

UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA GEOLÓGICA, MINERA Y METALÚRGICA



TESIS

**“MACHINE LEARNING
APLICADO AL CONTROL DE LA FRAGMENTACIÓN
DE ROCAS EN LA VOLADURA DE MINAS A TAJO ABIERTO”**

**PARA OBTENER EL TÍTULO PROFESIONAL DE
INGENIERO DE MINAS**

**ELABORADO POR
YESSICA PAMELA IDROGO ZAMORA**

**ASESOR
MBA. ING. HENRY MAMFREDO ZAMBRANA FLORES**

LIMA-PERÚ

2022

AGRADECIMIENTO

Agradezco a mi madre, padre y familia por su constante apoyo y motivación en todas las etapas de mi vida, a los profesionales de DMH, Codelco por el apoyo brindándome la data necesaria para la realización del modelo de aprendizaje automático en el que se centra mi tesis, así como su guía en los temas asociados a voladura de rocas.

RESUMEN

La presente tesis desarrolla el análisis de datos y modelamiento predictivo en la voladura para una mina a tajo abierto.

La primera etapa consiste en el planteamiento del problema a resolver, esto incluye identificar los objetivos, y en el caso de un modelamiento de machine learning, definir cuál o cuáles son las variables objetivo que se van a predecir, esta variable a predecir será vital para resolver el problema en cuestión. En esta etapa se identifica cuál es el algoritmo o los algoritmos que se van a utilizar para resolver el problema en cuestión.

La segunda etapa consiste en el análisis de datos, donde se analiza los tipos de variables, la cantidad de datos disponibles, las categorías que se presentan, las correlaciones entre variables, y la distribución estadística de estas variables.

La tercera etapa es el tratamiento de datos, donde se realizan transformaciones sobre las variables de la base de datos. En la presente tesis se realizó los tratamientos de estandarización para homogeneizar el rango de cada parámetro del modelo y la transformación de Yeo Johnson para normalizar las distribuciones de datos que poseen un alto sesgo.

La cuarta etapa consiste en realizar el modelo de machine learning para conseguir la variable de interés, en este caso se usaron redes neuronales artificiales, que tuvieron

como variables de entrada los parámetros que ya fueron analizados y tratados en las etapas anteriores; y el output fue la variable de interés que permite resolver el problema planteado en la primera etapa.

La quinta etapa consistió en análisis de resultados donde se evaluó el impacto del programa basado en inteligencia artificial para cumplir el objetivo deseado de generar la ubicación de perforación de los holes que cumpla con el target de fragmentación; y, esto mediante una comparativa de los resultados obtenido mediante el método tradicional versus el método usando redes neuronales artificiales.

Mencionar que, la flexibilidad de la red neuronal artificial permite a los ingenieros generar modelos que se adecuen a sus necesidades de diseño, costo y objetivo de la fragmentación como llevar a un chancado y molienda.

ABSTRACT

This thesis develops the data analysis and predictive modeling in blasting for an open pit mine.

The first stage consists of posing the problem to be solved, this includes identifying the objectives, and in the case of a machine learning modeling which is the target variable or variables to be predicted, this variable to be predicted will be vital to solve the problem in question. At this stage, the algorithm or algorithms that are going to be used to solve the problem in question are identified.

The second stage consists of data analysis where the types of variables, the amount of data available, the categories that are presented, the correlations between variables, and the statistical distribution of these variables are analyzed.

The third stage is the data treatment where transformations are carried out on the variables of the database in the present thesis, the standardization treatments were carried out to homogenize the range of each parameter of the model and the Yeo Johnson transformation to normalize the distributions of data that are highly based.

The fourth stage consists of carrying out the machine learning model to obtain the variable of interest, in this case artificial neural networks were used, which had as input variables the parameters that were already analyzed and treated in the previous stages and the output was the variable of interest that allows solving the problem posed in the first stage.

The fifth stage consisted of analysis of results where the impact of the program based on artificial intelligence was evaluated to meet the desired objective of generating the drilling location of the holes that meets the fragmentation target and this through a comparison of the results obtained through the traditional method versus the method using artificial neural networks.

Mention that, the flexibility of the artificial neural network allows engineers to generate models that adapt to their design needs, cost and fragmentation objective such as leading to crushing and grinding.

INDICE

INTRODUCCIÓN.....	1
CAPITULO I.....	2
GENERALIDADES	2
1.1 PLANTEAMIENTO DEL PROBLEMA	4
1.1.1 FORMULACIÓN DEL PROBLEMA.....	5
1.2 OBJETIVOS.....	6
1.2.1 OBJETIVO GENERAL.....	6
1.2.2 OBJETIVOS ESPECÍFICOS	6
1.3 HIPÓTESIS.....	7
1.3.1 HIPÓTESIS GENERAL	7
1.3.2 HIPÓTESIS ESPECÍFICA	7
1.4 VARIABLES.....	8
1.4.1 VARIABLE DEPENDIENTE (Y).....	8
1.4.2 VARIABLE INDEPENDIENTE (X)	8
1.5 JUSTIFICACIÓN	8
1.6 METODOLOGÍA DE LA INVESTIGACIÓN	9
1.6.1 TIPO Y DISEÑO DE LA INVESTIGACIÓN.....	9
1.6.2 UNIDAD DE ANÁLISIS.....	10
1.6.3 ETAPAS DE LA INVESTIGACIÓN.....	22
1.6.4 MATRIZ DE CONSISTENCIA	23

1.7	TRABAJOS PREVIOS	24
1.8	ESTADO DEL ARTE	24
CAPITULO II.....		26
MARCO TEÓRICO Y CONCEPTUAL.....		26
2.1	MARCO TEÓRICO.....	26
2.1.1	<i>MACHINE LEARNING</i>	28
2.1.2	<i>APRENDIZAJE SUPERVISADO</i>	29
2.1.3	<i>APRENDIZAJE NO SUPERVISADO</i>	32
2.1.4	<i>VARIABLES USADAS</i>	34
2.1.5	<i>HIPÓTESIS DE ALGORITMOS NECESARIOS</i>	37
2.2	MARCO CONCEPTUAL	39
2.2.1	<i>DEFINICIÓN DE TÉRMINOS</i>	39
CAPITULO III.....		44
ANÁLISIS DE DATOS		44
3.1	RECOLECCIÓN DE DATOS	45
3.1.1	<i>DATOS</i>	45
3.1.2	<i>GBLAST</i>	46
3.1.3	<i>CARGA DE DATOS</i>	51
3.2	LIMPIEZA DE DATOS	53
3.2.1	<i>FACTOR DE CARGA</i>	53
3.2.2	<i>IDs</i>	57
3.2.3	<i>BURDEN Y ESPACIAMIENTO</i>	64
3.2.4	<i>DRILL Y REDRILLS</i>	64
3.2.5	<i>UCS</i>	65
3.3	ANÁLISIS DE DATOS.....	68
3.3.1	<i>ESTADÍSTICA DESCRIPTIVA</i>	68
3.3.2	<i>VISUALIZACIÓN DE DATOS</i>	71
CAPITULO IV.....		80
TRATAMIENTO DE DATOS		80
4.1	POWER TRANSFORMER POR MÉTODO BOX COX	81
4.2	ESTANDARIZACIÓN	84

4.3	TRAIN- TEST - SPLIT.....	87
4.4	MÉTRICAS.....	88
CAPITULO V.....		89
MODELADO CON REDES NEURONALES		89
5.1	REDES NEURONALES ARTIFICIALES	90
5.1.1	<i>PERCEPTRON</i>	90
5.1.2	<i>DESCENSO DE GRADIENTE ESTOCÁSTICO</i>	91
5.1.3	<i>ALGORITMO DE BACKPROPAGATION</i>	92
5.1.4	<i>FUNCIÓN DE PÉRDIDA</i>	93
5.2	PREDICCIÓN VERSUS DATA DE VALIDACIÓN	94
CAPITULO VI.....		97
IMPLEMENTACIÓN EN LOS PROCESOS DE VOLADURA		97
6.1	VARIABLES USADAS.....	98
6.2	CLUSTERIZACIÓN DEL TERRENO EN GRUPOS LITOLÓGICOS.....	98
6.3	COMPARATIVA RED NEURONAL ARTIFICIAL VERSUS MÉTODO TRADICIONAL DE VOLADURA. 101	
6.3.1	<i>MÉTODO TRADICIONAL</i>	101
6.3.2	<i>PROCESAMIENTO MÓDULO OPTIMIZACIÓN DE DISEÑO USANDO RED NEURONAL ARTIFICIAL</i>	102
CONCLUSIONES.....		107
RECOMENDACIONES.....		109
BIBLIOGRAFÍA		110
ANEXOS		

INDICE DE FIGURAS

FIGURA 1. 1. ESQUEMA DE LA INTELIGENCIA ARTIFICIAL.	3
FIGURA 1. 2. PLANO GEOLÓGICO DISTRITO CHUQUICAMATA.	12
FIGURA 1. 3. BRECHAS MAGMÁTICAS- HIDROTERALES DE INYECCIÓN. LA MATRIZ ESTÁ CONSTITUIDA POR LA ASOCIACIÓN CALCOSINA – ENARGITA – PIRITA (ALTA SULFIDIZACIÓN).	14
FIGURA 1. 4. EJEMPLOS DE BRECHAS MAGMÁTICAS – HIDROTERMALES UBICADAS EN LA PARTE ALTA DEL YACIMIENTO. A LA IZQUIERDA LA ASOCIACIÓN MINERALÓGICA CORRESPONDE A CALCOSINA – ENARGITA (BORNITA) – PIRITA. LA MATRIZ EN LA FOTO DE LA DERECHA ESTÁ COMPUESTA POR CALCOSINA.	15
FIGURA 1. 5. EDADES RADIOMÉTRICAS DE LOS PÓRFIDOS, ALTERACIÓN Y MINERALIZACIÓN DEL YACIMIENTO MMH (MODIFICADO DE TOBEY, 2005).	15
FIGURA 1. 6. PLANO ESTRUCTURAL DISTRITO CHUQUICAMATA.	18
FIGURA 1. 7. SECCIÓN ESQUEMÁTICA MMH CENTRAL.	19
FIGURA 1. 8. PLANO GEOLÓGICO ESTRUCTURAL MINA CASERONES.	19
FIGURA 1. 9. SECCIÓN LITOLÓGICA TIPO MMH.	20
FIGURA 1. 10. DOMINIOS DE DUREZA BOND EN MMH.	21
FIGURA 2. 1. DIAGRAMA DEL FUNCIONAMIENTO DE UN ALGORITMO DE MACHINE LEARNING EN UN APRENDIZAJE SUPERVISADO.	29
FIGURA 2. 2. CLASIFICACIÓN SUPERVISADA (RASCHKA, SEBASTIAN & MIRJALILI, VAHID (2017)).....	32
FIGURA 2. 3. EJEMPLO DEL USO DE UN ALGORITMO DE APRENDIZAJE NO SUPERVISADO, QUE HA CLUSTERIZADO LOS DATOS SIN NECESIDAD DE TENER SUS ETIQUETAS.	34
FIGURA 3. 1. MALLA A EVALUAR.	46
FIGURA 3. 2. PSEUDO UCS GENERADO POR EL GBLAST PARA LA MALLA Q1.	47
TABLA 3. 7. COLUMNA “HOLE” CON TRANSFORMACIÓN DE CÓDIGOS REPETIDOS.	62
FIGURA 5. 1. PROCESO ESQUEMATIZADO DE LAS REDES NEURONALES ARTIFICIALES.	90
FIGURA 5. 2. PREDICCIÓN DEL FACTOR DE CARGA PARA LOS HOLES DE LOS DATOS DE VALIDACIÓN (GRÁFICO DE VALIDACIÓN DEL MODELO PREDICTIVO).	95

FIGURA 5. 3. PREDICCIÓN DEL FACTOR DE CARGA PARA HOLES DE UNA NUEVA MALLA CON DATOS NO ENTRENADOS.	96
FIGURA 6. 1. IMAGEN QUE ESQUEMATIZA EL COMPORTAMIENTO UNITARIO POR CELDAS DEL FACTOR DE CARGA RECOMENDADO EN UN MAPA DE CALOR RESULTADO DE LA PREDICCIÓN CON REDES NEURONALES ARTIFICIALES.	99
FIGURA 6. 2. IMAGEN QUE ESQUEMATIZA LA CLUSTERIZACIÓN DEL FACTOR DE CARGA EN UN MAPA DE CALOR RESULTADO DE LA PREDICCIÓN CON LAS REDES NEURONALES ARTIFICIALES. SE OBSERVAN LOS BURDEN Y ESPACIAMIENTO QUE SE RECOMIENDA PARA CADA GRUPO LITOLÓGICO.	100

INDICE DE TABLAS

TABLA 1. 1. RECUPERACIÓN DE MINERAL DMH, CODELCO.	21
TABLA 1. 2. MATRIZ DE CONSISTENCIA.	23
TABLA 3. 1. LA TABLA MUESTRA EL DATASET DE LOS DATOS UTILIZADOS Y PODEMOS OBSERVAR EN CADA COLUMNA LOS PARÁMETROS UTILIZADOS.	45
TABLA 3. 2. DATASET FINAL GENERADO A PARTIR DE UNIR LOS CSV EXPORTADO DESDE EL PROGRAMA GBLAST.	53
TABLA 3. 3. DATASET CONCATENADO DEL FACTOR DE CARGA.	56
TABLA 3. 4. DATASET CONCATENADO DE LA CARGA DE EXPLOSIVO.	58
TABLA 3. 5. TRANSFORMACIÓN A ÍNDICES DE LA VARIABLE “HOLE”.	59
TABLA 3. 6. TRANSPUESTA DE FILAS POR COLUMNAS EN LA BASE DE DATOS.	60
TABLA 3. 8. BASE DE DATOS RESULTANTE CON COLUMNA “HOLE” TRANSFORMADA PARA CÓDIGOS REPETIDOS.	63
TABLA 3. 9. BASE DE DATOS QUE CONTIENE LA VARIABLE UCS (1152 FILAS).	67
TABLA 3. 10. TABLA DE ESTADÍSTICA DESCRIPTIVA DE LAS VARIABLES INVOLUCRADAS EN MODELO DE RED NEURONAL ARTIFICIAL.	71
TABLA 3. 11. HISTOGRAMAS DE LAS VARIABLES USADAS EN EL MODELO DE APRENDIZAJE AUTOMÁTICO.	72
TABLA 3. 12. BOXPLOT DE LAS VARIABLES USADAS EN EL MODELO DE APRENDIZAJE AUTOMÁTICO.	74
TABLA 3. 13. DIAGRAMA DE CORRELACIÓN DE LAS VARIABLES USADAS EN EL MODELO DE APRENDIZAJE AUTOMÁTICO.	76
TABLA 3. 14. DIAGRAMA DE DISPERSIÓN DE LAS VARIABLES USADAS EN EL MODELO DE APRENDIZAJE AUTOMÁTICO.	78
TABLA 4. 1. HISTOGRAMAS DE DISTRIBUCIÓN DE LOS PARÁMETROS DEL MODELO ANTES DE LA TRANSFORMACIÓN POR EL MÉTODO DE BOX COX.	81
TABLA 4. 2. BASE DE DATOS RESULTANTE DE LA TRANSFORMACIÓN DE LOS PARÁMETROS POR EL MÉTODO DE BOX COX.	83

TABLA 4. 3. HISTOGRAMA DE DISTRIBUCIÓN DE LOS PARÁMETROS DESPUÉS DE LA TRANSFORMACIÓN POR EL MÉTODO DE BOX COX.	84
--	----

INTRODUCCIÓN

En la actualidad técnicas como el machine learning o aprendizaje automático están permitiendo optimizar procesos en distintas industrias, así como la automatización de estos. La predicción de variables de interés utilizando algoritmos matemáticos expresados en un lenguaje de programación lo hacen posible mediante el uso de base de datos disponibles.

La presente tesis realiza un modelo con algoritmo de redes neuronales artificiales aplicados al control de la fragmentación en la voladura de rocas para minas a tajo abierto.

Se ha aprovechado la base de datos de voladuras realizadas para extraer datos de las principales variables involucradas en el proceso de voladura.

CAPITULO I.

GENERALIDADES

Actualmente el avance de la tecnología informática permite conservar y disponer de grandes volúmenes de datos; en la cantidad necesaria para realizar el uso de herramientas de inteligencia artificial como lo son el machine learning y el deep learning, también conocido como aprendizaje automático. En esta era de la tecnología moderna, hay un recurso que tenemos en abundancia: una gran cantidad de datos estructurados y no estructurados (Raschka, Sebastian & Mirjalili, Vahid (2017)).

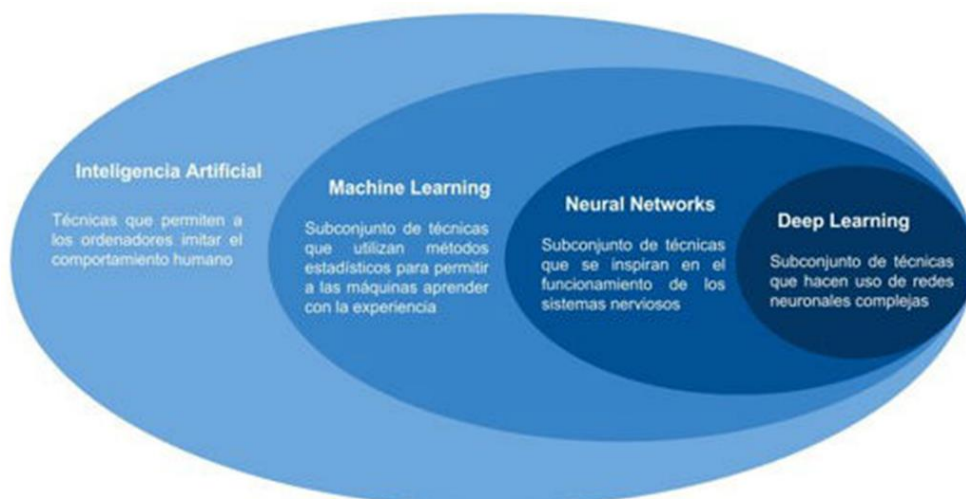


Figura 1. 1. Esquema de la inteligencia artificial.

