standard practices.

The key sections in this chapter includes the ML model training process, and evaluation metrics, which are essential for understanding model optimization and performance measurement. The Machine Learning model's performance is compared against the industry-standard model, establishing a baseline for innovation. Two prototype models named Simple Neural Network model, and a Linear Regression Random Forest model were created to enable an comparison to the Machine Learning model.

7.1 Dataset Partitioning: Training, Validation, and Testing

The original dataset is split into training (70%), validation (15%), and testing (15%) sets to ensure robust model training and evaluation. The rationale behind these splits is to ensure that the model not only learns well but also generalizes well to new data and is not biased or overfitted to the training data.

The percentages are not strict rules but common practices. Depending on the size and nature of the dataset, these ratios can be adjusted. The key is to provide the model with enough data to learn effectively while also keeping a substantial amount of unseen data for validation and testing.

- **Training set (70%):** This is the largest portion of the data and is used to train the model. The high percentage ensures that the model has access to a diverse and extensive range of data points to learn from. Training on a substantial dataset helps the model capture the underlying patterns and relationships effectively. However, training exclusively on this set can lead to overfitting, where the model performs well on the training data but poorly on unseen data.

- **Validation Set (15%):** The validation set serves as a check during training. It is not used to train the model but to evaluate its performance after each training epoch. This allows for monitoring the model's generalization capabilities and tuning hyperparameters without biasing the model towards the test data. The validation set helps in deciding when to stop training with early stopping and to avoid overfitting.

- **Testing Set (15%):** After the model is trained and hyperparameters are tuned using the training and validation sets, the testing set is used to evaluate the model's performance. This set is never seen by the model during training and serves as new, unseen data. It provides an unbiased evaluation of the final model's performance and generalization ability. The performance on the testing set is a good indicator of how the model will perform in real-world scenarios or on unseen data.

7.1.1 The Impact Of Feature Correlations

Understanding the dataset and the relationship between its features is crucial for creating a Machine Learning model to accurately predict Peak Particle Velocity.

In figure 7.1 the dataset shows negative correlation between Scaled Distance (SD) and Peak Particle Velocity (PPV). As the SD increases, the PPV generally decreases, which is expected as the energy from the blast dissipates with increased distance or lower explosive mass.

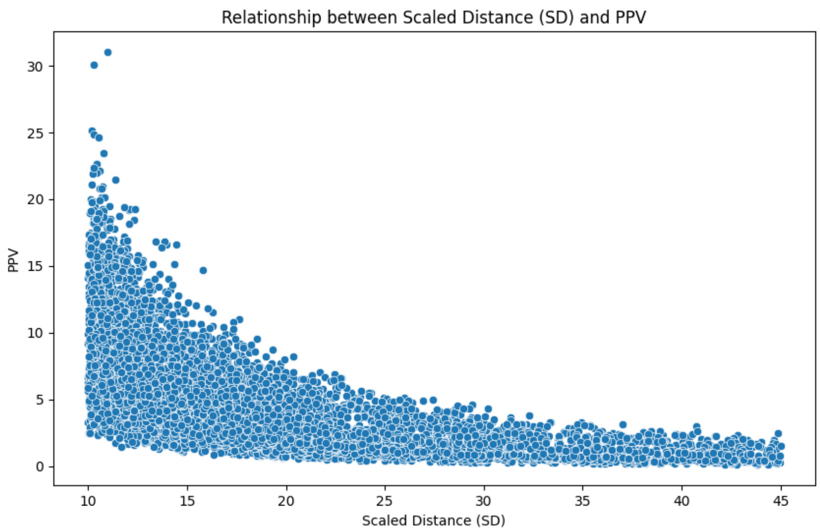

Figure 7.1: The relationship of Scaled Distance and PPV.

- **Clustering of Data Points:** The data points are clustered more densely at lower PPV values. As the Scaled Distance increases beyond a certain point (approximately SD > 30), the PPV values cluster closer to the lower end of the PPV scale, which indicates a rapid decrease in PPV as distance increases.

- **Data Distribution:** The PPV values have greater variance at lower Scaled Distances, showing high variability in PPV at lower Scaled Distance values. This variability decreases as scaled distance increases, suggesting more consistent (and lower) PPV measurements at greater distances.

- **Minimum and Maximum Values:** The plot reveals a noticeable lower boundary for Scaled Distance at approximately 10 kg/m$^{0.5}$. According to information provided by the data source, this boundary likely results from an enforced design limit.

In the figure below 7.2 there does not appear to be a clear trend or strong correlation between Maximum Instantaneous Charge (MIC) and PPV, The data points are widely spread across the range of MIC values. There's significant variability in PPV at all levels of MIC. This suggests that MIC is not the sole factor influencing PPV, or that other variables at play may affect the PPV.

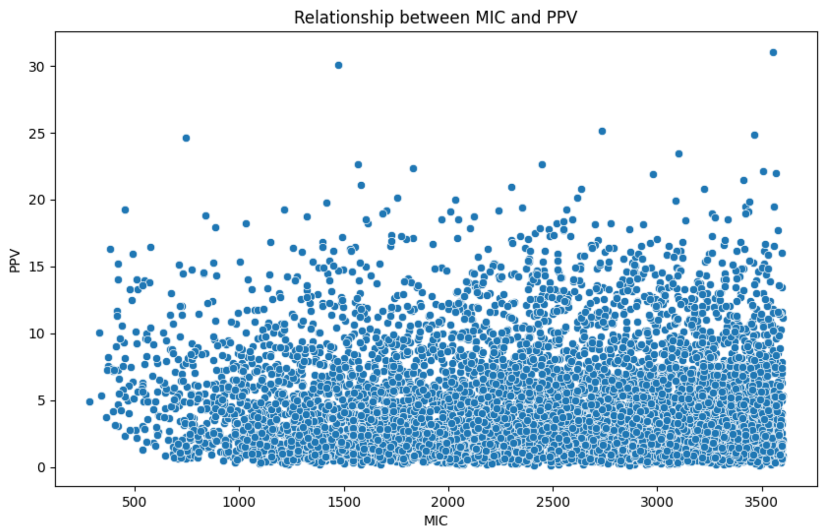

Figure 7.2: The relationship of Maximum Instantaneous Charge (MIC) and PPV.

Interpretation of the relationship of 'Scaled Distance' and PPV:

- **Implications for Blasting Practices:** Given the spread of data, it's possible that other factors, such as blast design, timing, or geology, may have a more significant impact on PPV than 'MIC' alone. This indicates the complexity of predicting PPV and the need for more sophisticated models that can account for multiple variables.

- **Potential for Model Improvement:** The wide dispersion of PPV values at various MIC levels could indicate that a simple linear model may not be the best predictor of PPV. This might suggest the exploration of more complex models or the inclusion of additional explanatory variables to better understand the determinants of PPV.

7.2 Iterative Model Development and Prototyping

The development of the Deep Neural Network (DNN) was an iterative process, involving the incremental creation and comparison of various simpler models, before the final DNN model was established.

Initially, a Simple Neural Network model was developed, featuring one input layer and three hidden layers. This model, while basic in terms of depth and complexity, effectively balanced expressiveness and simplicity, reducing the risk of overfitting. However, its predictive performance was not sufficiently compelling to warrant further development, and time allocation.

Subsequently, a more complex model was created, employing a two-stage approach that combined a RandomForestRegressor and a LinearRegression model. This model offered a moderate complexity level and was designed to capture more intricate data patterns through the RandomForest's ability to model complex interactions. Despite its advancement over the simpler model, its predictive capabilities still fell short of expectations, leading to a decision against further refinement, and time allocation.

The final phase of the experimental process focused on the Deep Neural Network (DNN) model. This model marked a significant advancement in complexity compared to its predecessors, with extensive customization options and an advanced architecture. It was characterized by its adaptability, incorporating adjustable regularizers, constraints, and initializers to accurately model complex data intricacies. The iterative refinement of this DNN, including hyperparameter tuning and performance assessment, resulted in noticeably superior predictive outcomes compared to the earlier models, making it the chosen prototype for the study.

The experimental process is illustrated in 7.3 below.