Fuente: Tomado de <https://www.devacademy.es/>.

Actualmente distintas ramas de la ciencia ya vienen aplicando las herramientas de inteligencia artificial en busca de mejorar sus procesos y predecir el comportamiento de las variables que incidirán en el resultado final.

Las geociencias no son la excepción, en los últimos años se han realizado investigaciones sobre la aplicación del machine learning en minería y geología culminando en tesis, publicaciones y algunas de estas investigaciones vienen siendo implementadas por empresas del sector minero y energético.

Algunas de las aplicaciones del machine learning en geología son: mapeo geológico/geoquímico, detección automática de fracturas en testigos, prospección de targets económicos, predicción de facies. De igual manera en la actividad minera existen aplicaciones en ejecución las cuales como consecuencias de su gran mejora, favorecen el desarrollo de más iniciativas que permita automatizar y mejorar procesos; con el consiguiente rendimiento económico.

El presente trabajo muestra la aplicación del machine learning o aprendizaje automático, del análisis de datos de voladura con lo cual se desarrolla la red neuronal que permitirá generar el modelo predictivo para el diseño de la malla para la voladura de los bancos del tajo.

La complejidad de problemas en la industria minera peruana es un escenario favorable para el desarrollo e implementación de herramientas predictivas de procesos sensibles a los costos de operación. En medida de ello se hace necesario difundir técnicas como las expuestas en este trabajo.

1.1 PLANTEAMIENTO DEL PROBLEMA

La voladura es una de las principales actividades que se realizan en minería, para el caso de minería a tajo abierto no es la excepción y por ello influye sobre los costos operativos.

Para el efecto del presente estudio orientado a la voladura en tajo abierto puntualizaremos los siguientes conceptos de costo:

- Directos, son los asociados a la compra de insumos como los explosivos y a la adquisición de los servicios de perforación entre otros.
- Indirectos, son los asociados al tratamiento de la fragmentación de las rocas, el cual condiciona los procesos de molienda y chancado; por lo cual queda meridianamente claro lo trascendente de su correcta formulación y ejecución. El incremento de tiempos y de uso de energía, son el primer impacto en este costo.

Los costos de carga y transporte constituyen hasta el 50% de los costos operativos totales en las minas a tajo abierto. La fragmentación de la roca después de la voladura

es un determinante importante del costo asociado con estos dos componentes del desarrollo de la mina (K. Oraee, 2006).

Existen experiencias recurrentes sobre diseño de mallas de voladura que no cumplen con los objetivos propuestos. El control sistemático concurrente inadecuado de variables como: cantidad de explosivo, altura del banco, el UCS y target de fragmentación, será la explicación al incumplimiento indicado.

Las condiciones anisótropas de las zonas de voladura exigen el control multivariable. En el momento las aplicaciones informáticas utilizadas para este control no consiguen el objetivo total; provocando sobredimensionamiento de recursos.

Buscando proporcionar una solución que permita ejercer el control concurrente de las variables involucradas en el diseño de la malla de voladura, se propone un algoritmo mediante machine learning; soportado por el análisis de datos de voladura y el desarrollo de la consiguiente red neuronal artificial.

1.1.1 FORMULACIÓN DEL PROBLEMA

1.1.1.1 PROBLEMA GENERAL

¿Qué influencia tiene el sobredimensionamiento de recursos debido a que las aplicaciones actuales utilizadas para el control de la fragmentación en voladura no consiguen el objetivo total?

1.1.1.2 PROBLEMAS ESPECÍFICOS

¿Qué beneficios conlleva interpretar la correlación entre los parámetros de roca y variables de diseño para la fragmentación y diseño de una malla de voladura?

¿Cuál es el impacto al generar un ahorro de los recursos disponibles para el caso específico de diseño de mallas mediante un nuevo modelo predictivo?

¿Qué influencia tiene contar con un nuevo método de diseño de malla que se adecue completamente a las variaciones de las condiciones geológicas del terreno?

¿De qué manera es posible evitar una ubicación no optimizada de los holes de perforación en una malla de voladura?

1.2 OBJETIVOS

Se podrá considerar los siguientes:

1.2.1 OBJETIVO GENERAL

Diseñar una malla de voladura con control de la fragmentación de rocas en la voladura de minas a tajo abierto.

1.2.2 OBJETIVOS ESPECÍFICOS

- Interpretar la correlación entre los parámetros de roca como la resistencia mecánica de la roca y las variables de diseño, para determinar su influencia sobre la fragmentación y diseño de malla.
- Generar un ahorro en los metros perforados a través de un diseño de malla optimizada mediante el método de redes neuronales artificiales que se propondrá.
- Formular un modelo de aprendizaje automático que considere condiciones geológicas de terreno a mayor detalle que los modelos

actuales y que sea capaz de realizar recomendaciones para optimizar el dimensionamiento de la malla de voladura.

- Desarrollar un modelo de redes artificiales que prediga el Factor de Carga y proporcione la ubicación de los holes de perforación optimizados en base a la variabilidad del UCS.

1.3 HIPÓTESIS

1.3.1 HIPÓTESIS GENERAL

Un modelo de aprendizaje automático o machine learning basado en redes neuronales artificiales es capaz de ejercer control de la fragmentación de rocas en la voladura de minas a tajo abierto.

1.3.2 HIPÓTESIS ESPECÍFICA

- La fragmentación de las rocas depende principalmente de la resistencia mecánica de la roca y la carga de explosivo por su relación directa a la energía liberada que impacta sobre las rocas del terreno.
- Un modelo de aprendizaje automático genera un ahorro en los metros perforados por una mejor precisión.
- Un modelo de aprendizaje automático es capaz de recomendar las condiciones que se requieren para optimizar el dimensionamiento de la malla de voladura y garantiza la fragmentación requerida.

- La red neuronal artificial, en base a la variabilidad del UCS, predice el Factor de Carga, lo que implica el cambio espacial del diseño de malla y ello genera la ubicación óptima de los hole de perforación.

1.4 VARIABLES

1.4.1 VARIABLE DEPENDIENTE (Y)

Diseño de malla de voladura con control en la fragmentación en base a factor de carga predicho mediante redes neuronales.

1.4.2 VARIABLE INDEPENDIENTE (X)

Valores óptimos de factor de carga para una malla de voladura (UCS, Carga del explosivo, P50, P60, P80, P95).

1.5 JUSTIFICACIÓN

El desarrollo del presente trabajo permitirá ejecutar mallas de voladura, ajustando los costos de operación conforme a las condiciones geológicas del frente de voladura.

El análisis de datos y modelado predictivo son herramientas actuales muy potentes que nos permiten predecir el comportamiento de otras variables, en este caso buscamos predecir el mejor diseño en base al comportamiento mecánico de la roca y la carga de explosivo dado que esto permitirá controlar las variables de operación que impactan directamente en los costos de perforación y voladura, principales actividades en las minas a tajo abierto.

1.6 METODOLOGÍA DE LA INVESTIGACIÓN

1.6.1 TIPO Y DISEÑO DE LA INVESTIGACIÓN

1.6.1.1 ENFOQUE

El enfoque de la presente tesis se basa en un enfoque de carácter cuantitativo por ser un estudio en base a métricas, es decir en base a datos reales exactos obtenidos de una unidad minera.

El desarrollo de este trabajo utiliza la recolección de datos para probar las hipótesis, con base a la medición numérica y el análisis estadístico, para establecer patrones de comportamiento y probar las hipótesis establecidas. Lo ventajoso de esta investigación no solo se basa en la generalización de resultados, sino también en el control sobre las variables para lograr obtener una réplica, precisión y mejor predicción de estas, logrando así definir un diseño de malla de voladura óptimo.

1.6.1.2 ALCANCE

El alcance del estudio se define como correlacional debido a que se basa en identificar el grado de relación o asociación que existe entre las variables independientes (condiciones geológicas, carga de explosivo, etc) para predecir el valor de las variables dependientes establecidas en la presente tesis: diseño de una malla de voladura óptima en base a la fragmentación requerida.

Además de cuantificar, medir y establecer correlaciones entre variables, esta tesis explica la vinculación de estas.

1.6.1.3 DISEÑO DE LA INVESTIGACIÓN

Al tener un enfoque cuantitativo, la presente tesis contempla un diseño de investigación experimental, debido a, según teoría de diseños de investigación, se cuenta con un grupo de control que involucra las variables dependientes y/o no manipulables, y un grupo experimental, donde es posible la manipulación de las variables (variables independientes).

1.6.2 UNIDAD DE ANÁLISIS

El presente trabajo de tesis ha estudiado un proceso; por lo que, al ser un proceso, este puede ser utilizado para cualquier otra unidad minera. El desarrollo de la presente tesis ha obtenido base de datos de la unidad minera a tajo abierto División Ministro Hales, Codelco Chile.

A continuación, se describe las características, especificaciones y ubicación del sujeto de estudio, División Ministro Hales, Codelco Chile.

1.6.2.1 DIVISIÓN MINISTRO HALES

División Ministro Hales está ubicada a 2.600 metros de altura, a casi 10 kilómetros al norte de Calama por el camino que une a esta ciudad con Chuquicamata. En operaciones desde 2010, es una mina a rajo abierto. Produce calcina de cobre, concentrado de cobre y plata.

El yacimiento Mina Ministro Hales (MMH) corresponde a un pórfido cuprífero el cual presenta de manera particular, una sobre imposición de brechas hidrotermales tardías que contienen mineralización de cobre y plata la que es acompañada de altos contenidos de arsénico como impureza.

Marco geológico

El yacimiento MMH forma parte junto a Radomiro Tomic, Chuquicamata y Cluster Toki de un grupo de yacimientos de cobre de clase mundial alineados en una franja de orientación norte sur. Estos yacimientos del tipo pórfido cuprífero, están controlados por el Sistema Estructural de Domeyko o Sistema Estructural de la Falla Oeste. Esta estructura corresponde a una falla de rumbo (“strike slip fault”) de extensión regional, la cual es reconocida entre el Salar de Punta Negra y Quebrada Blanca, por una longitud cercana a los 500 km.

Las unidades de rocas identificadas en el depósito MMH, tanto al oeste como al este de la Falla Oeste se describen brevemente a continuación:

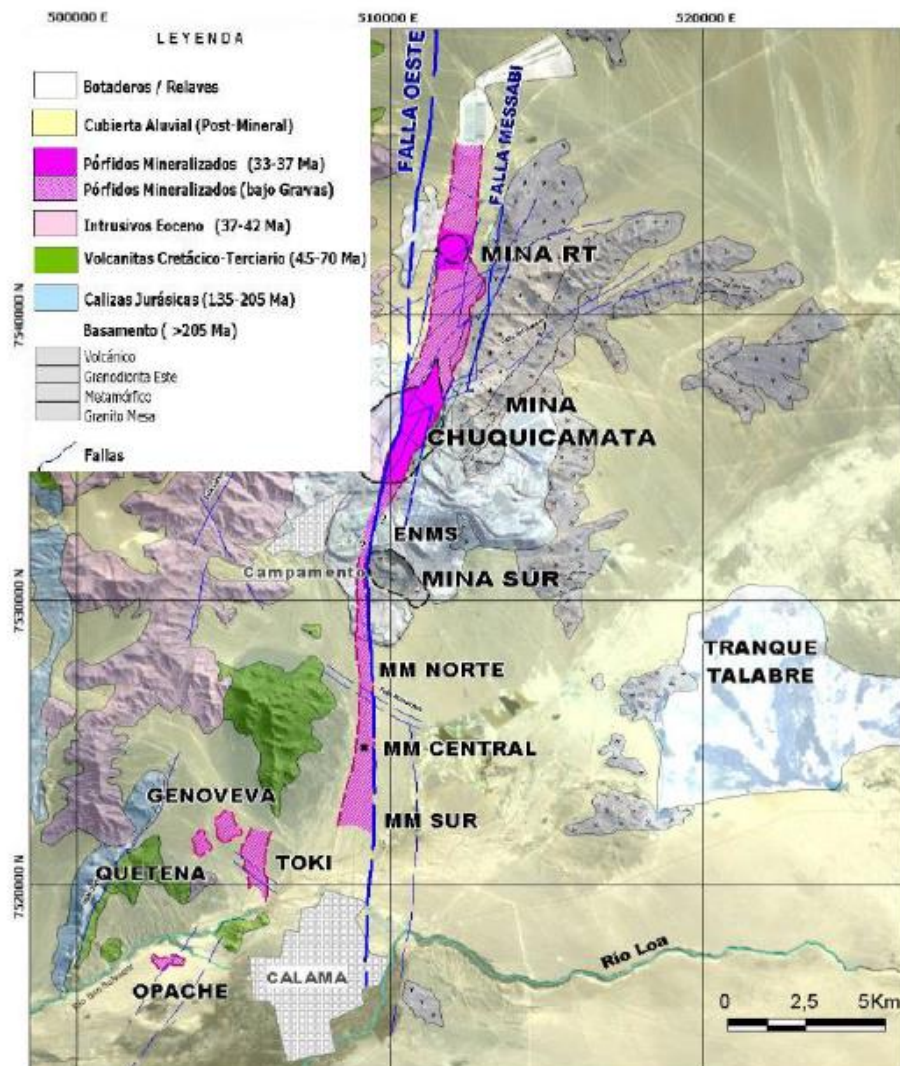


Figura 1. 2. Plano Geológico distrito Chuquicamata.

Fuente: Tomado del "Taller Geológico MMH, agosto 2015".

En términos generales, en el distrito Chuquicamata, el sistema de la Falla Oeste separa dos ambientes litológicos: en el bloque oriental predominan rocas volcánicas e intrusivas de un basamento Paleozoico, hasta ahora considerado estéril (Granodiorita Este, Granodiorita Elena, Formación Agua Dulce, Complejo Ígneo Metamórfico Chuquicamata, etc), mientras que en el

bloque occidental predominan rocas intrusivas del Eoceno (Dioritas Los Picos y Complejo Intrusivo Chuquicamata Oeste) y volcánicos del Cretácico Superior a Paleoceno (Formación Cerro Empexa e Icanche). Dentro de las rocas intrusivas del oeste es destacable la evolución espacial, composicional y temporal de los intrusivos desde el este hacia el oeste pasando por el emplazamiento del Cluster Toki y finalizando con la intrusión de Chuquicamata (figura 1.2).

Litología

a) Rocas Volcánicas

Corresponden a flujos andesíticos verdosos indiferenciados y brechas andesíticas con clorita, epidota y pirita. Esta unidad se encuentra separada en “Andesitas Occidentales”, Cerro Negro y en “Andesitas Orientales” dispuestas en una franja adosadas a la Falla Oeste. Se le asigna una edad de carbonífera a pérmica (Paleozoico Superior), U/Pb LAM-ICPMS. Bertens.

b) Rocas Intrusivas

Un complejo granodiorítico porfídico compuesto por las unidades Pórfido MM y Pórfido Cuarcífero, intruyó a rocas granodioríticas en el Eoceno. La actividad intrusiva en el depósito MMH se puso término a comienzos del Oligoceno con el emplazamiento de brechas magmáticas-hidrotermales.

c) Brechas Hidrotermales

Corresponden brechas magmáticas-hidrotermales de inyección, cuyos clastos están total o parcialmente aislados por una matriz de sulfuros, de sílice-alunita

o de polvo de roca, y que comúnmente grada externamente a una zona de “stockwork” que rodea periféricamente a las brechas.

La matriz de las brechas presenta las asociaciones de sulfuros y sulfosales de cobre de alta sulfidización y los fragmentos o clastos son predominantemente de composición dacítica.

Estas estructuras están acompañadas por un intenso desarrollo de alteración argílica avanzada, representado por sílice, alunita, pirofilita, sericita y dickita.

Estos cuerpos están zonados en la vertical, de tal forma que las brechas más profundas están definidas por la asociación mineralógica de alta sulfidización con bornita-calcosinapirita; las brechas de la zona media están representadas por la asociación calcopiritatenantita-pirita y las brechas de la parte alta están caracterizadas por la asociación calcosina-enargita-pirita-(bornita)-(covelina), figura 1.3 y 1.4. Los agregados de alteración asociados a estas zonas son invariablemente del tipo argílico avanzado.

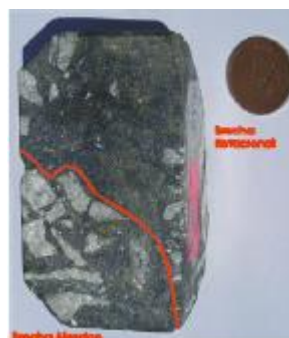


Figura 1. 3. Brechas magmáticas- hidroterales de inyección. La matriz está constituida por la asociación calcosina – enargita – pirita (alta sulfidización).

Fuente: Tomado del “Taller Geológico MMH, agosto 2015”.