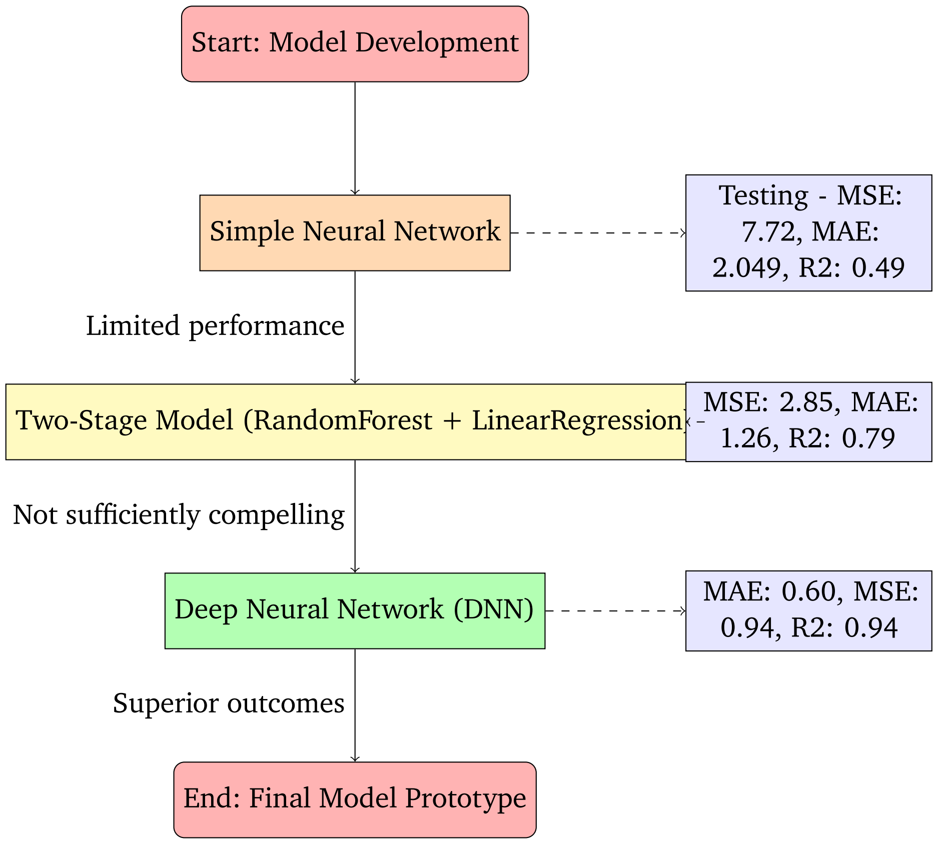

Figure 7.3: Iterative Model Development and Prototyping Flowchart

7.2.1 Model Parameters and Architecture

The architectures and parameters of the developed models are summarized in Table 7.1 below. Each model's architecture is unique in attempting to best predict Peak Particle Velocity (PPV). Understanding the structure and parameters of these models provides a simple insights into their functioning and adaptability to predict PPV. Additionally, the selection and tuning of hyperparameters play a significant role in optimizing model performance and ensuring generalization beyond the training dataset.

Model	**Simple Neural Network**	**Two-Stage Model**	**DNN**
Architecture	1 Input, 3 Hidden	RandomForest + Linear Reg.	Multiple Dense Layers
Layer Types	Dense	RandomForest, Linear	Dense, Dropout
Activation Func.	ReLU, Linear (output)	N/A	ReLU, Linear (output)
Hyperparameters	Learning Rate: 0.01, Neurons: 64, Epochs: 100	Trees: 100, Max Depth: 10	Learning Rate: 0.001, Dropout: 0.1, Epochs: 50
Hyperparam. Relevance	Learning speed, convergence, training duration	Complexity, accuracy, model depth	Overfitting, learning optimization, training duration

Table 7.1: Summary of Model Parameters and Architectures

8
Evaluation

This chapter focuses on evaluating the predictive performance of the developed Machine Learning models for Peak Particle Velocity (PPV), with a primary emphasis on evaluating the advanced DNN model and comparing its results with those of the industry-standard model.

8.1 Evaluation of Key Variables in Mining Blast Operations

The correlation heatmap below 8.1 provides a visual representation of the relationships between various variables from the raw dataset.

Strength of Correlation:
Intensity of the color indicates a stronger correlation between features. Stronger correlations are represented by more intense colors, and weaker correlations are shown with less intense, more muted colors.

Strength of Correlation:
One color spectrum represents positive correlation, while another spectrum represents negative correlation. No correlation or zero correlation is usually shown with a neutral color.

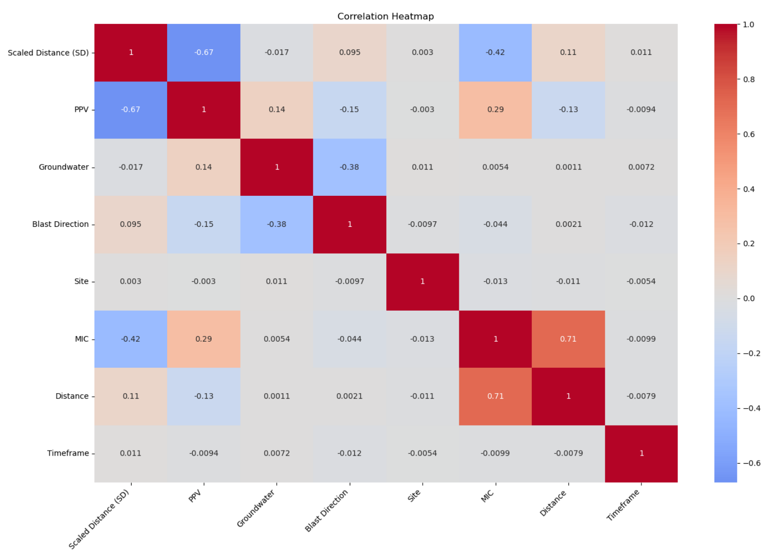

Figure 8.1: Displays the correlation between the variables as a color-coded matrix.

The key features included in this heatmap are:

- Scaled Distance (SD)
- Groundwater Presence
- Blast Direction
- Site Location
- Maximum Instantaneous Charge (MIC)
- Distance to Monitoring Point
- Timeframe

Correlation Heatmap Analysis

The scale on the right of the figure indicates that red colors represent positive correlation and blue colors represent negative correlation. The light grey color represent no correlation.

- **Scaled Distance (SD) and PPV (Peak Particle Velocity):** There is a strong negative correlation (-0.67), indicated by the dark blue color. This suggests that as the 'Scaled Distance' increases, the Peak Particle Velocity tends to decrease, or vice versa.
- **Groundwater and Blast Direction:** There is a moderate negative correlation (-0.38), indicated by the medium blue color. This suggests some level of inverse relationship between groundwater levels and the direction of the blast.
- **MIC (Maximum Instantaneous Charge) and Distance:** There is a strong positive correlation (0.71), indicated by the dark red color. This indicates that as the 'MIC' increases, the distance also tends to increase, or vice versa.
- **Site:** The correlations with the 'Site' variable are very weak, as indicated by the very light colors. This suggests that 'Site' has little linear relationship with other variables in this dataset.
- **Timeframe:** Like 'Site', 'Timeframe' also shows very weak correlations with other variables, suggesting it does not have a strong linear relationship with the measures considered in this heatmap.

Heatmap correlation does not necessarily imply causation, further analysis is required to determine the underlying causes of these relationships.

8.2 Evaluation of the Industry-standard Model

Chapter 2 introduced the industry-standard model used for predicting Peak Particle Velocity (PPV) along with its underlying mathematical representation. PPV measures the maximum velocity attained by particles in a blast, across transverse (T), vertical (V), and longitudinal (L) axes, as referenced in [19].

The industry-standard model computes PPV using the Duvall and Fogleson formula from the United States Bureau of Mines (USBM), articulated in Equation 8.1, reproduced below for clarity:

$$\text{Duvall and Fogleson (USBM)} \quad PPV = K\left(\frac{D}{Q^{1/2}}\right)^{-b} \tag{8.1}$$

In this equation, Q denotes the Maximum Instantaneous Charge, and D refers to the separation between the blast site and the monitoring point. The constants K and b are derived through the regression analysis of the dataset, which encapsulates the relationship between 'MIC', 'Distance', and the observed PPV. It is important to note that 'Scaled Distance (SD)' is defined as:

$$\text{Scaled Distance Equation} \quad SD = \left(\frac{D}{Q^{1/2}}\right) \tag{8.2}$$

The industry-standard predictive model relies on accurately defining constants K and b, in Equation 8.1. To determine these constants, a regression fitting technique is applied to the training dataset, where Scaled Distance (SD) is the independent variable and PPV is the dependent variable. This technique is depicted in Figures 8.2 and 8.3.

The least squares regression method was used to minimize the sum of squared residuals, leading to the determination of K and b. Specifically, on the logarithmic scale, the slope of the regression line corresponds to constant b (-1.67), reflecting the rate of PPV decrease with increasing SD. The y-intercept of this line provides constant K (487), indicative of the initial PPV at a unit Scaled Distance. These constants, critical for the model's accuracy, allow for PPV predictions to be made when 'MIC' and 'Distance' are known.

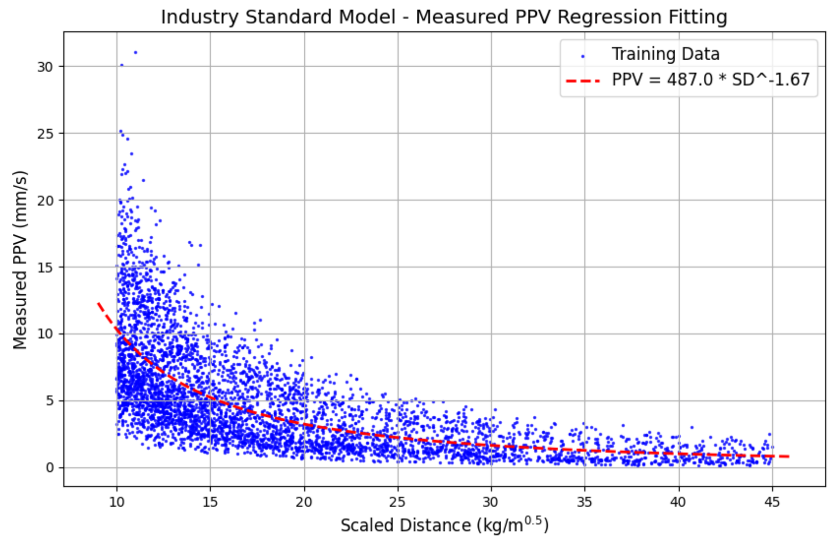

Figure 8.2: Industry Standard method, displaying the fitted regression line amongst the training dataset.

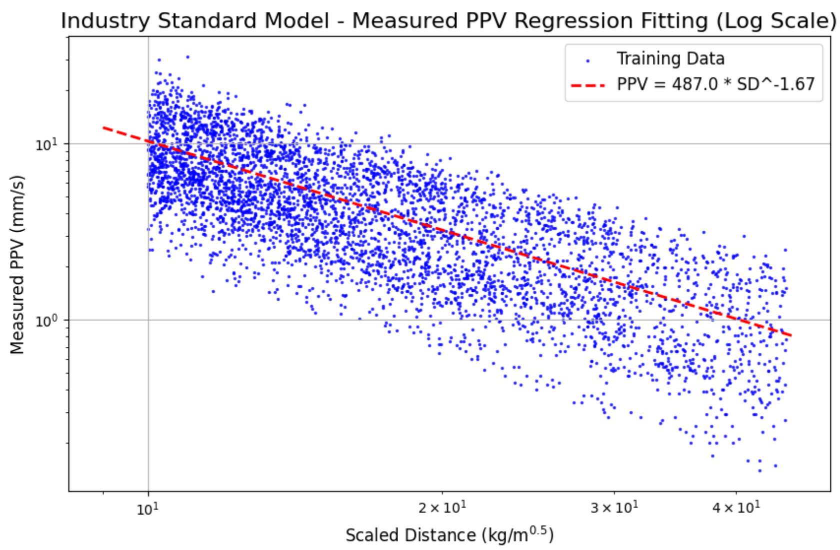

Figure 8.3: Industry Standard method displayed on a log scale, indicating the regression line which is used to determine the K and b constant values.

After defining the industry-standard model, by determining the *K* and *b* values using the regression method, the model was applied to the testing dataset, these results are shown below in Figure 8.4.

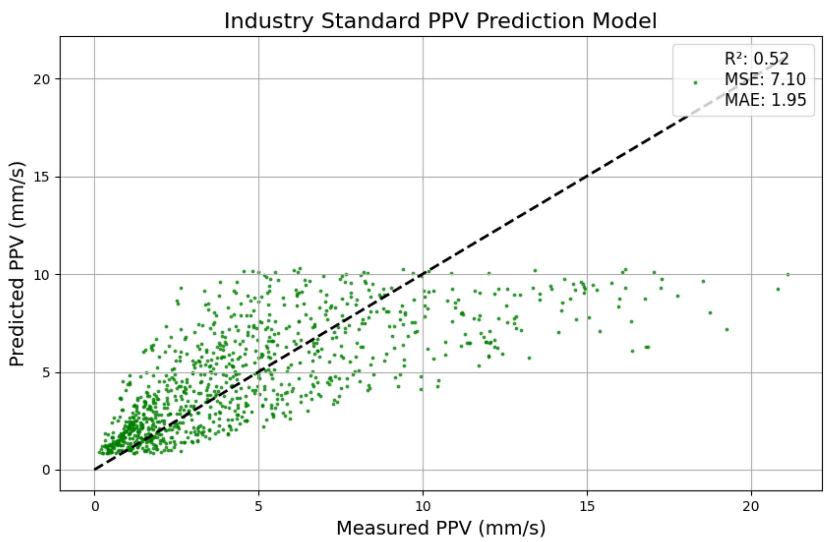

Figure 8.4: Predicted vs. Measured PPV Comparison Using the Industry Standard Model.

This visual representation is valuable for evaluating the performance of the industry-standard model. It also serves as a comparative baseline when compared against the advanced capabilities of the Deep Neural Network developed in this research. The evaluation metrics for the industry-model's performance are summarized in the table below.

Evaluation Metric	Industry Standard Model
Mean Absolute Error (MAE)	2.04
Mean Squared Error (MSE)	7.41
R-squared	0.51

Table 8.1: Evaluation results of the industry-standard model.

8.3 Evaluation of the Machine Learning Models

8.3.1 Deep Neural Network Model

In the development of a Deep Neural Network (DNN) model for predicting Peak Particle Velocity (PPV) in mining operations, understanding the features importance is crucial for interpretability and the optimization of the model. Figure 8.5 presented below is a visual representation of feature importance derived from the model after training, showing the relative contribution of each feature to the predictive ability of the network.

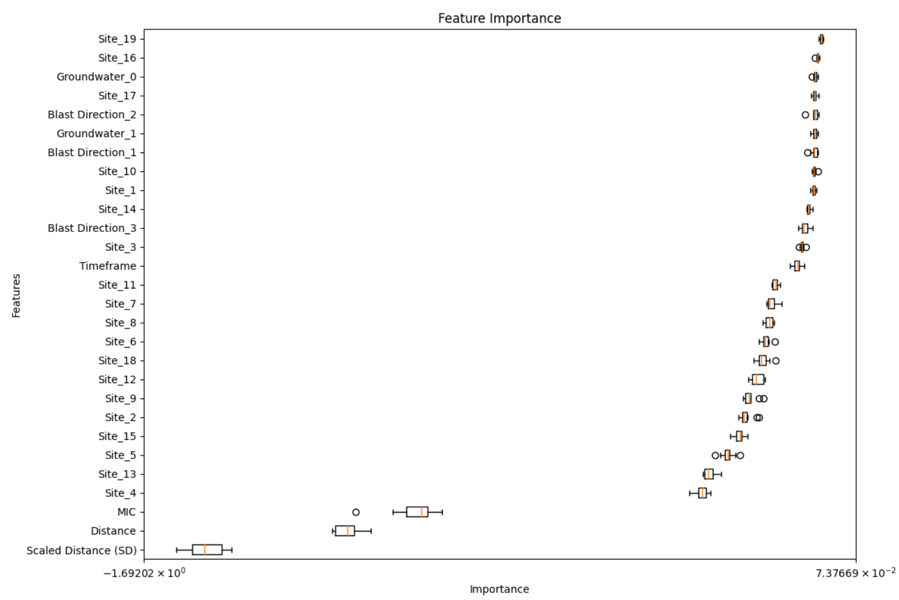

Figure 8.5: Feature importance identified in the DNN model.