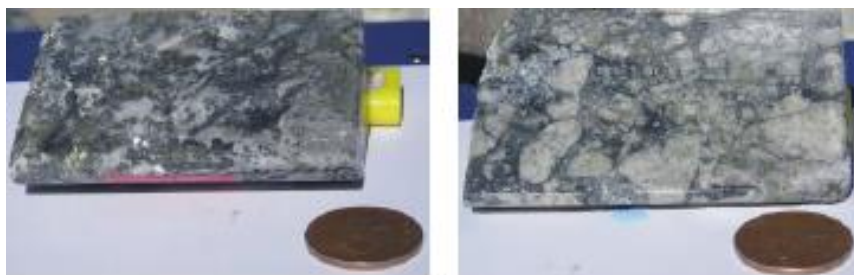


Figura 1. 4. Ejemplos de brechas magmáticas – hidrotermales ubicadas en la parte alta del yacimiento. A la izquierda la asociación mineralógica corresponde a calcosina – enargita (bornita) – pirita. La matriz en la foto de la derecha está compuesta por calcosina.

Fuente: Tomado del “Taller Geológico MMH, agosto 2015”.

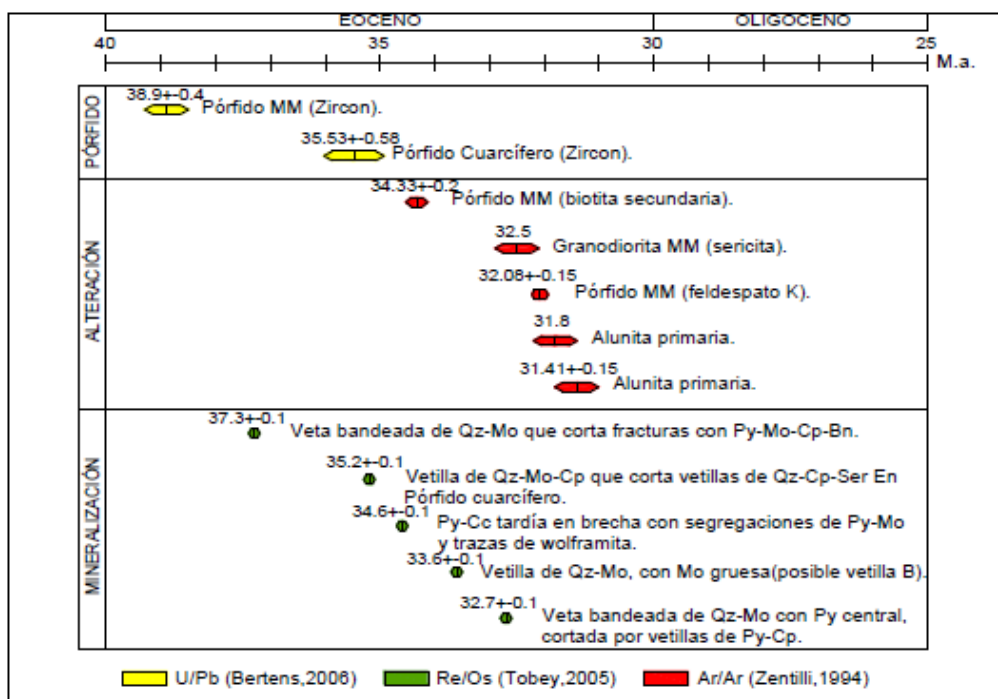


Figura 1. 5. Edades radiométricas de los pórfidos, alteración y mineralización del yacimiento MMH (modificado de Tobey, 2005).

Fuente: Tomado del “Taller Geológico MMH, agosto 2015”.

d) Brechas Fluidizadas y “Pebble Dikes”

Una serie de cuerpos menores con formas de diques que cortan a todas las rocas y a todos los estadios de mineralización-alteración acontecidos en el depósito. El control de los cuerpos tabulares post-minerales es estructural, de tal forma su emplazamiento ocurre en estructuras paralelas a la Falla Oeste. La matriz en ambas unidades consiste en material clástico, lo que sugiere que el mecanismo de fluidización fue el agua subterránea. La separación entre ellos está basada en el grado de redondeamiento de los clastos, para el caso de las Brechas Fluidizadas, y de los guijarros para los “pebble dikes”.

e) Rocas Sedimentarias

Las rocas sedimentarias presentes en el yacimiento MMH corresponden a distintos tipos de gravas, las que según su edad, composición y distribución se han agrupado informalmente en dos categorías: “Gravas Antiguas” y “Gravas Modernas”. En general estas rocas son estériles, aunque un manto de óxidos de cobre exótico se aloja en la interfase entre ambas unidades en el sector este del depósito.

f) Estructuras

- **Esquema Geológico - Estructural**

Producto de estos eventos, se han originado fallas de distintas magnitudes, las cuales se pueden dividir en mega estructuras y fallas distritales.

El sistema de Falla Oeste (o Sistema Estructural Domeyko) es la principal estructura presente en el distrito y constituye el metalotecto de primer orden

a través del cual se han emplazado los principales eventos de mineralización de cobre porfídico. Este sistema es reconocido entre el Salar de Punta Negra y Quebrada Blanca, alcanzando una corrida cercana a los 500 km. Dentro del distrito, la Falla Oeste corresponde a una estructura NS / subvertical, cuya traza ha sido reconocida en diversos afloramientos y atravesada mediante sondajes, rajos y galerías subterráneas. En MMH, la Falla Oeste presenta un rumbo norte sur a $N15^{\circ} W$ y un manteo subvertical de 75 a 80° al oeste. La estructura posee un espesor de 1 m de salbanda gris-roja (cizalle puro) y una zona de daño en el bloque pendiente (donde se emplaza la mineralización cuprífera) de 30 a 50 m de espesor.

Otros sistemas estructurales relevantes reconocidos en el distrito son el Sistema Montecristo, (NW), y el Sistema Opache (NE).

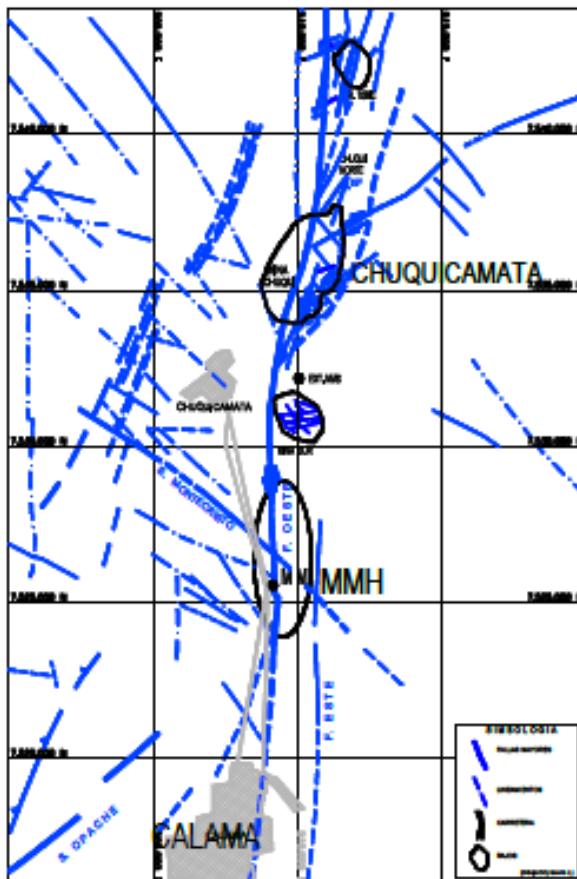


Figura 1. 6. Plano Estructural Distrito Chuquicamata.

Fuente: Tomado del "Taller Geológico MMH, agosto 2015".

g) Mineralización

Las fases de mineralización y de alteración reconocidas son descritas en las figuras 1.7, 1.8 y 1.9 se muestran en dos secciones esquemáticas con las diferentes zonas y asociaciones mineralógicas descritas y una planta.

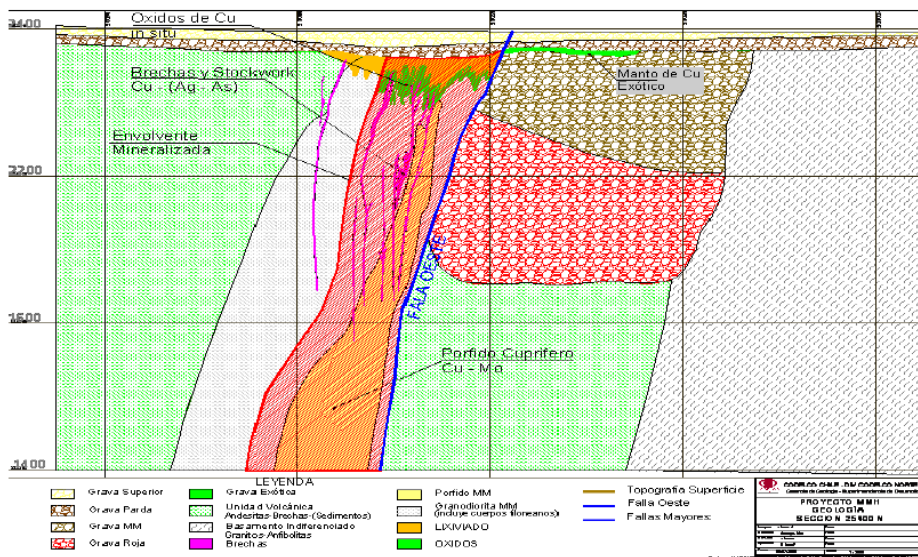


Figura 1. 7. Sección esquemática MMH Central.

Fuente: Tomado del "Taller Geológico MMH, agosto 2015".

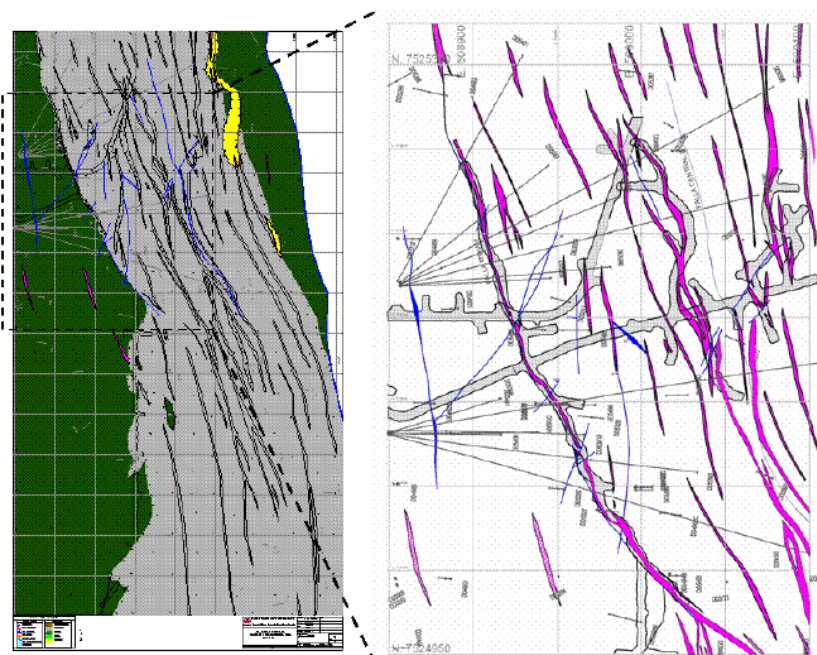


Figura 1. 8. Plano Geológico Estructural Mina Caserones.

Fuente: Tomado del "Taller Geológico MMH, agosto 2015".

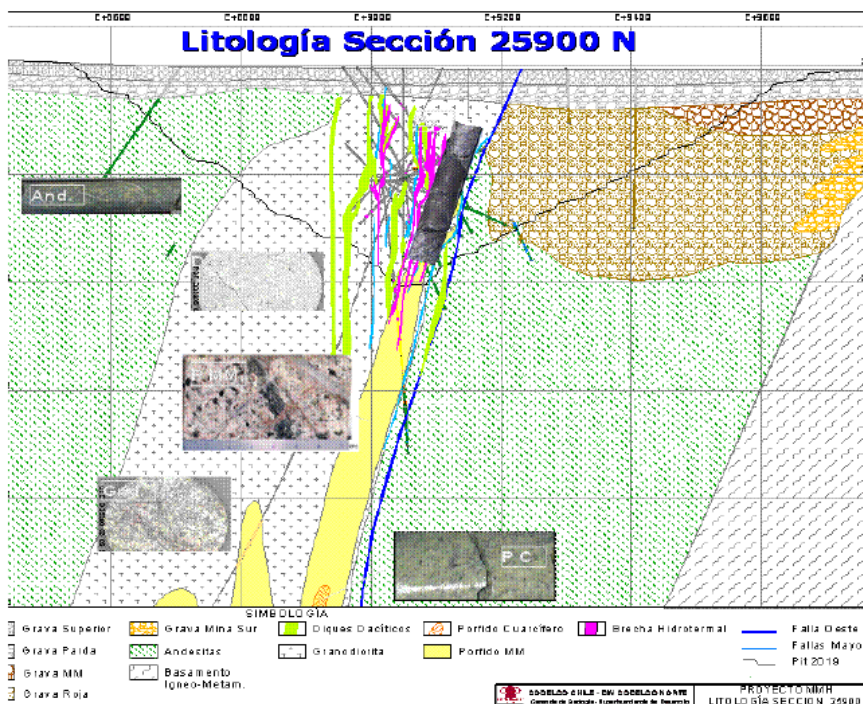


Figura 1. 9. Sección litológica tipo MMH.

Fuente: Tomado del "Taller Geológico MMH; Agosto, 2015".

Geometalurgia

Para una adecuada evaluación de los parámetros económicos de un yacimiento por parte de planificación minera, se requiere que el modelo geológico de estimación de recursos tenga incorporado algunos parámetros metalúrgicos específicos como recuperación y dureza.

a) Modelo de recuperación

El modelo de recuperaciones de MMH está basado en datos obtenidos a partir de 370 pruebas metalúrgicas, los resultados por UGM se muestran en la tabla a continuación.

Tabla 1. 1. Recuperación de mineral DMH, Codelco.

RESULTADOS TEST GEOMETALURGICOS UGM % RECUPERACION CuT							
UGE	SUB UGE	DOMINIOS					
		Alto Arsenico > 500 ppm			Bajo Arsenico < 500 ppm		
		subzona Sulfuro	Subzona Óxido	subzona Mixto	subzona Sulfuro	Subzona Óxido	subzona Mixto
BRECHAS	BXC	92.6%		76.3%	92.6%		
	BXCS						
	BXCW	91.9%					
	MY	91.9%					
C5		91.4%		71.1%	91.4%		77.0%
C1		91.5%		77.0%	89.3%		73.4%
PMM							77.0%

Fuente: (tomado del "Taller Geológico MMH, agosto 2015").

b) Modelo de durezas

Con el propósito de estudiar la moliendabilidad del mineral de MMH, se desarrolló un modelo preliminar de dureza para molienda convencional a partir de los datos de Índice de Bond (WI). Los datos fueron agrupados por dominios que se representan en sólidos 3D y en el modelo de bloques como un valor índice representativo.

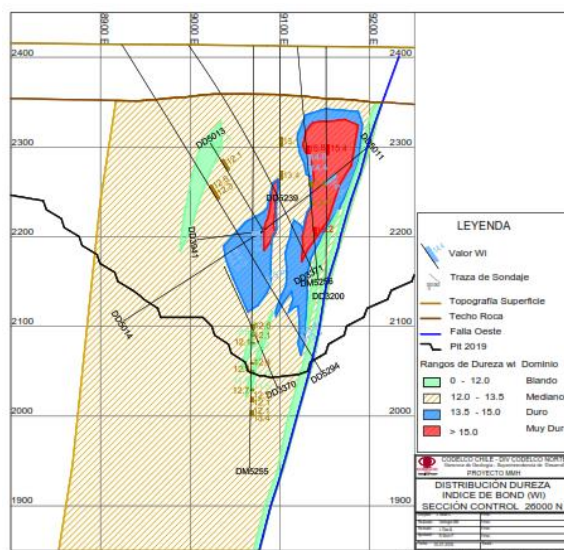


Figura 1. 10. Dominios de dureza Bond en MMH.

Fuente: Tomado del "Taller Geológico MMH, agosto 2015".

1.6.3 ETAPAS DE LA INVESTIGACIÓN

En la presente investigación se pueden reconocer las siguientes etapas:

- a) **Recolección de datos:** donde se uniformizará la data, es decir de todos los datos que se tiene disponibles, se comienza a generar un solo archivo recolectando la información relevante o necesaria para el objetivo de la presente tesis.
- b) **Procesamiento de datos:** donde se realizará limpieza de datos, eliminación de valores faltantes, análisis estadístico y uniformización de variables.
- c) **Análisis de la información:** donde se realiza la descripción de los resultados, descripción del modelo predictivo, análisis del comportamiento de la variable.

De acuerdo con estas etapas de la investigación, se tiene el paso a paso de lo que se realiza en la presente tesis:

1. Planteamiento del problema
2. Análisis de datos
3. Tratamiento de datos
4. Modelamiento
5. Análisis y discusión de resultados.

1.6.4 MATRIZ DE CONSISTENCIA

Tabla 1. 2. Matriz de consistencia.