Error bars in feature importance plots are a way to visualize the stability or consistency of the feature importance. The x-axis shows the scale of importance, with values close to zero indicating lower importance and higher values indicating greater importance.

'Scaled Distance' is identified as having the greatest importance in the DNN. This is not surprising as it this specific parameter is the sole parameter of significance in the current industry-standard model, and therefore validating it's significance. 'Distance', similar to 'MIC' shows great importance to the DNN model, but to a lesser extend than 'Scaled Distance (SD)'variable. The multiple 'Site' variables have lesser importance in the model, as they are likely encompassing the geological ground conditions.

The categorical variables, such as 'Blast Direction', 'Groundwater', 'Timeframe' and 'Site', have been one-hot encoded, as evidenced by the multiple entries for each in the chart.

The predictive performance of the DNN model is depicted in Figure 8.6. The scatter plot demonstrates a strong correlation between the predicted and measured PPV values, as evidenced by the high R-squared value of 0.94. The proximity of data points to the dashed line of perfect prediction indicates the model's accuracy, with a Mean Squared Error (MSE) of 0.94 and a Mean Absolute Error (MAE) of 0.60 mm/s. Such metrics underscore the DNN model's effectiveness in generalizing from the training data to make reliable predictions on unseen data.

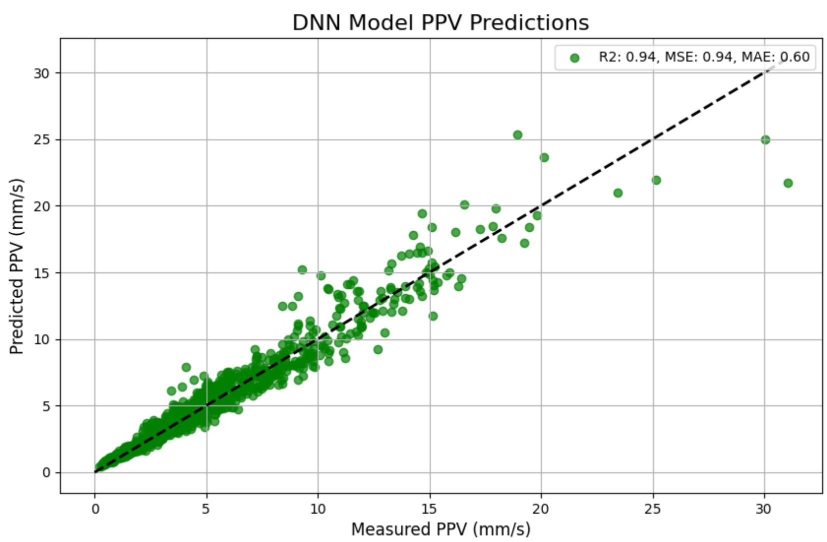

Figure 8.6: DNN model PPV predictions on testing data.

8.3.2 DNN Evaluation Metrics

Evaluating the model's performance was quantified using several statistical metrics. These are considered essential in validating the effectiveness of the Machine Learning model, and its results is presented in the table, together with the Industry Standard Model results which serves as a comparison.

Evaluation Metric	DNN Model	Industry Standard Model
Mean Absolute Error (MAE)	0.60	2.04
Mean Squared Error (MSE)	0.94	7.41
R-squared	0.94	0.51

Table 8.2: Evaluation results of the Machine Learning model compared to the industry standard model.

- **Mean Absolute Error (MAE):** The MAE, which represents the average absolute discrepancy between the predicted values and the actual data points was calculated to be approximately 0.60. This indicates that, on average, the model's predictions deviate from the true PPV values by less than one unit. This level of accuracy is promising for practical applications where estimations within a margin of less than one are often acceptable.

- **Mean Squared Error (MSE):** The MSE, providing a squared average of prediction errors, was determined to be 0.94. The squaring of errors penalizes larger discrepancies more severely. A lower MSE is desirable. An MSE value close to one suggests that the model's predictions are, in general, close to the observed values, with fewer large errors.

- **R-Squared (R^2):** The R^2 metric is a statistical measure that represents the proportion of variance for the dependent variable that's explained by the independent variables in the model. In this context, an R^2 value of approximately 0.94 indicates that the model explains over 94% of the variance in PPV from the features provided. This high R^2 value demonstrates a strong relationship between the model's inputs and the predicted PPV, suggesting that the model is highly predictive of the expected PPV outcomes.

8.3.3 DNN Model Training And Validation Loss Evaluation

Training Loss represents the model's performance on the training set. It's the error between the predicted values and the actual values for the training dataset. The steady decline and stabilization indicate that the DNN model has learned effectively from the training data.

The primary goal during training is to minimize this loss. A lower training loss indicates that the model is learning the patterns in the training data effectively, and is calculated using the loss functions Mean Square Error Equation 3.16, the Root Mean Square Error Equation 3.4.3, and R-squared Equation 3.19.

Validation Loss is the error on a separate set of data not used for training, typically used to evaluate the model's ability to generalize. The fact that it follows the training loss closely and doesn't increase suggests that the model is not overfitting and has good generalizability.

The purpose of training validation is to identify issues like overfitting, where the model performs well on training data but poorly on unseen data. A model that generalizes well will have a validation loss comparable to its training loss. The validation loss is calculated using the same loss functions but on the validation dataset.

Figure 8.7 shows the training and validation loss for the DNN model.

Figure 8.7: Training and validation loss results of the DNN model.

Key results from the training and validation loss shown in Figure 8.7 are summarized below:

- **Epochs:** 50 epochs are used based on the results from the training and validation loss Figure 8.7, where the losses are fairly flat in the final epochs, towards the end of the training. The losses flattens between 30-50 epochs, suggesting that running more epochs may not result in significant improvements in the model performance.

- **Learning Dynamics:** Initially, both training and validation loss values are high as the model starts learning. There is a sharp decline in both training and validation loss as the model begins to learn from the data. This is most pronounced in the first 10 epochs. As training progresses, the training loss typically decreases. However, the behavior of the validation loss is crucial to monitor for overfitting.

- **Convergence:** After the initial steep descent, the loss for both training and validation begins to plateau, showing that the model is starting to converge to its optimal parameters.

- **Stable Loss:** As the epochs increase, the loss for both training and validation stabilizes, indicating that the model is not learning significantly from additional training. This stabilization is happening at a relatively low loss value, which is a good sign of model performance.

- **Closeness of Curves:** The training and validation loss curves are close to each other throughout the training process, suggesting that the model is generalizing well to unseen data. There is no significant divergence, which implies that overfitting is likely not a problem for this model.

- **Overfitting and Generalization:** When the validation loss starts increasing while training loss continues to decrease, it indicates overfitting. A good model should have both low training loss and low validation loss, showing that it has learned well and can generalize to new data. The figure shows no significant divergence between the training and validation loss, which is a common sign of overfitting. Overfitting would be indicated by a decrease in training loss but an increase in validation loss, which is not observed here.

This table provides a description of the model's learning behavior for each specified range of epochs.

Epoch Range	Observations
0-10	Rapid decrease in loss indicating significant learning from the data. The model quickly adjusts weights from their initial random state, capturing large gains in performance.
10-30	Slower rate of decrease in loss. The model begins to fine-tune its parameters, with the initial and easiest learning improvements already realized.
30-50	Loss stabilizes and shows minimal decrease. This suggests the model has largely converged, with additional training providing diminishing returns. The model's performance is optimized and stable.

Table 8.3: Behavior of the model across different epoch ranges

8.4 Model Performance and Comparative Evaluation

This study developed a Deep Neural Network (DNN) model that proved to be the most advanced machine learning tool for predicting Peak Particle Velocity (PPV). To benchmark its predictive improvement, a baseline was established using the conventional industry-standard model, with linear regression serving as the primary point of comparison. The Simple Neural Network, representing the initial model in the experimental process, surpassed the capabilities of the linear model, while linear regression provided a basic, yet interpretable, reference point.

The DNN model's performance was compared against these benchmarks to validate its effectiveness and contextualize its superior predictive ability. The two-stage model, which blends RandomForestRegressor and LinearRegression, provided an intermediate complexity level for comparison. The most critical aspect of this comparative analysis was to demonstrate the DNN model's advanced capabilities over the industry-standard model, as well as the other machine learning prototypes developed during this research.

Evaluation Metric	Simple Neural Network	Two-Stage Model	DNN Model	Industry Standard Model
Mean Absolute Error (MAE)	2.049	1.26	**0.60**	2.04
Mean Squared Error (MSE)	7.72	2.85	**0.94**	7.41
R-squared	0.49	0.79	**0.94**	0.51

Table 8.4: Evaluation results of the Machine Learning models compared to the industry standard model.

The evaluation metrics, as summarized in Table 8.4, illustrate the DNN model's superior accuracy and reliability in PPV prediction. With the lowest Mean Absolute Error (MAE), the lowest Mean Squared Error (MSE), and the highest R-squared value, the DNN model outperformed not just the industry-standard model but also the other developed prototypes. This clearly indicates the potential of DNNs to revolutionize predictive accuracy in mining operations, significantly contributing to safer and more efficient blasting practices.

8.4.1 Analysis of Benchmark Models

The Deep Neural Network (DNN) model's performance is compared against a simpler neural network model and a linear regression and random forest model to identify areas for improvement. The results are evaluated in table 8.4. The reason for comparing the DNN model with two different models are:

- **Point of Reference:** Benchmark models provide a point of reference to evaluate the DNN model. By comparing against well-understood models, the DNN's strengths and areas for improvement become evident.

- **Model Validation:** The performance of the DNN model, when contrasted with benchmark models, can validate the complexity and sophistication of the DNN approach.

9 Discussion

This chapter critically examines the process and implications of developing a Deep Neural Network (DNN) model for predicting blast-induced ground vibrations in collaboration with an external Mining Consultant Company. It highlights the inherent limitations encountered in this collaborative effort and the strategies deployed to address these challenges.

This chapter interprets the analysis of the model's findings and explores potential enhancements to the model, by examining the application of Machine Learning (ML) within the mining sector. The chapter aims to contextualize the significance of the research findings within the scope of mining, by underscoring the practical utility and theoretical contributions of the study. This discussion seeks to bridge the gap between academic research and industry practice, offering insights into the future trajectory of ML integration in mining operations.

9.1 Evaluating Machine Learning Model Complexity

In Chapter 8, Figure 8.5 demonstrates the feature importance as presented by the Deep Neural Network (DNN) model. This visualization is crucial as it reveals a better understanding of the factors influencing PPV. The DNN model identifies 'Scaled Distance' as the most significant feature, aligning with industry standards, but it also highlights the substantial roles of 'MIC' (Maximum Instantaneous Charge) and 'Distance'. Interestingly, the model assigns considerable weight to multiple 'Site' variables, suggesting that environmental or location-specific factors have a noticable impact on PPV predictions. This is representative of the DNN's capability to capture complex, multi-dimensional relationships in the data, which traditional models might overlook.

Meanwhile, the industry standard model, which is based on simpler regression approaches, shows a heavy reliance on 'Scaled Distance' and 'MIC'. While these are undoubtedly key factors, the industry model's focus is narrower, primarily concentrating on these variables. This limited focus can be a significant drawback.

As Figure 8.5 demonstrates, other features, though less pronounced, contribute to the predictive accuracy and overall model robustness. The industry model's tendency to overlook these could lead to a less comprehensive understanding of PPV dynamics, potentially resulting in predictions that do not fully encapsulate all influencing factors.

The industry's emphasis on 'Scaled Distance' is understandable given its historical significance in regression models for PPV prediction. However, this approach, while simplified, might not capture the full complexity of the influencing factors. The DNN model's ability to integrate a broader range of features illustrates the potential for a more holistic approach to PPV prediction.

By incorporating the importance of more of the features from the dataset, the DNN model is not only aligning with traditional knowledge but also extending it, providing a more detailed and potentially more accurate picture of the factors influencing PPV. This approach is particularly valuable in complex geophysical contexts where multiple variables interact in non-linear ways to affect PPV outcomes

9.2 Requirements Approach

The thesis has been developed based on these requirements in agreement with the Mining Consultant Company:

- **Data Quality and Quantity:** Access to high-quality dataset that covers relevant blasting parameters.
- **Model Performance:** The model should provide greater predictive capabilities for Peak Particle Velocity (PPV) when compared with the current industry standard model (USBM equation).
- **Security and Privacy:** It should ensure confidentiality and integrity of shared data.
- **Evaluation and Reporting:** Clear metrics for evaluating the model's performance.

The Mining Consultant Company provided a real world dataset of historical features of multiple blasting operations, large enough to train a Machine Learning model. The development of the project was conducted entirely on a local machine, without being designed as an application or deployed externally. This approach greatly simplified data security and privacy concerns.

The security and privacy of the data was achieved by password security and limiting access. The data has been consistently backed up, safeguarding against potential data loss. Additionally, GitHub has been utilized to track changes and modifications made throughout the project. To maintain the integrity of the results, data validation has been a continuous process. This has involved regularly checking the data for accuracy and consistency, ensuring reliable outcomes from the modeling process.

9.3 Addressing Limitations and Findings

Understanding and acknowledging the limitations of a study is crucial for a grounded interpretation of its findings. As stated in Chapter 1 and further presented in Section 1.7, this thesis recognizes a set of limitations, established in the early stages in collaboration with the external Mining Consultant Company. These limitations have been integral to shaping the research approach and interpreting the outcomes.

One of the primary limitations involves the dataset used in this study. While it is comprehensive and well-suited for the objectives of this research, it inherently limits the model's broader applicability. The dataset, being a reflection of specific operational conditions and environmental characteristics of a particular mining site, might not be fully representative or applicable to other mining scenarios. This particularity raises concerns about the model's generalizability. The effectiveness of a Deep Neural Network (DNN) in predicting Peak Particle Velocity (PPV) heavily relies on the diversity and representativeness of its training data. A model trained on data from a specific geological or operational context may not perform as well when applied to a different setting, potentially limiting its practical utility beyond the context similar to the training dataset.

Regarding technological aspects, the constraints were found to be minimal. The computational resources at disposal were sufficient for the planned complexity of the DNN architecture and the depth of training it underwent. Although the model's performance and capabilities are inherently linked to the hardware and software resources available, this research did not encounter significant limitations in these areas. The hardware and software used were sufficiently capable, facilitating the development of a robust and effective model.

In light of these limitations and the technological context, the findings of this research should be interpreted with an understanding of their context-specific nature. The model's applicability and predictive accuracy are closely tied to the characteristics of the dataset it was trained on and the technological environment in which it was developed. Future research could focus on broadening the dataset to include a more varied range of mining conditions, thereby enhancing the model's applicability and generalizability. Further exploration into the scalability of the model and the potential use of advanced computational resources, such as cloud computing, could also be valuable avenues to extend the capabilities and reach of the predictive model in mining operations.

9.3.1 Challenges of Machine Learning Models in Mining

Utilizing Machine Learning models in the mining sector can offer promising advancements but also posing significant challenges, especially when it comes to safety and operational decisions.

Machine Learning models, including Deep Learning Networks can perform remarkably well but understanding how they reach their conclusions can be confusing. In mining, where decisions can impact worker safety and environmental preservation, being able to trust and understand model predictions is essential. If a model says a particular blast will not breach safety thresholds, it is vital to understand why and to be confident in that assertion.

The core of any Machine Learning model lies in lots of high-quality data. Mining operations generate vast amounts of data, but it can be inconsistent. If a model is trained on poor-quality data, its predictions could be off, potentially leading to unsafe conditions. The range of data, from numbers to categories and even sequences over time, adds complexity to this task.

Another issue is that a model trained in one mining environment might not work well in another. Each mine is as unique, with its own geological and operational characteristics. A model that shows great evaluation results in predicting vibrations for one mine might be completely inaccurate for another mine. This raises questions about the scalability and transferability of Machine Learning models across different mining settings.