FORMULACION DEL PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES	INDICADORES	DISEÑO METODOLOGICO
<p><u>PROBLEMA GENERAL</u></p> <p>¿Qué influencia tiene el sobredimensionamiento de recursos debido a que las aplicaciones actuales utilizadas para el control de la fragmentación en voladura no consiguen el objetivo total?</p>	<p><u>OBJETIVO GENERAL</u></p> <p>Diseñar una malla de voladura con control de la fragmentación de rocas en la voladura de minas a tajo abierto.</p>	<p><u>HIPOTESIS GENERAL</u></p> <p>Un modelo basado en redes neuronales artificiales permitirá cumplir óptimamente el objetivo de fragmentación de rocas en voladura.</p>	<p>Variable dependiente:</p> <p>Diseño de malla de voladura con control en la fragmentación en base a factor de carga predicho mediante redes neuronales. (y).</p>	<p><u>INDICADORES</u></p> <p>Variable dependiente</p> <p>Y: Factor de carga = g explosivo/ ton removidas</p> <p>Error cuadrático medio, error medio absoluto, R²</p>	<p><u>Tipo de Investigación</u></p> <p>Aplicada</p> <p><u>Nivel de Investigación</u></p> <p>Estudio de caso</p> <p><u>Diseño de la Investigación</u></p> <p>Experimental</p>
<p><u>PROBLEMAS ESPECÍFICOS</u></p> <p>1. ¿Qué beneficios conlleva interpretar la correlación entre los parámetros de roca y variables de diseño para la fragmentación y diseño de una malla de voladura?</p> <p>2. ¿Cuál es el impacto al generar un ahorro de los recursos disponibles para el caso específico de diseño de mallas mediante un nuevo modelo predictivo?</p> <p>3. ¿Qué influencia tiene contar con un nuevo método de diseño de malla que se adecue completamente a las variaciones de las condiciones geológicas del terreno?</p> <p>4. ¿De qué manera es posible evitar una ubicación no optimizada de los holes de perforación en una malla de voladura?</p>	<p><u>OBJETIVOS ESPECÍFICOS</u></p> <p>1. Interpretar la correlación entre los parámetros de roca como la resistencia mecánica de la roca y las variables de diseño, para determinar su influencia sobre la fragmentación y diseño de malla.</p> <p>2. Generar un ahorro en los metros perforados a través de un diseño de malla optimizada mediante el método de redes neuronales artificiales que se propondrá.</p> <p>3. Formular un modelo de aprendizaje automático que considere condiciones geológicas de terreno a mayor detalle que los modelos actuales y que sea capaz de realizar recomendaciones para optimizar el dimensionamiento de la malla de voladura.</p> <p>4. Desarrollar un modelo de redes artificiales que prediga el Factor de Carga y proporcione la ubicación de los holes de perforación optimizados en base a la variabilidad del UCS.</p>	<p><u>HIPOTESIS ESPECÍFICAS</u></p> <p>1. La fragmentación de las rocas depende principalmente de la resistencia mecánica de la roca y la carga de explosivo por su relación directa a la energía liberada que impacta sobre las rocas del terreno.</p> <p>2. Un modelo de aprendizaje automático genera un ahorro en los metros perforados por una mejor precisión.</p> <p>3. Un modelo de aprendizaje automático es capaz de recomendar las condiciones que se requieren para optimizar el dimensionamiento de la malla de voladura y garantiza la fragmentación requerida.</p> <p>4. La red neuronal artificial, en base a la variabilidad del UCS, predice el Factor de Carga, lo que implica el cambio espacial del diseño de malla y ello genera la ubicación óptima de los hole de perforación.</p>	<p>Variable independiente</p> <p>Valores óptimos de factor de carga para diseño de una malla de voladura: UCS (x1) Carga del explosivo (x2), P50 (x3), P60 (x4), P80 (x5), P95 (x6).</p>	<p>Variable independiente</p> <p>X1: Porcentaje</p> <p>X2: Porcentaje</p> <p>X3: Porcentaje</p> <p>X4: Porcentaje</p> <p>X5: Porcentaje</p> <p>X6: Porcentaje</p>	<p><u>Técnicas e instrumentos de recolección.</u></p> <p>*Observación</p> <p>*Experimentación</p> <p><u>Instrumento</u></p> <p>Base de datos</p> <p>Se genera un modelo de redes neuronales artificiales sobre los datos de voladuras ejecutadas, y se genera el aprendizaje necesario para cumplir el objetivo.</p>
<p>Matriz de consistencia de tesis</p> <p>TITULO: MACHINE LEARNING APLICADO AL CONTROL DE LA FRAGMENTACIÓN DE ROCAS EN LA VOLADURA DE MINAS A TAJO ABIERTO</p>					

Fuente: Elaboración propia.

1.7 TRABAJOS PREVIOS

En publicaciones previas se han hecho predicciones usando herramientas de machine learning como las redes neuronales artificiales con el objetivo de obtener el tamaño de fragmentación de rocas como resultado del proceso de voladura. En estas publicaciones el burden y espaciamiento forman parte de los inputs para predecir la fragmentación mientras que en la presente tesis el burden y espaciamiento es el objetivo final, donde este se obtendrá a partir del factor de carga.

Varios investigadores han empleado una serie de ecuaciones empíricas y estadísticas convencionales y sistemas de redes neuronales artificiales para predecir la fragmentación de la roca, la vibración del suelo y el air-blast antes de las operaciones de voladura. Sin embargo, se prefiere la red neuronal artificial (ANN) sobre las otras técnicas predictivas debido a su capacidad para incorporar los numerosos factores que afectan el resultado de una voladura, entre otras ventajas (Tiile ,2016).

Otro ejemplo es el de K. Oraee donde también se realizó la predicción de la fragmentación de roca resultado de un proceso de voladura en tajo abierto, a través del uso de redes neuronales artificiales. Se muestra que el método de red neuronal se puede utilizar de manera eficiente en tales casos y se puede esperar que los resultados finales tengan un alto grado de precisión (K. Oraee, 2006).

1.8 ESTADO DEL ARTE

Teniendo en cuenta los estudios previos realizados estos se enfocan en predecir la fragmentación de la roca resultado de una voladura para poder así tener un control de la fragmentación y ser mucho más optimizado en sus diseños.

Para efectos del presente; se considera como dato de entrada la fragmentación que se desea obtener, y se predice las condiciones de diseño de voladura específicamente

la variable factor de carga que nos da el valor del burden y espaciamento, es decir la ubicación de los pozos a perforar.

Esta predicción está controlada por la variable UCS, Carga de Explosivo, Altura de banco y target de fragmentación.

Con el algoritmo desarrollado se cumplen las condiciones mucho más óptimas para aprovechar al máximo los recursos disponibles.

CAPITULO II.

MARCO TEÓRICO Y CONCEPTUAL

2.1 MARCO TEÓRICO

La voladura es un proceso injerto en el negocio minero y sobre el cual tiene un gran impacto en el corto como en el largo plazo. La voladura en una mina a cielo abierto comienza generando un diseño de voladura donde a corto plazo se busca lograr un adecuado grado de fragmentación de la roca, de tal modo que haga mínimo el costo combinado de las operaciones aguas abajo (Carguío, Transporte, Chancado y Molienda). Antes de realizar una voladura se tiene la etapa de perforación, la perforación es la primera operación en la preparación de una voladura (Karlinski et al., 2009). Esta es la etapa previa antes de ejecutar la voladura y toma en cuenta para su realización el diseño elaborado por los especialistas, donde cada hole perforado servirá para introducir el explosivo necesario para generar la fragmentación deseada.

Las tecnologías en perforación más empleadas y desarrolladas se basan en sistemas de perforación mecánicos, conocidos como sistemas de perforación “a rotación” y “a percusión” (J.Bernaola, 2013).

En voladura, se deben considerar varios factores al diseñar una voladura e incluyen las necesidades de fragmentación, la geología, las estructuras cercanas, la integridad de las paredes, el tipo de explosivo, las consideraciones de vibración, de air-blast y el tipo de equipo de perforación (Tiile, 2016). Dentro de estas variables para realizar una voladura eficiente, se debe tener en cuenta la distribución de la energía en el macizo rocoso, ya que esta debe ser distribuida de manera uniforme y el parámetro que indica la energía distribuida es el factor de carga. También se debe tener en cuenta el confinamiento de la energía en el macizo rocoso, a fin de que la energía explosiva pueda fracturar y desplazar el material y que el nivel de energía explosiva debe ser suficiente para vencer la fuerza estructural de la roca y permitir su desplazamiento

El objetivo a mediano y largo plazo es minimizar el daño al macizo rocoso en su entorno, protegiendo la integridad de los techos, pilares, taludes y la estabilidad en general, disminuyendo los riesgos sobre las personas y equipos, cautelando las instalaciones/ infraestructuras más próximas y disminuyendo costos operativos.

Para cumplir el objetivo a corto plazo de obtener el adecuado grado de fragmentación que minimiza el costo combinado de las operaciones mencionadas líneas arriba y a la vez aprovechar al máximo lo invertido en la perforación usando por completo el espacio perforado en cada hole, minimizando aún más los costos operativos, es que se busca conocer la mejor distribución espacial de los holes que se perforarán cumpliendo los objetivos mencionados de target de fragmentación y aprovechamiento máximo de la carga que se puede tener en cada hole.

2.1.1 MACHINE LEARNING

Existen definiciones distintas pero similares a través de los años para el término Machine Learning también conocido como aprendizaje automático como las siguientes, el aprendizaje automático es la ciencia de conseguir que las computadoras aprendan y actúen como los humanos, y mejoren su aprendizaje con el tiempo de manera autónoma, alimentándose con datos e información en forma de observaciones e interacciones del mundo real (Jatana, 2018) o el aprendizaje automático (ML) es el estudio de algoritmos informáticos que mejoran automáticamente a través de la experiencia (Mitchell,1997), de estas se pueden concluir principalmente que su significado es datos que aprenden con la experiencia. Estamos en una era donde la abundancia de datos hace posible el auge del uso de esta herramienta.

De este modo, a partir de una base de datos ya sean datos estructurados como csv o no estructurados como una imagen o contenido multimedia se pueden construir modelos de aprendizaje automático para predecir variables de interés.

La aplicación que se presenta en la presente tesis ha sido trabajada sobre datos estructurados, donde podemos identificar las variables independientes y la variable target u objetivo dependiente de las otras, siendo este un ejemplo de aprendizaje supervisado que es uno de los tipos de aprendizaje que posee el aprendizaje automático.

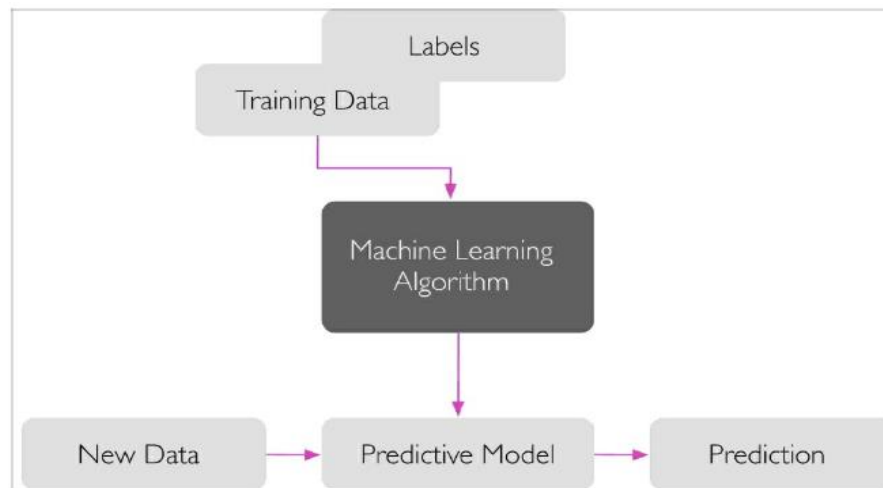


Figura 2. 1. Diagrama del funcionamiento de un algoritmo de machine learning en un aprendizaje supervisado.

Fuente: Tomado de Raschka, Sebastian & Mirjalili, Vahid (2017).

2.1.2 APRENDIZAJE SUPERVISADO

El aprendizaje supervisado recibe este nombre debido a que dentro de nuestra base de datos sobre la cual construiremos el modelo la variable target u objetivo es seleccionada por el usuario e introducida al modelo juntos con los inputs o variables independientes, todas estas variables mediante el uso de los algoritmos de machine learning previo tratamiento de datos permitirán predecir la variable de interés. El objetivo principal del aprendizaje supervisado es aprender un modelo a partir de datos de entrenamiento etiquetados que nos permita hacer predicciones sobre datos futuros o no vistos (Raschka, Sebastian & Mirjalili, Vahid 2017). Dentro del aprendizaje supervisado podemos resolver dos tipos de problemas: regresión y clasificación.

Podemos entender mejor este tipo de aprendizaje con un ejemplo simple, pensemos en categorizar imágenes de gatos y perros.

Para enseñar al algoritmo que es un gato y que es un perro, al algoritmo se le debe presentar miles de ejemplos de cada uno, y cada imagen debe ser etiquetado como un “gato” o un “perro”. En el ejemplo la etiqueta es el output deseado para cada imagen, y analizando cada ejemplo el algoritmo será eventualmente capaz de clasificar nuevas imágenes de gatos y perros con un mínimo error.

2.1.2.1 REGRESIÓN

Los problemas de regresión son aquellos donde la variable target u objetivo a predecir es un valor numérico continuo, sobre el cual se construirá un modelo que represente el comportamiento de las variables de entrada tal que generen el output del valor del target.

Algunos algoritmos para problemas de regresión son: Regresión Lineal, Regresión Lineal multivariable, Ridge Regression, Lasso, Elastic Net, Redes Neuronales, Decision Tree Regressor y Support Vector Regression, a continuación, una breve mención acerca de algunos de estos algoritmos de regresión.

El algoritmo de regresión lineal supone una relación lineal o en línea recta entre las variables de entrada (X) y la variable de salida única (y). El algoritmo de regresión lineal supone que las variables de entrada tienen una distribución gaussiana. También se supone que las variables de entrada son relevantes para la variable de salida y que no están altamente correlacionadas entre sí.

Ridge Regression: Es un modelo lineal generalizado a través de la máxima probabilidad penalizada que funciona para problemas de regresión. El algoritmo es extremadamente rápido y puede explotar la dispersión en la

matriz de entrada x . Se ajusta a los modelos de regresión lineal, logística y multinomial, poisson y de Cox. Ridge Regression es una extensión del algoritmo de regresión lineal donde la función de pérdida se modifica para minimizar la complejidad del modelo medido como el valor de la suma cuadrática de los valores del coeficiente (también llamada norma L2).

LASSO es una modificación de la regresión lineal, como también lo es Ridge Regression donde la función de pérdida se modifica para minimizar la complejidad del modelo medido como el valor absoluto de los valores del coeficiente (también llamada la norma L1).

2.1.2.2 CLASIFICACIÓN

Los problemas de clasificación son aquellos donde la variable target u objetivo posee 2 o más clases categóricas y el objetivo del modelo es predecir a qué clase pertenece la variable target dependiendo de las variables de entrada que se introduzcan. En estos problemas se usan algoritmos distintos a los usados en los problemas de regresión.

Algunos algoritmos para problemas de clasificación son: Linear Discriminant Analysis, K-Nearest Neighbors, Support Vector Classifier, Redes Neuronales, Naive Bayes y Decision Tree Classifier.

A continuación, una breve descripción de algunos de estos algoritmos de clasificación.

Árboles de clasificación y regresión o CART, La representación del modelo CART es un árbol binario. Un nodo representa una sola variable de entrada (X) y un punto de división en esa variable, asumiendo que la variable es numérica. Los nodos hoja (también llamados nodos terminales) del árbol contienen una

salida variable (y) que se usa para hacer una predicción (Brownlee, Jason (2017)).

K Vecinos cercanos, Cuando se requiere una predicción, se localizan los k registros más similares a un nuevo registro del conjunto de datos de entrenamiento. A partir de estos vecinos, se hace una predicción resumida. La similitud entre registros se puede medir de muchas formas diferentes (Brownlee, Jason (2017)).

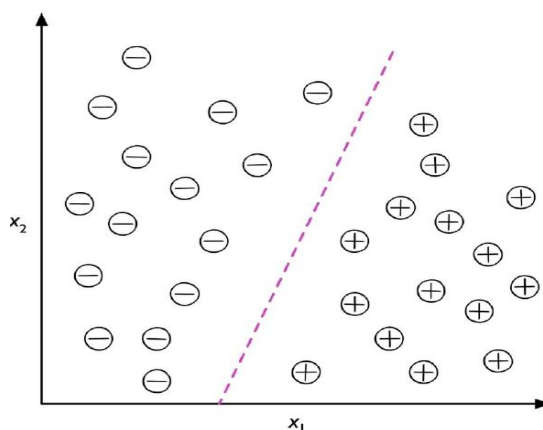


Figura 2. 2. Clasificación supervisada (Raschka, Sebastian & Mirjalili, Vahid (2017)).

Fuente: Tomado de Raschka, Sebastian & Mirjalili, Vahid (2017).

2.1.3 APRENDIZAJE NO SUPERVISADO

El aprendizaje no supervisado recibe este nombre debido a que los modelos de solución de este tipo trabajan con un dataset de parámetros, pero no tienen un target u objetivo escogido, en cambio lo que busca el modelo es encontrar estructuras y patrones por sí mismo entre los datos que se están usando para la realización del modelo.

Utilizando técnicas de aprendizaje no supervisadas, podemos explorar la estructura de nuestros datos para extraer información significativa sin la guía de una variable de resultado conocida (Raschka, Sebastian & Mirjalili, Vahid 2017). Para entender mejor este tipo de aprendizaje se muestra el siguiente ejemplo: Si tenemos una imagen a color podemos clusterizar la imagen para que esta imagen está representada solo en 2 colores, 3 colores o el número de clusters que queramos, no tenemos una variable que etiquetar o definir más bien el algoritmo encuentra el patrón en los datos para clusterizar correctamente como una función de distancia.