It exists a gap between theory and practice. Models might show excellent performance in simulations or controlled test environments but fail to replicate those results in the real-world chaos of a mining operation, where unexpected variables and complex interactions are the norms.

Another limiting factor is the technological infrastructure in mining. Machine Learning models often demand substantial computational power, data storage, and processing capabilities, which might not be available in remote or resource-limited mining sites. This can restrict the use of sophisticated models that require significant computational resources to operate effectively.

Recognizing and addressing these limitations is crucial in ensuring that the benefits of machine learning can be fully realized without compromising the safety and integrity of mining operations.

9.4 Deep Neural Network Predictive Efficacy

The Deep Neural Network (DNN) model developed as part of this research has demonstrated outstanding performance in the prediction of blast-induced vibrations, achieving an R-squared (R^2) value of 0.94. This is a significant improvement compared to the current industry standard, which has an R^2 of 0.51. Such an increase in the R^2 value indicates that the DNN model can explain 94% of the variability in the blast vibration data, offering a highly accurate model for predicting the effects of blasting operations.

A high R^2 may not always mean a model is the best or most appropriate. It does not account for overfitting, nor does it indicate if the model assumptions are met. Therefore, R^2 should be considered alongside other performance metrics and model validation techniques.

The Deep Neural Network (DNN) model's ability to consider multiple variables simultaneously is a key factor in its superior performance. Unlike traditional models that may rely heavily on a few features such as 'Scaled Distance' and 'MIC', the DNN model integrates various features, creating a more comprehensive understanding of the blast dynamics. This multifaceted approach allows for the consideration of complex interactions between different variables, which is crucial in capturing the intricacies of blast-induced vibrations.

The evaluation metrics further demonstrate the model's enhanced predictive capabilities. With a Mean Absolute Error (MAE) of 0.60, the DNN model shows a closer average prediction to the actual vibration values than the industry standard model, which has a MAE of 2.04. This lower MAE indicates that the DNN model's predictions are, on average, much closer to the true data points.

Similarly, the Mean Squared Error (MSE) of 0.94 for the DNN model, compared to 7.41 for the industry standard, suggests that the DNN model's predictions deviate less from the true values, especially when considering the square of those deviations, which emphasizes larger errors.

These metrics not only exhibit the model's accuracy but also its reliability and robustness in various situations. By accurately predicting the outcomes of blasting, the DNN model can significantly contribute to environmental safety, ensuring that vibration levels remain within safe limits to protect nearby structures and habitats. Additionally, this level of predictive accuracy can enhance operational efficiency by aiding in the planning and execution of blasting operations, ultimately leading to a reduction in unexpected downtime and costs associated with over- or under-blasting.

9.5 Theoretical And Practical Implications

This study shows important implications for both the understanding and application of Machine Learning in environmental care and risk management within the mining industry. From a theoretical perspective, using a Deep Neural Network (DNN) to predict Peak Particle Velocity (PPV) shows that advanced Machine Learning methods can make sense of complex data. This supports the idea that using Machine Learning to analyze environmental data can lead to better predictions and a clearer picture of the patterns and forces at play.

On a practical level, the successful creation of a Machine Learning model has the potential to change the mining sector's approach to environmental risk management. By accurately predicting PPV, mining companies can proactively implement control measures to limit the impact of blasting effects on nearby communities and environments. This contributes to the broader goals of sustainable mining practices and corporate social responsibility. Furthermore, the model's capacity to consider a range of variables beyond the traditionally used 'Scaled Distance' indicates a more nuanced understanding of blast impacts, enabling a more comprehensive risk assessment.

The findings emphasizes the practicality of local development and application of ML models. Given the often resource-constrained environment of on-site mining operations, the ability to develop, train, and refine predictive models locally without the need for extensive computational infrastructure is a significant advantage. This democratization of Machine Learning can empower smaller mining entities to leverage cutting-edge technology for environmental monitoring.

In conclusion, the theoretical implications of this study reaffirm the value of Machine Learning in environmental science, while the practical implications point to Machine Learning's transformative potential in enhancing environmental risk management in mining. Future work could focus on integrating such models into real-time monitoring systems, providing immediate feedback and action points for mining operations to address environmental concerns promptly.

10 Conclusion And Future Work

This study focused on the application of Machine Learning techniques, particularly the implementation of a Deep Neural Network (DNN) model, for the prediction of blast-induced ground vibrations utilizing a provided data source. The performance of the DNN model was evaluated and benchmarked against the prevailing industry-standard model used for similar vibration predictions. The findings of this research were noteworthy, showcasing a marked enhancement in predictive accuracy. By achieving this, the study successfully met its primary objective, which was to simply surpass the predictive capabilities of the current industry benchmark. This advancement not only highlights the effectiveness of DNN models in this specific domain but also provides significant confidence for future research and practical applications in the field of blast-induced ground vibration prediction.

10.1 Achievements

The primary objective of this thesis was to develop a Machine Learning (ML) model that surpasses the predictive accuracy of conventional industry standard methods for predicting blast-induced ground vibrations. This objective was successfully met through the implementation of a Deep Neural Network (DNN) model, which demonstrated a substantial enhancement in performance metrics. Specifically, the DNN model achieved a remarkable 84% increase in the R-squared value, coupled with a significant reduction in error margins – a 70% decrease in Mean Absolute Error (MAE) and an 87% decrease in Mean Squared Error (MSE) – compared to the traditional industry model. These improvements in key evaluation metrics not only underscore the effectiveness of the DNN model but also reinforce the potential of employing DNN models for similar-scale datasets in future projects, showcasing their practical viability in the field.

10.2 Impact On Mining Operations

Enhancing the accuracy of predicting blast-induced ground vibrations through the use of ML models has a substantial impact on mining operations. It equips blast designers, management, and personnel with an improved tool to make more informative decisions regarding environmental, safety, and regulatory compliance risks associated with blasting activities. The adoption of a Deep Neural Network (DNN) model, as a replacement for the existing industry standard models, would mark a considerable advancement in prediction accuracy. This improvement would contribute to a significant reduction in the inherent risks associated with blasting.

10.3 Future Work

The outcomes of this research offer a promising foundation for further work in the field of blast-induced PPV prediction using Machine Learning (ML) algorithms. Future studies could focus on datasets characterized by 'higher risk' scenarios. These scenarios typically involve lower scale distance values coupled with higher potential PPV levels. Notably, this study identified that the greatest variability in predictive accuracy occurred in these 'high-risk' scenarios. Therefore, targeting improvements in predictive models for such scenarios could be extremely valuable for mitigating the risks associated with blast vibration exceedance.

Another key insight from this research is the value of incorporating a broader range of variables into predictive models. Future works should seek to expand the dataset by including even more variables. While challenges in data collection and record-keeping have been acknowledged by the data source provider, it would be extremely valuable to the mining industry to enhance their data collection and record-keeping systems. Prioritizing the accumulation of detailed, multi-faceted data will significantly contribute to the refinement and accuracy of ML-based PPV prediction models.

Furthermore, there is considerable scope for exploring a diverse array of ML algorithms or even hybrid approaches that combine multiple algorithms. The rapidly evolving landscape of ML offers continuously emerging techniques and methodologies. Future research should leverage these advancements to not only keep pace with the technological progress but also to explore innovative ways in which these evolving tools can be applied to blast-induced PPV predictions. The ongoing development and integration of advanced ML algorithms hold the potential to substantially elevate the precision and reliability of PPV predictions.

10.4 Final Thoughts

This research highlights the enhanced capability of Machine Learning (ML) models in accurately predicting blast-induced ground vibrations. To facilitate wider adoption in the mining industry's daily operations, it's essential that these ML models are user-friendly for engineers and blast designers, particularly those who may not have a background in programming. Future developments could focus on making these models more accessible in an operational setting. This could be achieved by developing a straightforward web application, featuring a non-intimidating user interface, tailored for operational staff who seek reliable predictive results. Such an approach would promote broader industry adoption, streamlining the process for daily users who rely on these predictions for their operational decisions.

11

Deep Neural Network Model Architecture

Figure 11.1 represents a complex Deep Neural network (DNN) model architecture used for supervised learning tasks. Starting with an input layer that matches the feature count of the data, it progresses through several densely connected layers where computations occur, interspersed with non-linear activation functions like ReLU to capture complex patterns.