Dentro de este tipo de aprendizaje encontramos con los algoritmos de clustering o agrupamiento que pueden ser jerárquicos o no jerárquicos.

Dentro de los algoritmos jerárquicos podemos encontrar los aglomerativos y los divisivos, mientras que los no jerárquicos en los que el número de clusters o grupos se designa de antemano y sus etiquetas son función de distancia, podemos encontrar algoritmos como el K-Means.

La presente tesis se centra en una de las aplicaciones de aprendizaje supervisado donde etiquetamos el valor del factor de carga, que será la variable a predecir, para esto usamos las bases de datos extraídas de las mallas de voladuras pasadas y los concatenamos con dataset extraídos del programa GBlast de las mismas voladuras pasadas, esta data unida servirá como conjunto de entrenamiento y validación.

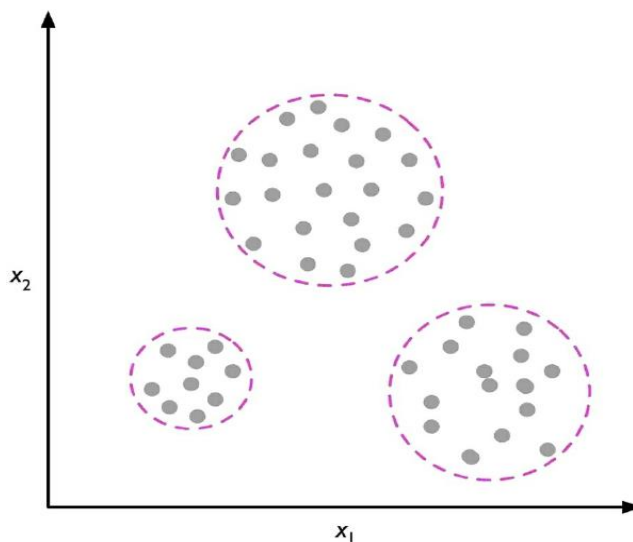


Figura 2. 3. Ejemplo del uso de un algoritmo de aprendizaje no supervisado, que ha clusterizado los datos sin necesidad de tener sus etiquetas.

Fuente: Tomado de Raschka, Sebastian & Mirjalili, Vahid (2017).

2.1.4 VARIABLES USADAS

Para poder desarrollar el programa de redes neuronales es necesario definir los parámetros que vamos a utilizar dentro de todas las variables que se consideran en el diseño de voladura. Muchos de los parámetros de diseño de voladuras se estiman en base a fórmulas empíricas y experiencia (Tiile, 2016).

2.1.4.1 PROPIEDADES DE LA ROCA INTACTA:

La resistencia y la dureza expresadas por el UCS y PLT, la densidad de la roca, el módulo de elasticidad de Young, así como su respuesta física que condiciona su fracturamiento ante la energía liberada y la presión que soporta por los gases expulsados al realizar la detonación.

2.1.4.2 LAS CONDICIONES ESTRUCTURALES DEL MACIZO ROCOSO.

El comportamiento del macizo rocoso dependerá y variará según este se encuentre en estado masivo, estratificado, en bloques, fracturado, homogéneo o variable. Ya que cada tipo de condición estructural está asociado a un tipo de distribución de las zonas de debilidad en el macizo rocoso y por lo tanto una respuesta física diferente frente a la energía liberada durante la detonación. Ya que cuando una carga explota, genera gases de alta presión y temperatura contra el pozo de contención, creando así un campo de tensión en la roca. La acumulación de presión del pozo de perforación depende de las características físicas de la roca y de la composición explosiva (Tiile, 2016).

2.1.4.3 CONDICIONES EN TERRENO.

Los accesos para las maquinarias, la proximidad a estructuras y maquinarias, y la presencia de agua son factores que influyen directamente sobre la operatividad y el aspecto técnico de la perforación.

2.1.4.4 PERFORADORAS.

Tipo de perforadora, largo máximo del pozo, diámetros de pozos, orientación de pozos y capacidad, variables que generalmente están estandarizadas pero que sin embargo el cambio en una de estas modifica todo el diseño.

2.1.4.5 EXPLOSIVOS.

Tipo de explosivos, propiedades de la detonación y propiedades físicas.

2.1.4.6 GEOMETRÍA DEL DISEÑO.

Tamaño y forma del diseño, las caras libres, las mallas y el decking.

En las variables que se mencionó tenemos parámetros sobre los cuales podemos tener control como la carga del explosivo, el factor de carga, el burden y el espaciamiento. Mientras que existen otras variables que no podemos controlar que son intrínsecas de la geología del terreno como la resistencia de las rocas expresados por el pseudo UCS que nos entrega el programa GBlast a partir de los signos vitales de la perforadora.

Ahora bien, habrá que definir cuáles serán nuestras variables independientes y cuál es nuestro target. Para la presente tesis se busca predecir la mejor configuración de la ubicación de los pozos a fin de usar el taco crítico en cada hole o pozo lo que implica que la carga de explosivo es máxima por hole. Usamos el parámetro del pseudo UCS para caracterizar el tipo de litología presente en cada punto del terreno.

Usamos los resultados de fragmentación X50, X60 y X90 como parámetros de aprendizaje del modelo para tener un control de comportamiento del terreno según el diseño de voladura que se usó. El valor de la carga usado en cada hole es también necesario para el modelo para asociarla a la energía liberada en la voladura. Finalmente, nuestra variable queda define la posición de los pozos en la malla es el factor de carga ya que esta se encuentra directamente asociados al producto de las variables burden y espaciamiento, cuya relación burden-espaciamiento para el diseño puede tomar solo algunos valores como

1:1 o 1:1.15; por lo tanto, la variable factor de carga será nuestro target ya que esta engloba la ubicación de cada hole en su cálculo.

Como consecuencia logramos generar un diseño de malla que cumpla las condiciones de fragmentación que deseamos, y así también usamos la carga con el taco crítico en cada pozo, esto con el algoritmo de red neuronal artificial que ha aprendido de las experiencias de voladuras anteriores.

2.1.5 HIPÓTESIS DE ALGORITMOS NECESARIOS

Para poder desarrollar el programa de machine learning que permitan obtener el target deseado planteamos la lógica de solución y elegimos qué algoritmos son los que permiten ejecutar esta solución. El target es el factor de carga porque nos permite conocer los valores del burden y espaciamiento que es nuestro objetivo, el factor de carga es una variable de tipo numérica que se predecirá como una función continua de los parámetros independientes (la carga, la fragmentación X50, X60, X90, el pseudo UCS y la carga de explosivo usado que son variables numéricas, es decir tenemos un target numérico en función de los otros parámetros, por lo tanto nos encontramos frente a un problema de regresión con varios parámetros de entradas y lo resolveremos usando una red neuronal artificial de regresión.

2.1.5.1 DEFINICIÓN DEL PROBLEMA EN PERSPECTIVA DE APRENDIZAJE AUTOMÁTICO

El target es una variable numérica que es una función continua de los parámetros independientes. Por lo cual estamos ante un problema de regresión, existen distintos tipos de algoritmos especializados en la resolución

de este tipo de problema pueden ser algoritmos lineales como los algoritmos de regresión lineal, lasso, elastic net, regresión ridge y algoritmos no lineales como las redes neuronales artificiales de regresión, escogeremos el segundo tipo de algoritmo (red neuronal para regresión) por tener mayor flexibilidad en cuanto al modelado, esta herramienta también modela un comportamiento lineales.

Nuestros datos son parámetros asociados a las respuestas que tuvo la roca en cada hole después de la detonación, es decir cada instancia o fila de la base de datos es un ejemplo o experiencia de aprendizaje para la red neuronal que se entrenará y permitirá el aprendizaje del modelo. Esta instancia o fila contiene un dato de cada parámetro asociado al comportamiento de la roca en ese hole.

2.1.5.2 ALGORITMOS NECESARIOS PARA EL PROBLEMA

Al ser un problema de regresión tenemos muchos algoritmos de aprendizaje que podemos utilizar. Dada la complejidad del comportamiento de las variables no se utilizará algoritmos lineales de regresión más bien redes neuronales para regresión. Algunos de los principales algoritmos de regresión se mencionaron anteriormente. En este tipo de algoritmos la métrica de la predicción se basa en la función de error que compara el valor predicho con el valor experimental.

2.2 MARCO CONCEPTUAL

2.2.1 DEFINICIÓN DE TÉRMINOS

POZO

Se denomina pozo a una perforación en la roca, de largo y diámetro predeterminado, identificado con un código (ID de pozo), el cual es cargado con explosivo para la voladura. Cada pozo tiene una ubicación única en el espacio (x,y,z).

HOLE

Un hole es un pozo por su traducción al idioma inglés.

EXPLOSIVO

Cualquier mezcla química que reacciona a alta velocidad para liberar gas y calor y que causa grandes presiones. Los explosivos industriales son mezclas de cuerpos sólidos y líquidos.

MALLA DE VOLADURA

Se define como malla al sector de banco donde se realizará la voladura definida por un conjunto de pozos cargados con explosivos y listos para ser detonados.

MATERIAL DE TACO

Material particulado (fragmentos y polvo de roca) producto de la perforación que se acumula en el entorno del collar del pozo y que es utilizado en ciertas ocasiones para el tapado de los pozos cargados con explosivo. Se recomienda el uso de material gravilla en reemplazo del detritus debido a que permite una

óptima retención de taco durante la voladura aprovechando mejor la energía del explosivo.

TACO

Corresponde a la longitud sin carga explosiva del pozo la cual es rellenada con material inerte (detritus de perforación, gravilla, etc.). La función principal del taco es confinar el explosivo.

VOLADURA

Proceso que comprende todas las operaciones necesarias para la fragmentación de rocas mediante el uso de explosivos. También se refiere a la malla de perforación que posteriormente será volada (malla de disparo).

MALLA DE PERFORACIÓN

Zona en la unidad minera debidamente definida y limitada, la cual es perforada de acuerdo con un diseño definido en forma técnica.

ID

Se refiere a la identificación o código del pozo o hole en una malla de voladura. El ID es un número por lo que podría repetirse dentro de una malla de voladura.

FACTOR DE CARGA

El factor de carga es el indicador que mide la cantidad de explosivo utilizado por tonelada de roca fragmentada.

CLUSTERIZACIÓN

Al agrupamiento de variables en machine learning se le denomina clusterización.

REDES NEURONALES

Son algoritmos de machine learning diseñados para reconocer patrones y comportamientos. Una red neuronal está interconectada por nodos, donde cada nodo es capaz de realizar cálculos, y al llevar un orden cada uno de estos nodos, finalmente una red neuronal es capaz sacar sus propias conclusiones y de tomar decisiones precisas por sí mismas.

DISEÑO DE MALLA

Diseño de malla se refiere a definir la ubicación de pozos y carga de explosivo en un sector a explotar de acuerdo con los requerimientos que se desee.

UCS

UCS es la resistencia a la compresión uniaxial de la roca intacta. El UCS es una de las propiedades más conocidas de la roca y se refiere al esfuerzo máximo en la dirección de un solo eje axial que puede soportar la roca antes de romperse (Geotecnia, Vol. 7(3): 43-48 (2004)).

PLT

PLT es el ensayo de carga puntual que se le realiza a la roca. El PLT es uno de los parámetros obtenidos de la base de datos del modelo de bloques de la unidad minera, el cual no fue utilizado en el modelo predictivo.

DECKING

Cargas explosivas discontinuas dentro de un pozo separadas por bolsas de aire o taco intermedio. Los pozos con un solo decking significa que solo tienen una carga explosiva continua dentro del pozo; por otro lado, si un pozo tiene 02 decking quiere decir que el pozo contiene 02 cargas explosivas separadas por una bolsa de aire o por un taco intermedio.

CSV

Es un archivo de valores delimitado por comas de formato excel que puede ser leído por diversos softwares de minería. Un archivo csv contiene diferente tipo de información y para el presente trabajo la información que contiene el archivo csv son los valores del pseudo UCS estimado en el área de perforación para los pozos perforados en la unidad minera, y la información de todos los pozos/holes de las mallas de voladura (ID de Pozo, ubicación de pozo: este, norte, elevación, collar, profundidad); siendo esta información utilizada para construir el modelo predictivo.

PANDAS

Es una librería del lenguaje de programación de python utilizado para el procesamiento de la base de datos.

NAN

Nan se refiere a valores faltantes en el archivo.

TARGET

Representa el objetivo que se desea lograr en las diferentes etapas del proceso del modelo predictivo.

P50

Es el tamaño de abertura de malla en la que pasa el 50% del material fragmentado. En la presente tesis, el archivo csv lo representa como X50.

P60

Es el tamaño de abertura de malla en la que pasa el 60% del material fragmentado. En la presente tesis, el archivo csv lo representa como X60.

P80

Es el tamaño de abertura de malla en la que pasa el 80% del material fragmentado. En la presente tesis, el archivo csv lo representa como X80.

P95

Es el tamaño de abertura de malla en la que pasa el 95% del material fragmentado. En la presente tesis, el archivo csv lo representa como X95.

CAPITULO III.

ANÁLISIS DE DATOS

Para realizar un programa de machine learning es necesario hacer un detallado análisis, procesamiento y transformación de los datos sobre los cuales se construirá el modelo de machine learning, empezado por conocer la anatomía del dataset con el que se trabajará, la tabla a continuación nos muestra la base de datos con los parámetros de entrada (inputs) y la variable de salida (output), a continuación se detallan las principales consideraciones a seguir para poder realizar un programa de machine learning las cuales se tuvieron en cuenta al momento de realizar el modelo de red neuronal artificial.

Tabla 3. 1. La tabla muestra el dataset de los datos utilizados y podemos observar en cada columna los parámetros utilizados.

	UCS [MPa]	x95[cm]	x80[cm]	x60[cm]	x50[cm]	Carga[Kg]	Factor de Carga [g/ton]
0	3.85	19.66	10.46	5.90	4.444	972.0	315.0
1	30.68	17.42	9.77	5.79	4.462	1047.0	329.0
2	36.54	17.21	9.69	5.76	4.448	1065.0	325.0
3	70.78	18.03	10.00	5.86	4.498	998.0	311.0
4	105.96	17.21	9.69	5.76	4.448	1065.0	338.0
...
1080	146.06	17.50	9.80	5.80	4.466	1041.0	356.0
1081	84.79	17.12	9.66	5.75	4.441	1074.0	364.0
1082	145.18	17.32	9.73	5.77	4.454	1056.0	361.0
1083	192.19	17.62	9.85	5.81	4.473	1031.0	348.0
1084	155.16	17.62	9.85	5.81	4.473	1031.0	349.0

1085 rows × 7 columns

Fuente: Elaboración propia.

3.1 RECOLECCIÓN DE DATOS

3.1.1 DATOS

Los datos disponibles pertenecen a los usados para voladura en una unidad minera al norte de Chile, estas son las mallas usadas para la voladura que contienen la ubicación de los holes en coordenadas este, coordenadas norte y elevación, los signos vitales de la perforación de cada hole y el modelo de bloques que se tiene del área de voladura. Estos datos son procesados en el programa GBlast para obtener valores de entrada que usará la red neuronal para su aprendizaje.

3.1.2 GBLAST

El software GBlast es un programa desarrollado por la empresa Geoblast que permite obtener el valor del pseudo UCS a partir de los signos vitales de las máquinas de perforación como el torque, air, rop, rpm, etc y recomienda la carga de explosivo que debe ser introducida en cada hole para obtener la fragmentación deseada.

Para este fin lo primero es cargar el modelo de bloques del área donde se han hecho el diseño de las mallas de voladura en el programa GBlast. Seguido a esto cargar una malla de voladura, esta malla debe contener todos los signos vitales tomados en la perforación de los pozos, a manera de un resumen de perfil de pozo como se muestra en el anexo 1.

Al cargar la malla ya podemos visualizarla como se muestra a continuación para la malla de código Q1.

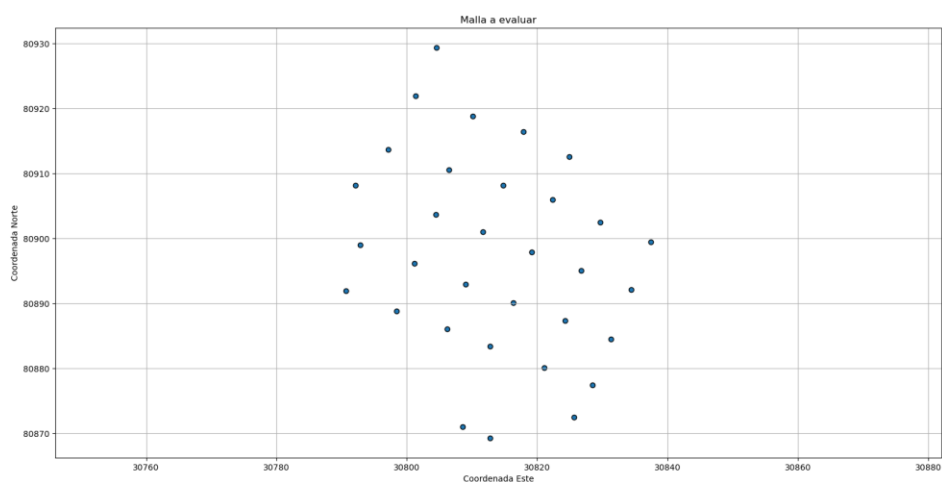


Figura 3. 1. Malla a evaluar.

Fuente: Obtenido a través del software Gblast.

Utilizamos el programa de Gblast para generar el pseudo UCS del terreno a partir de los signos vitales de los taladros perforados y obtenemos el siguiente mapa de calor correspondiente al pseudo UCS del área perforada (malla Q1).

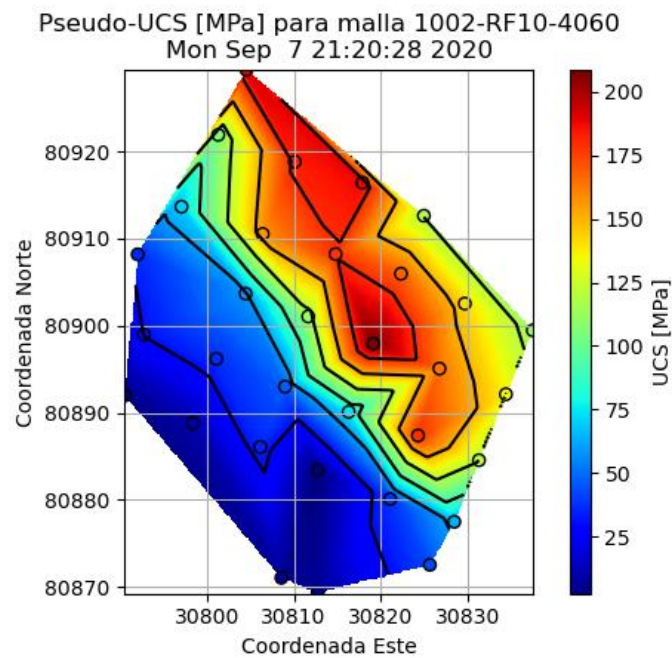


Figura 3. 2. Pseudo UCS generado por el Gblast para la malla Q1.

Fuente: Obtenido a través del software Gblast.

A continuación, exportamos el archivo csv que contiene los valores del pseudo UCS estimado en el área de perforación para los pozos perforados como se muestra a continuación.

```
;Hole;Este;Norte;Elev. Collar;UCS [MPa];Profundidad (Mts)
0;P101;30790.61;80891.99;4075.8;3.85;17.83
1;P102;30792.81;80898.98;4076.31;30.68;17.59
```

2;P103;30792.07;80908.19;4076.62;36.54;17.77
3;P104;30797.11;80913.66;4076.29;70.78;17.1
4;P105;30801.33;80921.92;4076.03;105.96;17.77
5;P106;30804.57;80929.37;4075.85;191.17;17.59
6;P201;30798.42;80888.82;4075.7;11.24;17.53
7;P202;30801.12;80896.19;4075.74;37.6;17.68
8;P203;30804.48;80903.67;4075.98;58.4;17.62
9;P204;30806.43;80910.53;4075.76;157.45;17.77
10;P205;30810.12;80918.83;4075.7;183.78;17.53
11;P301;30806.16;80886.07;4075.8;35.68;17.77
12;P302;30809.01;80893.0;4075.78;47.49;17.59
13;P303;30811.66;80901.06;4075.91;113.7;17.68
14;P304;30814.8;80908.22;4075.79;178.28;16.03
15;P305;30817.9;80916.42;4075.52;188.86;17.34
16;P401;30812.75;80883.42;4075.45;2.61;17.68
17;P402;30816.31;80890.1;4075.59;75.62;17.59
18;P403;30819.13;80897.94;4075.96;208.67;17.59
19;P404;30822.32;80905.97;4075.81;168.44;17.68
20;P405;30824.95;80912.63;4075.85;119.15;18.01
21;P501;30821.08;80880.09;4075.27;38.82;17.59
22;P502;30824.27;80887.38;4075.41;178.97;17.59
23;P503;30826.74;80895.07;4075.78;164.96;17.77
24;P504;30829.65;80902.54;4075.87;142.39;15.97
25;P601;30828.44;80877.47;4075.06;63.96;17.34
26;P602;30831.28;80884.55;4075.44;121.54;17.77
27;P603;30834.37;80892.12;4075.43;136.08;18.01
28;P604;30837.46;80899.46;4075.78;112.8;17.77
29;R708;30825.64;80872.53;4075.89;43.52;16.03
30;B7;30808.58;80871.04;4075.45;6.89;16.72
31;B8;30812.74;80869.26;4075.37;6.89;17.05

De este archivo csv se extraerá el valor del pseudo más las coordenadas correspondientes de cada pozo.

A continuación, procedemos a extraer los datos de fragmentación y carga para esto debemos de generar la curva ROM de fragmentación y extraer el csv que se muestra a continuación.

```
;Hole;Norte;Este;x95 [cm];x80 [cm];x60 [cm];x50 [cm];Carga [Kg];Carga [m]
0;P101;80891.99;30790.61;19.66;10.46;5.9;4.444;972.0;9.83
1;P102;80898.98;30792.81;17.42;9.77;5.79;4.462;1047.0;10.59
2;P103;80908.19;30792.07;17.21;9.69;5.76;4.448;1065.0;10.77
3;P104;80913.66;30797.11;18.03;10.0;5.86;4.498;998.0;10.1
4;P105;80921.92;30801.33;17.21;9.69;5.76;4.448;1065.0;10.77
5;P106;80929.37;30804.57;17.41;9.77;5.78;4.461;1047.0;10.59
6;P201;80888.82;30798.42;18.72;10.19;5.87;4.467;991.0;10.03
7;P202;80896.19;30801.12;17.08;9.66;5.76;4.455;1056.0;10.68
8;P203;80903.67;30804.48;17.35;9.75;5.78;4.459;1050.0;10.62
9;P204;80910.53;30806.43;17.15;9.67;5.75;4.448;1065.0;10.77
10;P205;80918.83;30810.12;18.55;10.13;5.86;4.465;1041.0;10.53
11;P301;80886.07;30806.16;17.8;9.88;5.8;4.448;1065.0;10.77
12;P302;80893.0;30809.01;17.05;9.65;5.76;4.462;1047.0;10.59
13;P303;80901.06;30811.66;16.91;9.6;5.75;4.455;1056.0;10.68
14;P304;80908.22;30814.8;19.31;10.48;6.03;4.579;893.0;9.03
15;P305;80916.42;30817.9;17.72;9.88;5.82;4.48;1022.0;10.34
16;P401;80883.42;30812.75;17.49;9.79;5.78;4.455;1056.0;10.68
17;P402;80890.1;30816.31;17.03;9.65;5.76;4.462;1047.0;10.59
18;P403;80897.94;30819.13;17.16;9.69;5.77;4.461;1047.0;10.59
19;P404;80905.97;30822.32;17.31;9.73;5.77;4.454;1056.0;10.68
20;P405;80912.63;30824.95;16.94;9.59;5.72;4.43;1088.0;11.01
21;P501;80880.09;30821.08;17.54;9.81;5.79;4.462;1047.0;10.59
22;P502;80887.38;30824.27;17.0;9.64;5.76;4.461;1047.0;10.59
23;P503;80895.07;30826.74;16.86;9.58;5.74;4.448;1065.0;10.77
24;P504;80902.54;30829.65;20.64;10.9;6.11;4.583;887.0;8.97
25;P601;80877.47;30828.44;17.72;9.89;5.82;4.48;1022.0;10.34
26;P602;80884.55;30831.28;17.21;9.69;5.76;4.448;1065.0;10.77
27;P603;80892.12;30834.37;16.94;9.59;5.72;4.43;1088.0;11.01
28;P604;80899.46;30837.46;17.21;9.69;5.76;4.448;1065.0;10.77
29;R708;80872.53;30825.64;19.63;10.58;6.05;4.58;893.0;9.03
```

30;B7;80871.04;30808.58;18.55;10.2;5.93;4.5280;961.0;9.72
 31;B8;80869.26;30812.74;19.46;10.46;5.95;4.503;944.0;9.55

De este archivo csv extraemos los datos de fragmentación x95 [cm]; x80 [cm];x60 [cm];x50 [cm], el valor de la carga para cada pozo y sus coordenadas para asociarlas a cada pozo.

Finalmente extraemos el archivo csv que contiene el dato target que es el factor de carga como se muestra a continuación.

```
;Hole;Norte;Este;Factor de Carga [g/ton];Carga [Kg];Carga [m]
0;P101;80891.99;30790.61;315.0;972.0;9.83
1;P102;80898.98;30792.81;329.0;1047.0;10.59
2;P103;80908.19;30792.07;325.0;1065.0;10.77
3;P104;80913.66;30797.11;311.0;998.0;10.1
4;P105;80921.92;30801.33;338.0;1065.0;10.77
5;P106;80929.37;30804.57;336.0;1047.0;10.59
6;P201;80888.82;30798.42;324.0;991.0;10.03
7;P202;80896.19;30801.12;375.0;1056.0;10.68
8;P203;80903.67;30804.48;338.0;1050.0;10.62
9;P204;80910.53;30806.43;350.0;1065.0;10.77
10;P205;80918.83;30810.12;234.0;1041.0;10.53
11;P301;80886.07;30806.16;274.0;1065.0;10.77
12;P302;80893.0;30809.01;392.0;1047.0;10.59
13;P303;80901.06;30811.66;399.0;1056.0;10.68
14;P304;80908.22;30814.8;322.0;893.0;9.03
15;P305;80916.42;30817.9;335.0;1022.0;10.34
16;P401;80883.42;30812.75;320.0;1056.0;10.68
17;P402;80890.1;30816.31;399.0;1047.0;10.59
18;P403;80897.94;30819.13;368.0;1047.0;10.59
19;P404;80905.97;30822.32;336.0;1056.0;10.68
20;P405;80912.63;30824.95;345.0;1088.0;11.01
21;P501;80880.09;30821.08;327.0;1047.0;10.59
```

22;P502;80887.38;30824.27;409.0;1047.0;10.59
23;P503;80895.07;30826.74;395.0;1065.0;10.77
24;P504;80902.54;30829.65;214.0;887.0;8.97
25;P601;80877.47;30828.44;340.0;1022.0;10.34
26;P602;80884.55;30831.28;346.0;1065.0;10.77
27;P603;80892.12;30834.37;355.0;1088.0;11.01
28;P604;80899.46;30837.46;339.0;1065.0;10.77
29;R708;80872.53;30825.64;282.0;893.0;9.03
30;B7;80871.04;30808.58;316.0;961.0;9.72
31;B8;80869.26;30812.74;314.0;944.0;9.55

De este último archivo csv se extrae el valor del factor de carga para cada pozo, incluyendo sus coordenadas asociadas.

De este modo, a partir del Gblast se obtuvieron los archivos csv que contienen los parámetros de entrada (Pseudo UCS, Carga, Fragmentación en X50, X60, X90) y el parámetro de salida (Factor de Carga) para el modelo.

Este procedimiento se repetirá para todas las mallas usadas en el modelo de machine learning a fin de obtener los parámetros de entrada y la suficiente data para poder entrenar el modelo.

3.1.3 CARGA DE DATOS

Es necesario conocer cómo cargar los datos, antes de usarla para un entrenamiento de un modelo de machine learning. Un formato estándar para las bases de datos es aquel con valores separados por comas (.csv).

El formato estándar de archivo para pequeños datasets es de valores separados por comas o CSV. En su forma más simple los archivos csv están

comprimidos en filas de datos. Cada fila está dividida en columnas usando una (,) como separador.

Para la presente tesis se trabajó con un archivo csv de base de datos estructurada que contiene los datos de los parámetros input y el output para el aprendizaje de la red neuronal, para esto en primer lugar se importa la librería de pandas, usando la función “read_csv” de pandas que toma el nombre del archivo “DTOTALSINREDRILLSq1_w11.csv” este archivo csv se obtiene de la unión de los archivos csv resultados de la concatenación de sus parámetros (input y output) extraídos del Gblast para cada malla como se señaló en el punto anterior. Finalmente, la función read_csv mostrará el csv sobre el que se trabajará el modelo, además se usa la función “usecols” para solo seleccionar las columnas del 2 al 9 y no la primera columna que es la columna ID de cada pozo que no entra dentro del modelo de predicción. En la parte inferior se observa el código para la carga de la base de datos.

```
import pandas as pd
```

```
data= pd.read_csv('DTOTALSINREDRILLq1_w11.csv', usecols =  
[2,4,5,6,7,8,9])
```

```
data
```

Para esto primero se tuvo que extraer distintos csv que contienen los parámetros necesarios para realizar el modelo estos csv provienen del programa Gblast y son los siguientes: csv de UCS, csv de Factor de carga y csv de ROM (fragmentación). Luego se unieron las columnas que contenían los parámetros de importancia para el modelo usando código python.

Tabla 3. 2. Dataset final generado a partir de unir los csv exportado desde el programa Gblast.

	UCS [MPa]	x95[cm]	x80[cm]	x60[cm]	x50[cm]	Carga[Kg]	Factor de Carga [g/ton]
0	3.85	19.66	10.46	5.90	4.444	972.0	315.0
1	30.68	17.42	9.77	5.79	4.462	1047.0	329.0
2	36.54	17.21	9.69	5.76	4.448	1065.0	325.0
3	70.78	18.03	10.00	5.86	4.498	998.0	311.0
4	105.96	17.21	9.69	5.76	4.448	1065.0	338.0
...
1089	146.06	17.50	9.80	5.80	4.466	1041.0	356.0
1090	84.79	17.12	9.66	5.75	4.441	1074.0	364.0
1091	145.18	17.32	9.73	5.77	4.454	1056.0	361.0
1092	192.19	17.62	9.85	5.81	4.473	1031.0	348.0
1093	155.16	17.62	9.85	5.81	4.473	1031.0	349.0

Fuente: Elaboración propia.

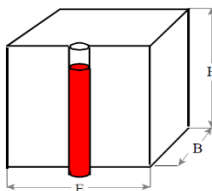
3.2 LIMPIEZA DE DATOS

La limpieza de datos es una parte esencial y muy importante al momento de desarrollar un programa de machine learning ya que si no se realiza generará error en las predicciones.

3.2.1 FACTOR DE CARGA

El factor de carga es nuestro target o variable objetivo cuyos datos para entrenamiento lo hemos obtenido a través del Gblast, el factor de carga es la relación entre el peso del explosivo y el peso de la roca, es conveniente y fácil de calcular, su importancia recae en que en su fórmula contiene el valor

numérico de la multiplicación del valor del burden por el valor del espaciamiento tal como se muestra en la siguiente fórmula:



$$FC(g/ton) = \frac{W_{exp}}{B * E * H * \delta_{roca}}$$

Donde

FC = factor de carga (g/ton)

W = peso del explosivo (gramos)

B = burden (metros)

E = espaciamiento (metros)

δ = densidad de la roca (toneladas/m³)

Cada hole tiene un factor de carga asociado, del cual obtendremos el valor del burden y espaciamiento para cada hole una vez se haya entrenado el modelo de redes neuronales.

¿Cuál es el factor de carga más adecuado?, ello depende de la definición de adecuado, que dependerá del objetivo entonces podrá ser lo que dicen los investigadores, lo que pide operación mina (pala, cargador, camión), lo que piden los procesos de aguas abajo(chancadores, molinos SAG, molino de

bolas, pila de lixiviado), lo apropiado para los botaderos(estéril o lastre), lo que dice el presupuesto o lo que justifica el negocio; por este motivo es que usar la red neuronal nos da una gran ventaja ya que da la flexibilidad de obtener el factor de carga adecuado para cada caso mencionado según el interés que se tenga.

Los datos se cargan desde los csv de factor de carga que genera el GBlast para todas las mallas de voladura usadas.

Este parámetro se concatena con los otros a partir de la librería de pandas para crear un solo csv con todos los parámetros que se usarán en el modelo.

A continuación, se muestra el código usado para cargar todos los datasets del factor de carga:

```
import pandas as pd
```

```
facarq1 =pd.read_csv('Factor_de_cargaq1.csv', names = None, usecols = [1,2,3,4])
facarq2 =pd.read_csv('Factor_de_cargaq2.csv', names = None, usecols = [1,2,3,4])
facarq4 =pd.read_csv('Factor_de_cargaq4.csv', names = None, usecols = [1,2,3,4])
facarq6 =pd.read_csv('Factor_de_cargaq6.csv', names = None, usecols = [1,2,3,4])
facarq7 =pd.read_csv('Factor_de_cargaq7.csv', names = None, usecols = [1,2,3,4])
facarq8 =pd.read_csv('Factor_de_cargaq8.csv', names = None, usecols = [1,2,3,4])
facarq9=pd.read_csv('Factor_de_cargaq9.csv', names = None, usecols = [1,2,3,4])
facarr1 =pd.read_csv('Factor_de_cargar1.csv', names = None, usecols = [1,2,3,4])
facarw0 =pd.read_csv('Factor_de_cargaw0.csv', names = None, usecols = [1,2,3,4])
facarw1 =pd.read_csv('Factor_de_cargaw1.csv', names = None, usecols = [1,2,3,4])
facarw2 =pd.read_csv('Factor_de_cargaw2.csv', names = None, usecols = [1,2,3,4])
facarw4 =pd.read_csv('Factor_de_cargaw4.csv', names = None, usecols = [1,2,3,4])
facarw5 =pd.read_csv('Factor_de_cargaw5.csv', names = None, usecols = [1,2,3,4])
facarw6 =pd.read_csv('Factor_de_cargaw6.csv', names = None, usecols = [1,2,3,4])
facarw7 =pd.read_csv('Factor_de_cargaw7.csv', names = None, usecols = [1,2,3,4])
facarw10=pd.read_csv('Factor_de_cargaw10.csv', names = None, usecols = [1,2,3,4])
facarw11=pd.read_csv('Factor_de_cargaw11.csv', names = None, usecols = [1,2,3,4])
```

A continuación, se procede a concatenar todos los datasets:

```
facarq1_w11=pd.concat([facarq1,facarq2,facarq4,facarq6,facarq7,facarq8,facarq9,
facarr1,facarw0,facarw1,facarw2,
facarw5,facarw6,facarw7,facarw10,facarw11], axis = 0)
```

Al imprimir el dataset encontramos que nos muestra los datos ya concatenados.