Batch normalization layers aid in stabilizing the training by normalizing neuron inputs, while dropout layers mitigate overfitting by randomly disabling neurons. The final dense layer outputs predictions, with an Adam optimizer fine-tuning the weights via backpropagation. The network's design, including layer depth and neuron count, is typically tailored through trials to best address the specific dataset and learning task at hand.

The visualization is provided with the use of TensorBoard.

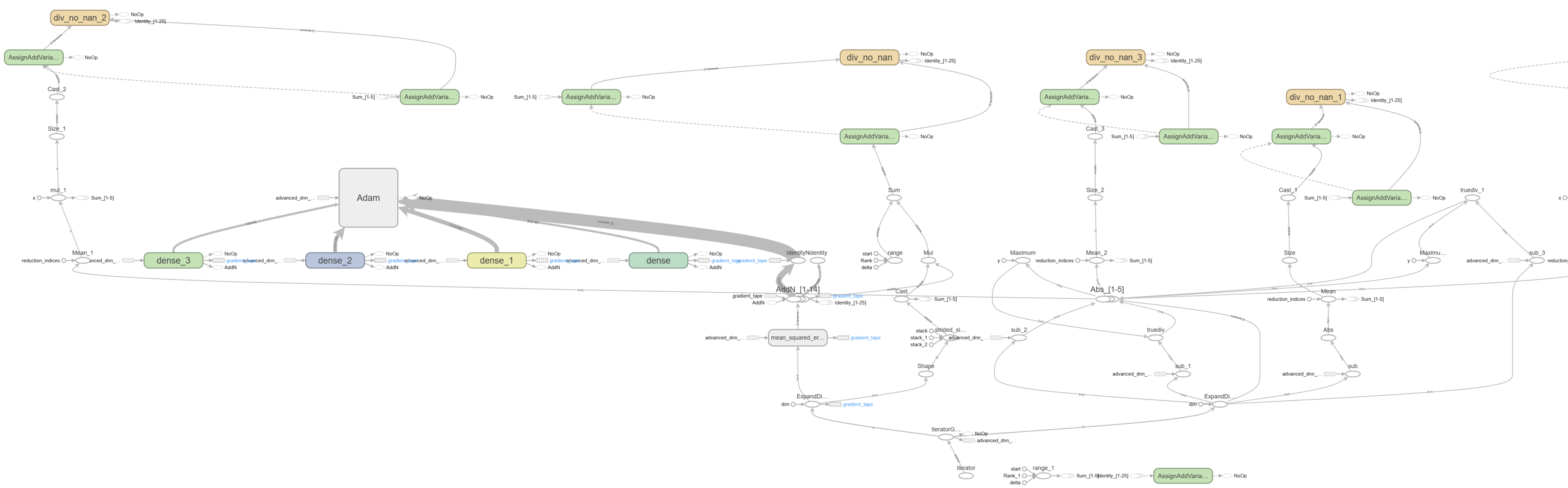

Figure 11.1: Graphical representation of the Deep Neural Network model developed in this study.

Bibliography

[1] Martin Abadi et al. “Tensorflow: Large-scale machine learning on heterogeneous distributed systems.” In: *arXiv preprint arXiv:1603.04467* (2016).

[2] Abien Fred Agarap. “Deep learning using rectified linear units (relu).” In: *arXiv preprint arXiv:1803.08375* (2018).

[3] Hemant Agrawal and AK Mishra. “Modified scaled distance regression analysis approach for prediction of blast-induced ground vibration in multi-hole blasting.” In: *Journal of Rock Mechanics and Geotechnical Engineering* (2019), pp. 202–207.

[4] Javier M Aguiar-Pérez et al. “Understanding Machine Learning Concepts.” In: *Encyclopedia of Data Science and Machine Learning*. IGI Global, 2023, pp. 1007–1022.

[5] Maryam Amiri et al. “A new combination of artificial neural network and K-nearest neighbors models to predict blast-induced ground vibration and air-overpressure.” In: *Engineering with Computers* (2016), pp. 631–644.

[6] Danial Jahed Armaghani et al. “Feasibility of ICA in approximating ground vibration resulting from mine blasting.” In: *Neural Computing and Applications* (2018), pp. 457–465.

[7] TS Bajpayee et al. “Blasting injuries in surface mining with emphasis on flyrock and blast area security.” In: *Journal of Safety Research* (2004), pp. 47–57.

[8] Ezzeddin Bakhtavar et al. “Air pollution risk assessment using a hybrid fuzzy intelligent probability-based approach: mine blasting dust impacts.” In: *Natural Resources Research* (2021), pp. 2607–2627.

[9] Ezzeddin Bakhtavar et al. “Green blasting policy: simultaneous forecast of vertical and horizontal distribution of dust emissions using artificial causality-weighted neural network.” In: *Journal of Cleaner Production* (2021).

[10] Mohammad Mahdi Bejani and Mehdi Ghatee. “A systematic review on overfitting control in shallow and deep neural networks.” In: *Artificial Intelligence Review* (2021), pp. 1–48.

[11] Mikael Berndtsson et al. *Thesis projects: a guide for students in computer science and information systems*. Springer Science & Business Media, 2007.

[12] Bradley Camburn et al. "Design prototyping methods: state of the art in strategies, techniques, and guidelines." In: *Design Science* (2017), e13.

[13] SR Dindarloo. "Peak particle velocity prediction using support vector machines: a surface blasting case study." In: *Journal of the Southern African Institute of Mining and Metallurgy* 115.7 (2015), pp. 637–643.

[14] Aline Dresch et al. *Design science research*. Springer, 2015.

[15] Sasidhar Duggineni. "Data Integrity Controls: The Universal basis for Authenticity and Reliability of Data." In: *INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY* 1 (2023), pp. 53–58.

[16] Faisal C Emetumah, Patrice AC Okoye, and Chinedu O Okoye. "Sustainable Approach in the Assessment of Safety and Environmental Regulatory Compliance During Mining Activities." In: *Journal of Human Ecology* (2020), pp. 104–115.

[17] Wolfgang Ertel. *Introduction to artificial intelligence*. Springer, 2018.

[18] Roohollah Shirani Faradonbeh et al. "Prediction and minimization of blast-induced flyrock using gene expression programming and firefly algorithm." In: *Neural Computing and Applications* (2018), pp. 269–281.

[19] Yewuhalashet Fissha et al. "Application of Bayesian Neural Network (BNN) for the Prediction of Blast-Induced Ground Vibration." In: *Applied Sciences* 13 (2023), p. 3128.

[20] Ebrahim Ghasemi et al. "Application of artificial intelligence techniques for predicting the flyrock distance caused by blasting operation." In: *Arabian Journal of Geosciences* (2014), pp. 193–202.

[21] Jiang Guo, Peidong Zhao, and Pingfeng Li. "Prediction and Optimization of Blasting-Induced Ground Vibration in Open-Pit Mines Using Intelligent Algorithms." In: *Applied Sciences* (2023), p. 7166.

[22] Anne Håkansson. "Portal of research methods and methodologies for research projects and degree projects." In: *The 2013 World Congress in Computer Science, Computer Engineering, and Applied Computing WORLDCOMP 2013; Las Vegas, Nevada, USA, 22-25 July*. CSREA Press USA. 2013, pp. 67–73.

[23] Anne Håkansson and Ronald Lee Hartung. "Artificial intelligence: concepts, areas, techniques and applications." In: *Lund: Studentlitteratur* (2020), p. 371.

[24] M Hasanipanah et al. "Prediction of an environmental issue of mine blasting: an imperialistic competitive algorithm-based fuzzy system." In: *International Journal of Environmental Science and Technology* (2018), pp. 551–560.

[25] Mahdi Hasanipanah et al. "Feasibility of indirect determination of blast induced ground vibration based on support vector machine." In: *Measurement* (2015), pp. 289–297.

[26] Peter Hatherly et al. “Drill monitoring results reveal geological conditions in blasthole drilling.” In: *International Journal of Rock Mechanics and Mining Sciences* (2015), pp. 144–154.

[27] Shahab Hosseini, Amin Mousavi, and Masoud Monjezi. “Prediction of blast-induced dust emissions in surface mines using integration of dimensional analysis and multivariate regression analysis.” In: *Arabian Journal of Geosciences* (2022), pp. 1–14.