Tabla 3. 3. Dataset concatenado del factor de carga.

	Hole	Norte	Este	Factor de Carga [g/ton]
0	P101	80891.99	30790.61	315.0
1	P102	80898.98	30792.81	329.0
2	P103	80908.19	30792.07	325.0
3	P104	80913.66	30797.11	311.0
4	P105	80921.92	30801.33	338.0
...
145	P814	81015.11	30813.64	356.0
146	P815	81023.63	30813.24	364.0
147	P816	81032.41	30813.20	361.0
148	P817	81040.48	30812.68	348.0
149	P818	81049.99	30812.62	349.0

1152 rows × 4 columns

Fuente: Elaboración propia.

Finalmente se exporta la base de datos en formato csv que contiene la concatenación de todas las bases de datos para cada malla, extrayendo así la variable factor de carga de todos los holes.

Para este fin usamos el siguiente código que guardará el archivo en el ordenador para usarlo más adelante, este procedimiento se repetirá para la extracción de las demás variables que entran en el modelo:

```
facarq1_w11.to_csv (r'C:\Users\user\Desktop\facarq1_w11.csv', index =  
False, header=True)
```

3.2.2 IDs

ID se refiere a la identificación o al nombre que lleva cada hole (pozo y/o taladro) dentro de una malla de voladura, los cuales son generados al momento de diseñar las mallas de voladura y que en ocasiones suelen repetirse al ser un número, por ejemplo "hole de ID705". Además cada hole es único y tiene una ubicación propia en el espacio (x,y,z) sin importar si el ID se repita dentro de una misma malla de voladura.

Dentro de la base de datos tenemos holes de distintas mallas de voladura, por este motivo encontramos IDs repetidos en la variable holes al hacer la concatenación. Para poder diferenciar estos IDs esto se ha automatizado la adición de un diferenciador en los IDs repetidos, este proceso se hizo mediante un algoritmo que detecta automáticamente los IDs repetidos y añade el diferenciador.

A continuación, el procedimiento, tomemos por ejemplo el dataset concatenado de la variable fragmentación:

```
import pandas as pd  
Ye = "fragq1_w11.csv"
```

```
data = pd.read_csv(Ye, names = None, usecols =
[0,1,2,3,4,5,6,7])
data
```

Observamos la base de datos de la concatenación que contiene la variable fragmentación en X95, X80, X60, X50, y la variable carga que también es uno de los inputs del modelo de redes neuronales.

Tabla 3. 4. Dataset concatenado de la carga de explosivo.

	Hole	Norte	Este	x95[cm]	x80[cm]	x60[cm]	x50[cm]	Carga[Kg]
0	P101	80891.99	30790.61	19.66	10.46	5.90	4.444	972.0
1	P102	80898.98	30792.81	17.42	9.77	5.79	4.462	1047.0
2	P103	80908.19	30792.07	17.21	9.69	5.76	4.448	1065.0
3	P104	80913.66	30797.11	18.03	10.00	5.86	4.498	998.0
4	P105	80921.92	30801.33	17.21	9.69	5.76	4.448	1065.0
...
1147	P814	81015.11	30813.64	17.50	9.80	5.80	4.466	1041.0
1148	P815	81023.63	30813.24	17.12	9.66	5.75	4.441	1074.0
1149	P816	81032.41	30813.20	17.32	9.73	5.77	4.454	1056.0
1150	P817	81040.48	30812.68	17.62	9.85	5.81	4.473	1031.0
1151	P818	81049.99	30812.62	17.62	9.85	5.81	4.473	1031.0

Fuente: Elaboración propia.

Convertimos nuestra variable 'Hole' en datos tipo string para poder hacer la codificación.

```
data.Hole.astype("str")
```

Ahora que se encuentra en formato string, convertimos la columna hole en el índice de nuestra base de datos.

```
data.set_index('Hole',inplace=True)
data
```

La base de datos es transformada, a continuación, el resultado, podemos observar que la columna 'Hole' ahora es el índice de la base de datos:

Tabla 3. 5. Transformación a índices de la variable "Hole".

	Norte	Este	x95[cm]	x80[cm]	x60[cm]	x50[cm]	Carga[Kg]
Hole							
P101	80891.99	30790.61	19.66	10.46	5.90	4.444	972.0
P102	80898.98	30792.81	17.42	9.77	5.79	4.462	1047.0
P103	80908.19	30792.07	17.21	9.69	5.76	4.448	1065.0
P104	80913.66	30797.11	18.03	10.00	5.86	4.498	998.0
P105	80921.92	30801.33	17.21	9.69	5.76	4.448	1065.0
...
P814	81015.11	30813.64	17.50	9.80	5.80	4.466	1041.0
P815	81023.63	30813.24	17.12	9.66	5.75	4.441	1074.0
P816	81032.41	30813.20	17.32	9.73	5.77	4.454	1056.0
P817	81040.48	30812.68	17.62	9.85	5.81	4.473	1031.0
P818	81049.99	30812.62	17.62	9.85	5.81	4.473	1031.0

1152 rows × 7 columns

Fuente: Elaboración propia.

A continuación, se realizó una transposición de intercambio de filas por columnas, a través del siguiente código:

```
data1 = data.transpose()
data1
```

Obteniendo como resultado la base de datos que se muestra a continuación:

Tabla 3. 6. Transpuesta de filas por columnas en la base de datos.

Hole	P101	P102	P103	P104	P105	P106	P201	P202	P203	P204	...	P714	P715	P716
Norte	80891.990	80898.980	80908.190	80913.660	80921.920	80929.370	80888.820	80896.190	80903.670	80910.530	...	81011.930	81020.870	81028.650
Este	30790.610	30792.810	30792.070	30797.110	30801.330	30804.570	30798.420	30801.120	30804.480	30806.430	...	30820.870	30821.650	30821.840
x95[cm]	19.660	17.420	17.210	18.030	17.210	17.410	18.720	17.080	17.350	17.150	...	18.030	17.860	17.870
x80[cm]	10.460	9.770	9.690	10.000	9.690	9.770	10.190	9.660	9.750	9.670	...	10.000	9.950	9.950
x60[cm]	5.900	5.790	5.760	5.860	5.760	5.780	5.870	5.760	5.780	5.750	...	5.860	5.860	5.860
x50[cm]	4.444	4.462	4.448	4.498	4.448	4.461	4.467	4.455	4.459	4.448	...	4.496	4.503	4.503
Carga[Kg]	972.000	1047.000	1065.000	998.000	1065.000	1047.000	991.000	1056.000	1050.000	1065.000	...	1001.000	992.000	992.000

7 rows × 1152 columns

Fuente: Elaboración propia.

A continuación, ya teniendo la etiqueta de cada columna como el dato string de lo que inicialmente era la columna 'Hole' procedemos a hacer la adición del diferenciador para todas las columnas con el mismo nombre, usando el siguiente código:

```
def _maybe_dedup_names(self, names):
```

```

if self.mangle_dupe_cols:
    names = list(names)
    counts = {}

    for i, col in enumerate(names):
        cur_count = counts.get(col, 0)

        if cur_count > 0:
            names[i] = '%s.%d' % (col, cur_count)

        counts[col] = cur_count + 1

    return names

```

Para finalmente obtener los códigos diferenciados de la columna 'Hole':

```

Nuevo_Hole=pd.io.parsers.ParserBase({'names':data1.columns})._maybe_de
dup_names(data1.columns)

```

Entonces creamos un archivo pandas que contendrá los valores de 'Nuevo_Hole' como columna.

```

nuevo = pd.DataFrame(Nuevo_Hole)
nuevo.columns = ['HOLE']

```

Imprimimos el dataframe 'nuevo', y como se observa ahora cada valor repetido tiene un diferenciador:

Tabla 3. 7. Columna "Hole" con transformación de códigos repetidos.

HOLE	
0	P101
1	P102
2	P103
3	P104
4	P105
...	...
1147	P814.1
1148	P815.1
1149	P816.1
1150	P817.1
1151	P818

Fuente: Elaboración propia.

Procedemos a añadir esta nueva columna 'HOLE' en el dataframe original y reemplazarlo por la antigua columna 'HOLE'.

Primero eliminamos la columna 'HOLE' en la base de datos original.

```
Ye = "fragq1_w11.csv"
data = pd.read_csv(Ye, names = None, usecols = [0,1,2,3,4,5,6,7])
del data['Hole']
data
```

Luego definimos la nueva columna 'Hole':

```
data['Hole'] = nuevo
data
```

Obteniendo finalmente el dataset con los IDs transformados.

Tabla 3. 8. Base de datos resultante con columna “Hole” transformada para códigos repetidos.

	Norte	Este	x95[cm]	x80[cm]	x60[cm]	x50[cm]	Carga[Kg]	Hole
0	80891.99	30790.61	19.66	10.46	5.90	4.444	972.0	P101
1	80898.98	30792.81	17.42	9.77	5.79	4.462	1047.0	P102
2	80908.19	30792.07	17.21	9.69	5.76	4.448	1065.0	P103
3	80913.66	30797.11	18.03	10.00	5.86	4.498	998.0	P104
4	80921.92	30801.33	17.21	9.69	5.76	4.448	1065.0	P105
...
1147	81015.11	30813.64	17.50	9.80	5.80	4.466	1041.0	P814.1
1148	81023.63	30813.24	17.12	9.66	5.75	4.441	1074.0	P815.1
1149	81032.41	30813.20	17.32	9.73	5.77	4.454	1056.0	P816.1
1150	81040.48	30812.68	17.62	9.85	5.81	4.473	1031.0	P817.1
1151	81049.99	30812.62	17.62	9.85	5.81	4.473	1031.0	P818

1152 rows × 8 columns

Fuente: Elaboración propia.

Exportamos y guardamos en el ordenador el dataset ‘data’ con el nombre ‘fragq1_w11’ indicando que contiene los datos de fragmentación usando el siguiente código.

```
data.to_csv(r'C:\Users\user\Desktop\fragq1_w11.csv', index = False,
header=True)
```

Adicionalmente como se mencionó anteriormente, los csv que se generaron por concatenación tienen ciertos parámetros de interés que serán los inputs de la red neuronal artificial por lo que para unir los csv concatenados deben tener una columna común que será la relacionante entre estos, esta columna será la columna 'HOLE' por lo que se repetirá el proceso de añadir la columna HOLE en los otros csv generados por concatenación como por ejemplo el csv 'facarq1_w11' que contiene la variable factor de carga.

3.2.3 BURDEN Y ESPACIAMIENTO

El Burden y espaciamento son dos variables de suma importancia al momento de hacer el diseño de la malla de perforación, estas nos permiten saber la ubicación donde se perforan los holes sin embargo en muchas ocasiones el diseño inicial se modifica debido a las condiciones operativas. El objetivo de este programa de machine learning es obtener el burden y espaciamento adecuado donde se perforarán los holes usando en cada uno de estos su taco crítico aprovechando así al máximo cada metro perforado. Sin embargo, será más práctico predecir el parámetro factor de carga como variable objetivo y que esta permita obtener los valores del burden y espaciamento.

3.2.4 DRILL Y REDRILLS

Solo trabajaremos con los holes tipo drill para mantener el diseño de voladura lo más cercano al diseño planificado y no generar anomalías en la distribución de holes que generen una mayor complejidad al modelo de aprendizaje automático por considerar los redrills.

3.2.5 UCS

El UCS o resistencia a la compresión uniaxial es una medida de la resistencia de un material. La resistencia a la compresión uniaxial (UCS) es el esfuerzo de compresión axial máximo que puede tolerar una muestra cilíndrica recta de material antes de fracturarse. El UCS será la variable tomada para representar la respuesta mecánica del terreno a la voladura.

El valor de esta variable se ha obtenido a partir del resultado generado por el programa GBlast que se obtiene de la respuesta mecánica de la roca detectado por los sensores de los taladros, en adición se realizó una limpieza de datos para hacer un control de calidad de calibrar los valores obtenidos de la variable UCS.

Por último, al igual que con las otras variables de entrada del modelo de red neuronal se realizó una concatenación de las bases de datos en formato csv que contienen la variable UCS.

Esta variable es muy importante debido a que en la detonación la roca circundante inmediata se tritura cuando se excede la resistencia a la compresión de la roca, por lo que es importante conocer este valor (UCS). Más allá de la zona aplastada, las rocas circundantes desarrollan grietas radiales a medida que la intensidad de la onda de choque excede la resistencia a la tracción de la roca y la presión del gas resultante viaja a través de las grietas extendiéndose más (Tiile, 2016), generando así la fragmentación resultante de la voladura.

Para cargar las bases de datos que contienen el parámetro UCS de las distintas mallas se realizó el siguiente código que permite cargar los archivos en formato csv de cada malla.

```

import pandas as pd
ucsq1 =pd.read_csv('UCSq1.csv', names = None, usecols = [1,2,3,5])
ucsq2 =pd.read_csv('UCSq2.csv', names = None, usecols = [1,2,3,5])
ucsq4 =pd.read_csv('UCSq4.csv', names = None, usecols = [1,2,3,5])
ucsq6 =pd.read_csv('UCSq6.csv', names = None, usecols = [1,2,3,5])
ucsq7 =pd.read_csv('UCSq7.csv', names = None, usecols = [1,2,3,5])
ucsq8 =pd.read_csv('UCSq8.csv', names = None, usecols = [1,2,3,5])
ucsq9 =pd.read_csv('UCSq9.csv', names = None, usecols = [1,2,3,5])
ucsr1 =pd.read_csv('UCSr1.csv', names = None, usecols = [1,2,3,5])
ucsw0 =pd.read_csv('UCSw0.csv', names = None, usecols = [1,2,3,5])
ucsw1 =pd.read_csv('UCSw1.csv', names = None, usecols = [1,2,3,5])
ucsw2 =pd.read_csv('UCSw2.csv', names = None, usecols = [1,2,3,5])
ucsw4 =pd.read_csv('UCSw4.csv', names = None, usecols = [1,2,3,5])
ucsw5 =pd.read_csv('UCSw5.csv', names = None, usecols = [1,2,3,5])
ucsw6 =pd.read_csv('UCSw6.csv', names = None, usecols = [1,2,3,5])
ucsw7 =pd.read_csv('UCSw7.csv', names = None, usecols = [1,2,3,5])
ucsw10 =pd.read_csv('UCSw10.csv', names = None, usecols = [1,2,3,5])
ucsw11 =pd.read_csv('UCSw11.csv', names = None, usecols = [1,2,3,5])

```

El objetivo de trabajar con distintas mallas es generar un aprendizaje de la respuesta del macizo rocoso a la voladura para distintas condiciones del terreno.

Posterior a la carga de las bases de datos en formato csv se procedió a realizar la concatenación de las bases de datos como se muestra a continuación:

```

ucsq1_w11=pd.concat([ucsq1,ucsq2,ucsq4,ucsq6,ucsq7,ucsq8,ucsq9,ucsr1,
ucsw0,ucsw1, ucsw2, ucsw4,ucsw5,ucsw6,ucsw7,ucsw10,ucsw11], axis = 0)

```

Obteniendo como resultado la base de datos que contiene la variable UCS

Tabla 3. 9. Base de datos que contiene la variable UCS (1152 filas).

	Hole	Este	Norte	UCS [MPa]
0	P101	30790.61	80891.99	3.85
1	P102	30792.81	80898.98	30.68
2	P103	30792.07	80908.19	36.54
3	P104	30797.11	80913.66	70.78
4	P105	30801.33	80921.92	105.96
...
145	P814	30813.64	81015.11	146.06
146	P815	30813.24	81023.63	84.79
147	P816	30813.20	81032.41	145.18
148	P817	30812.68	81040.48	192.19
149	P818	30812.62	81049.99	155.16

1152 rows × 4 columns

Fuente: Elaboración propia.

Finalmente exportamos esta base de datos que contiene la variable UCS de los puntos de ubicación de los holes, como se muestra a continuación.

```
ucsq1_w11.to_csv (r'C:\Users\user\Desktop\ucsq1_w11.csv', index = False,
header=True)
```

Esta base de datos se unirá a las otras que contienen las demás variables necesarias para el modelo de aprendizaje automático.

3.3 ANÁLISIS DE DATOS

El primer paso para realizar un programa de machine learning es el análisis de datos que permite analizar características de nuestros datos como el tipo de variable, si es que son numéricas o categóricas, el tipo de distribución de los datos, si poseen una distribución normal o si posee sesgo, la dimensión de las variables, la presencia de valores Nan, son algunas características a considerar ya que estas influyen en el modelo de aprendizaje automático que se usará.

Para empezar el análisis de datos previamente se realizó la unión de todas las bases de datos mencionadas anteriormente donde cada base de datos contiene variables de interés para el modelo de red neuronal son las siguientes: UCS[MPa], x95[cm], x80[cm], x60[cm], x50[cm], Carga[Kg], Factor de Carga [g/ton]

3.3.1 ESTADÍSTICA DESCRIPTIVA

La estadística descriptiva de los datos nos permite conocer el comportamiento estadístico de los datos como la dimensión del dataset, los tipos de datos, y la estadística descriptiva propiamente dicha.