[28] Shahab Hosseini et al. “Prediction of dust emission due to open pit mine blasting using a hybrid artificial neural network.” In: *Natural Resources Research* 6 (2021), pp. 4773–4788.

[29] Shahab Hosseini et al. “Prediction of ground vibration due to mine blasting in a surface lead–zinc mine using machine learning ensemble techniques.” In: *Scientific Reports* (2023), p. 6591.

[30] Lei Huang et al. “Normalization techniques in training dnns: Methodology, analysis and application.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[31] Muhammad Ahmad Iqbal. “Application of regression techniques with their advantages and disadvantages.” In: (2021), pp. 11–17.

[32] Valentin Isheyskiy, Mikhail Marinin, and Vadim Dolzhikov. “Combination of fracturing areas after blasting column charges during destruction of rocks.” In: *International Journal of Engineering Research and Technology* (2019), pp. 2953–2956.

[33] Danial Jahed Armaghani et al. “Application of two intelligent systems in predicting environmental impacts of quarry blasting.” In: *Arabian Journal of Geosciences* (2015), pp. 9647–9665.

[34] Feng Jia et al. “Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data.” In: *Mechanical systems and signal processing* (2016), pp. 303–315.

[35] Wenchao Jiang et al. “Simulating the peak particle velocity in rock blasting projects using a neuro-fuzzy inference system.” In: *Engineering with Computers* (2019), pp. 1203–1211.

[36] Michael I Jordan and Tom M Mitchell. “Machine learning: Trends, perspectives, and prospects.” In: *Science* (2015), pp. 255–260.

[37] Ferdin Joe John Joseph, Sarayut Nonsiri, and Annop Monsakul. “Keras and TensorFlow: A hands-on experience.” In: *Advanced deep learning for engineers and scientists: A practical approach* (2021), pp. 85–111.

[38] Vladislav Kecojevic and Mark Radomsky. “Flyrock phenomena and area security in blasting-related accidents.” In: 9 (2005), pp. 739–750.

[39] Manoj Khandelwal and TN Singh. “Prediction of blast induced ground vibrations and frequency in opencast mine: a neural network approach.” In: *Journal of sound and vibration* (2006), pp. 711–725.

[40] Manoj Khandelwal and TN Singh. "Prediction of blast-induced ground vibration using artificial neural network." In: *International Journal of Rock Mechanics and Mining Sciences* 46.7 (2009), pp. 1214–1222.

[41] Mohammadreza Koopialipoor et al. "Three hybrid intelligent models in estimating flyrock distance resulting from blasting." In: *Engineering with Computers* (2019), pp. 243–256.

[42] Nikolaus Kriegeskorte and Tal Golan. "Neural network models and deep learning." In: 7 (2019), R231–R236.

[43] Ranjan Kumar, Deepankar Choudhury, and Kapilesh Bhargava. "Determination of blast-induced ground vibration equations for rocks using mechanical and geological properties." In: *Journal of Rock Mechanics and Geotechnical Engineering* (2016), pp. 341–349.

[44] Machine Learning. "Tom mitchell." In: *Publisher: McGraw Hill* (1997).

[45] Karlo Leskovar et al. "Influence of Meteorological Parameters on Explosive Charge and Stemming Length Predictions in Clay Soil during Blasting Using Artificial Neural Networks." In: *Applied Sciences* (2021), p. 7317.

[46] Frank J Lucca. "Tight construction blasting: ground vibration basics, monitoring, and prediction." In: *Terra Dinamica LLC* (2003), pp. 1–21.

[47] Omar Abdulwahabe Mohamad. "New Virtual Environment Based on Python Programming." In: *Iraqi Journal For Computer Science and Mathematics* 3.2 (2022), pp. 111–118.

[48] M Monjezi et al. "Predicting blast-induced ground vibration using various types of neural networks." In: *Soil Dynamics and Earthquake Engineering* (2010), pp. 1233–1236.

[49] Hoang Nguyen and Xuan-Nam Bui. "Soft computing models for predicting blast-induced air over-pressure: A novel artificial intelligence approach." In: *Applied Soft Computing* (2020), p. 106292.

[50] Hoang Nguyen, Xuan-Nam Bui, and Erkan Topal. "Enhancing predictions of blast-induced ground vibration in open-pit mines: Comparing swarm-based optimization algorithms to optimize self-organizing neural networks." In: *International Journal of Coal Geology* (2023), p. 104294.

[51] Hoang Nguyen, Xuan-Nam Bui, and Erkan Topal. "Reliability and availability artificial intelligence models for predicting blast-induced ground vibration intensity in open-pit mines to ensure the safety of the surroundings." In: *Reliability Engineering & System Safety* (2023), p. 109032.

[52] Hoang Nguyen et al. "Evaluating and predicting blast-induced ground vibration in open-cast mine using ANN: a case study in Vietnam." In: *SN Applied Sciences* (2019), pp. 1–11.

[53] Hoang Nguyen et al. "Predicting blast-induced ground vibration in open-pit mines using vibration sensors and support vector regression-based optimization algorithms." In: *Sensors* (2019), p. 132.

[54] Hoang Nguyen et al. “Prediction of blast-induced ground vibration in an open-pit mine by a novel hybrid model based on clustering and artificial neural network.” In: *Natural Resources Research* (2020), pp. 691–709.

[55] Harry R Nicholls, Charles F Johnson, and Wilbur I Duvall. “Blasting vibrations and their effects on structures.” In: (1971).

[56] Umit Ozer et al. “Bench blasting design based on site-specific attenuation formula in a quarry.” In: *Arabian Journal of Geosciences* (2013), pp. 711–721.

[57] Hossein Samareh et al. “Optimization of a nonlinear model for predicting the ground vibration using the combinational particle swarm optimization-genetic algorithm.” In: *Journal of African Earth Sciences* (2017), pp. 36–45.

[58] Felipe Sánchez and Philipp Hartlieb. “Innovation in the mining industry: Technological trends and a case study of the challenges of disruptive innovation.” In: *Mining, Metallurgy & Exploration* 37 (2020), pp. 1385–1399.

[59] DP Singh and VR Sastry. “Rock fragmentation by blasting influence of joint filling material.” In: *Journal of Explosive Engineering* (1986), pp. 18–27.

[60] Hwanjun Song et al. “Learning from noisy labels with deep neural networks: A survey.” In: *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[61] Haoran Sun et al. “Learning to optimize: Training deep neural networks for interference management.” In: *IEEE Transactions on Signal Processing* (2018), pp. 5438–5453.

[62] Amey Thakur and Archit Konde. “Fundamentals of neural networks.” In: *International Journal for Research in Applied Science and Engineering Technology* (2021), pp. 407–26.

[63] Yingjie Tian and Yuqi Zhang. “A comprehensive survey on regularization strategies in machine learning.” In: *Information Fusion* (2022), pp. 146–166.

[64] Amit Kumar Verma and Trilok Nath Singh. “Intelligent systems for ground vibration measurement: a comparative study.” In: *Engineering with Computers* (2011), pp. 225–233.

[65] Jinjiang Wang et al. “Deep learning for smart manufacturing: Methods and applications.” In: *Journal of manufacturing systems* (2018), pp. 144–156.

[66] Richard Y Wang, Veda C Storey, and Christopher P Firth. “A framework for analysis of data quality research.” In: *IEEE transactions on knowledge and data engineering* (1995), pp. 623–640.

[67] John F Wiss. “Control of vibration and blast noise from surface coal mining.” In: *Bureau of Mines, Open File Rept.* (1978).

[68] Kanit Wongsuphasawat et al. "Visualizing dataflow graphs of deep learning models in tensorflow." In: *IEEE transactions on visualization and computer graphics* (2017), pp. 1–12.

[69] Jian Zhou et al. "Development of a new methodology for estimating the amount of PPV in surface mines based on prediction and probabilistic models (GEP-MC)." In: *International Journal of Mining, Reclamation and Environment* (2021), pp. 48–68.

[70] Kelly H Zou, Kemal Tuncali, and Stuart G Silverman. "Correlation and simple linear regression." In: 3 (2003), pp. 617–628.