3.3.1.1 DIMENSIÓN DEL DATASET

La dimensión del dataset es importante para tener un control de la cantidad de instancias debiendo ser éstas mucho mayor a la cantidad de columnas (cada columna es una variable del modelo) de lo contrario el aprendizaje podría ser muy pobre y la predicción tendrá un subajuste lejano del valor real, existen herramientas que se pueden utilizar para eliminar columnas en función de la importancia que aportan a la predicción del target, de las

variables así mismo tener la suficiente cantidad de instancias o filas para que el modelo pueda aprender. Para esto usamos la función “.shape” que mostrará la cantidad de filas y columnas de la base de datos tal como se muestra en el código a continuación.

```
import pandas as pd
data = pd.read_csv('DTOTALSINREDRILLq1_w11.csv', usecols =
[2,4,5,6,7,8,9])
data.shape
```

Que retorna el valor de la dimensión de la base de datos 1085 filas y 7 columnas como se muestra a continuación.

```
(1085, 7)
```

3.3.1.2 TIPOS DE DATOS

Podemos usar las funciones de python para saber si tenemos datos tipo numérico o categórico, esto es importante para el modelo porque las variables inputs deben ser valores numéricos para introducirlos en el modelo en caso fuera una variable categórica se realiza una conversión del parámetro categórico a numérico usando la herramienta de one hot encoding.

Usamos la función “.info” para mostrar el tipo de variable que es cada parámetro (inputs y output) como se muestra a continuación

```
data.info()
```

Y obtenemos:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1085 entries, 0 to 1084
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   UCS [MPa]             1085 non-null   float64
1   x95[cm]              1085 non-null   float64
2   x80[cm]              1085 non-null   float64
3   x60[cm]              1085 non-null   float64
4   x50[cm]              1085 non-null   float64
5   Carga[Kg]            1085 non-null   float64
6   Factor de Carga [g/ton] 1085 non-null   float64
dtypes: float64(7)
```

Los tipos de variable de cada parámetro en este caso son todos son variables tipo float que indican valores numéricos decimales, además se observa que poseemos 1085 datos de cada variable sin valores nulos en la data.

3.3.1.3 ESTADÍSTICA DESCRIPTIVA DE LOS DATOS

Podemos observar la cantidad de filas, los valores de la media, la mediana, la desviación estándar, el valor mínimo, el cuartil de 25%, cuartil de 75% y valor máximo de cada parámetro usando la función “.describe” `data.describe()`

Tabla 3. 10. Tabla de estadística descriptiva de las variables involucradas en modelo de red neuronal artificial.

	UCS [MPa]	x95[cm]	x80[cm]	x60[cm]	x50[cm]	Carga[Kg]	Factor de Carga [g/ton]
count	1094.000000	1094.000000	1094.000000	1094.000000	1094.000000	1094.000000	1094.000000
mean	101.641152	66.598528	12.781746	6.079196	4.513512	960.909506	332.188300
std	145.836487	940.814903	26.982724	1.327181	0.119634	154.344507	64.362489
min	2.610000	15.790000	9.130000	5.560000	4.348000	70.000000	29.000000
25%	50.085000	17.330000	9.750000	5.780000	4.463000	941.000000	312.000000
50%	89.325000	17.910000	9.960000	5.850000	4.485000	998.000000	336.000000
75%	125.460000	19.065000	10.357500	5.957500	4.522000	1041.000000	369.000000
max	4113.050000	30681.550000	773.890000	27.520000	5.263000	1199.000000	485.000000

Fuente: Elaboración propia.

Esta tabla nos permite tener una idea del comportamiento de nuestra variable sin embargo esta es mejor comprendida en gráficos de visualización

3.3.2 VISUALIZACIÓN DE DATOS

La visualización de los datos nos permite observar mediante gráficos el comportamiento de nuestros datos y de esta manera preparar los datos en base a su comportamiento para ser introducidas al modelo de aprendizaje automático.

3.3.2.1 HISTOGRAMAS

Se generan los histogramas de los parámetros y se observa un fuerte sesgo de los parámetros de entrada hacia el lado izquierdo a excepción de la carga que posee sesgo hacia el lado derecho, este sesgo influye al momento de aplicar

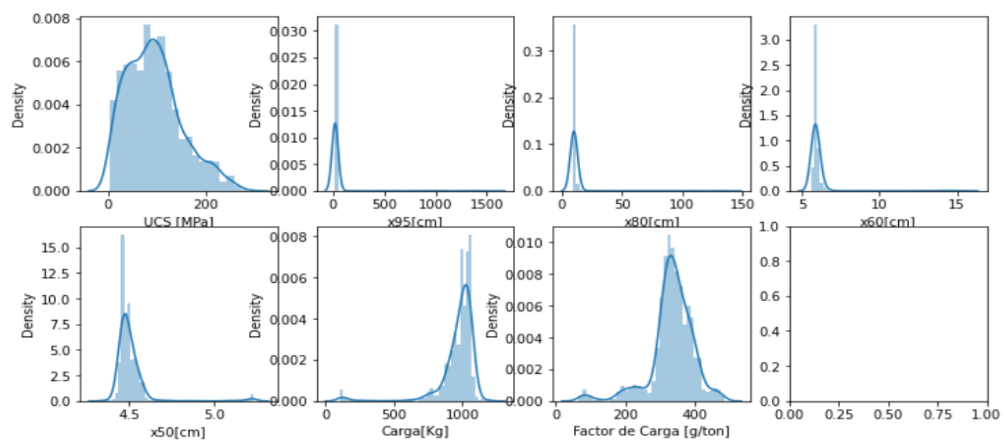
la red neuronal por lo cual tendremos que hacer un tratamiento de datos más adelante para reducir el sesgo, este podría ser el método de Box Cox o el de Yeo Johnson.

Usamos el siguiente código:

```
f, axes = plt.subplots(2,4, figsize = (12,6))
sns.distplot(data["UCS [MPa]"], ax=axes[0,0])
sns.distplot(data["x95[cm]"], ax=axes[0,1])
sns.distplot(data["x80[cm]"], ax=axes[0,2])
sns.distplot(data["x60[cm]"], ax=axes[0,3])
sns.distplot(data["x50[cm]"], ax=axes[1,0])
sns.distplot(data["Carga[Kg]"], ax=axes[1,1])
sns.distplot(data["Factor de Carga [g/ton]"], ax=axes[1,2])
```

Que nos muestra los histogramas para cada parámetro.

Tabla 3. 11. histogramas de las variables usadas en el modelo de aprendizaje automático.



Fuente: Elaboración propia.

Se observan los histogramas de los parámetros de entrada así como el histograma del target, otro conclusión importante que podemos deducir de observar los histogramas además del sesgo que poseen es el hecho que las variables están en distintas dimensiones, este comportamiento también necesitará un procesamiento más adelante antes de introducir las variables al modelo de aprendizaje automático, entre las herramientas disponibles se tiene la normalización, estandarización, binarización y el escalamiento.

3.3.2.2 BOXPLOT

El boxplots o diagrama de caja es un método estandarizado para representar gráficamente una serie de datos numéricos a través de sus cuartiles. De esta manera, el diagrama de caja muestra a simple vista la mediana y los cuartiles de los datos, pudiendo también representar los valores atípicos de estos. Conviene recordar que se utilizan las bisagras de Tukey, y no los cuartiles a la hora de dibujar la caja del gráfico, aunque los resultados son semejantes en muestras grandes.

El diagrama de caja es compuesto de los siguientes elementos:

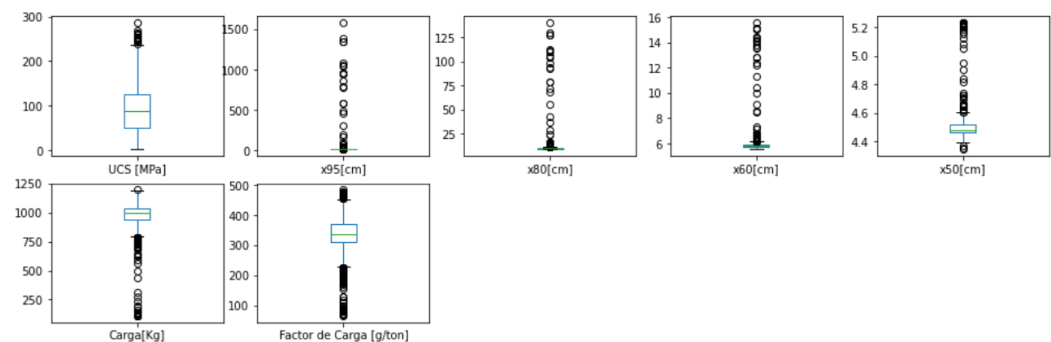
- Rango (sin datos atípicos)
- Datos atípicos.
- Rango intercuartil
- Cuartiles (denotados como Q1, Q2 y Q3)
- Mediana (Q2)
- Mínimo y máximo.

Para mostrar los boxplot de los datos se procede con el siguiente código:

```
import matplotlib.pyplot as plt
%matplotlib inline
fig = plt.figure(figsize=(16,16))
ax = fig.gca()
data.plot(ax=ax, kind='box', subplots=True, layout=(6,5), sharex=False)
plt.show()
```

Se generan los boxplot de los parámetros:

Tabla 3. 12. Boxplot de las variables usadas en el modelo de aprendizaje automático.



Fuente: Elaboración propia.

De los Box Plots para los parámetros del modelo observamos que muchos de estos presentan outliers y el fuerte sesgo que también se visualiza en los histogramas.

3.3.2.3 MATRIZ DE CORRELACIÓN

La matriz de correlación nos permite conocer el grado de correlación entre los parámetros, así también con el target y ver su influencia sobre este target que es el output del programa de machine learning.

En una matriz de correlación los valores de 1 (uno) son los de correlación positiva máxima, y el valor de correlación negativa máxima es de -1, los valores entre 0 y 1 pero con más cercanía a cero son correlación positiva y los valores entre -1 y cero son los de correlación negativa.

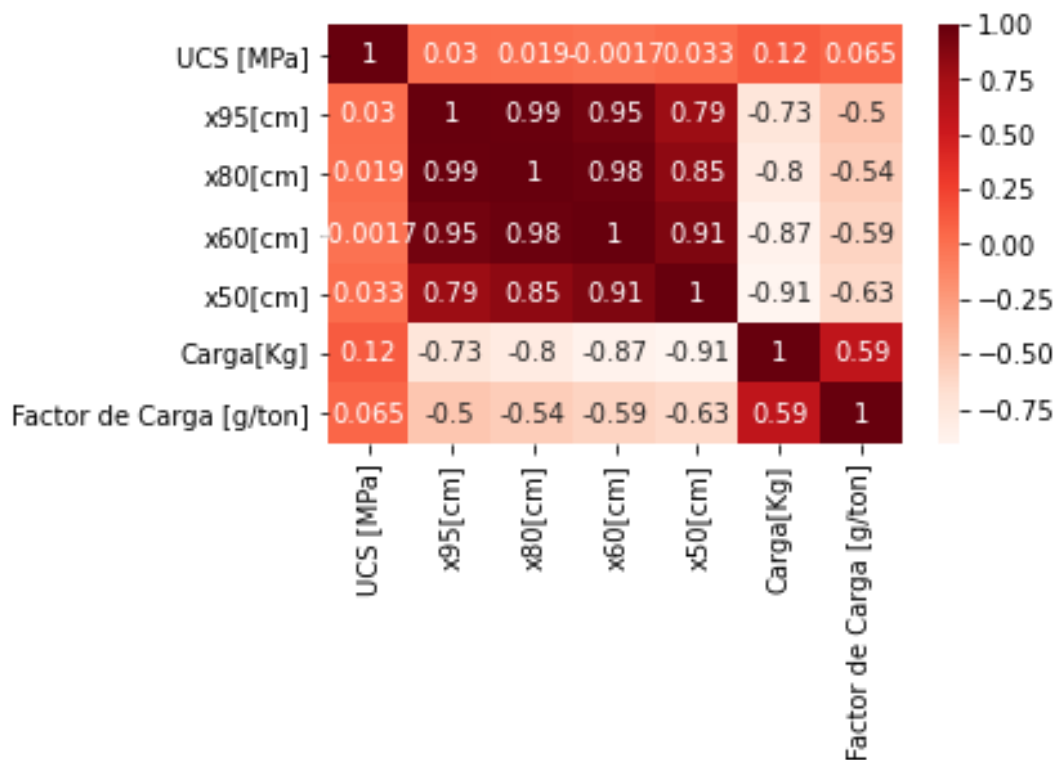
En el gráfico de correlación se observa una alta correlación entre los resultados de fragmentación en x95, x80, x60 y disminuye ligeramente para x50, esto asociado a la función de fragmentación.

También observamos la correlación de los parámetros input con el target (output) siendo el parámetro que tiene mejor correlación con el target el atributo carga. Usamos el siguiente código:

```
#MATRIZ MULTICLASE  
plt.figure(figsize=(5,3))  
corr = data.corr()  
sns.heatmap(corr, annot=True, cmap=plt.cm.Reds)
```

Se genera la matriz de correlación entre los parámetros:

Tabla 3. 13. Diagrama de correlación de las variables usadas en el modelo de aprendizaje automático.



Fuente: Elaboración propia.

En la matriz de correlación de los parámetros del modelo se observa en escala de colores los más oscuros son cercanos a 1 (uno) e indican una correlación positiva muy alta, también se observa una alta correlación negativa entre los resultados de fragmentación y la variable carga lo cual es lógico ya que ha mayor carga de explosivo menor es el tamaño de los fragmentos resultado de la voladura.

3.3.2.4 MATRIZ DE DISPERSIÓN

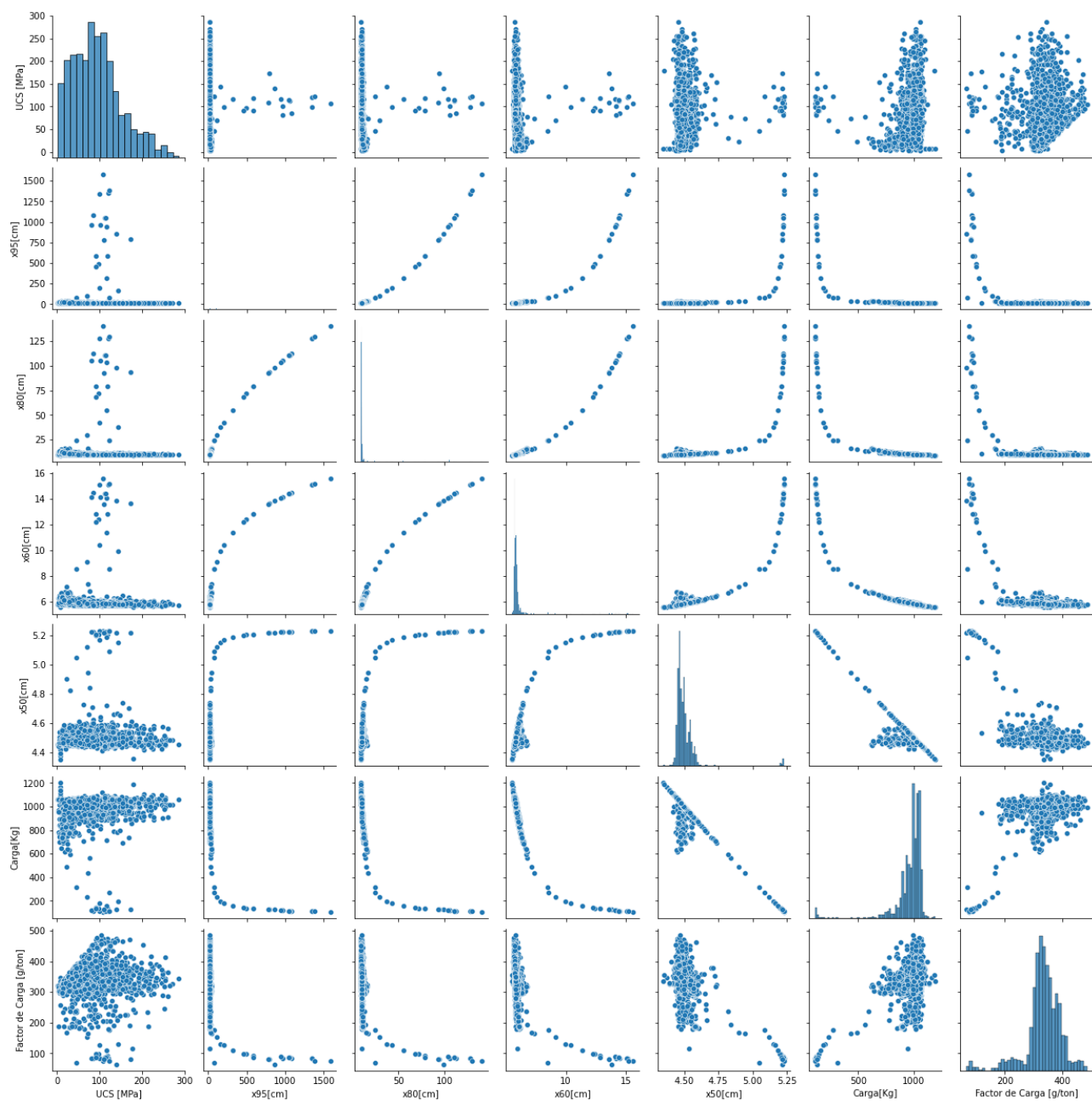
Una matriz de gráficos de dispersión es una cuadrícula de varios gráficos de dispersión de varias variables numéricas. La matriz incluye gráficos de dispersión individuales para cualquier combinación de variables. La matriz de dispersión permite responder a preguntas sobre sus datos, por ejemplo: ¿qué relaciones existen entre varias variables?

Se genera la matriz de dispersión, para este fin usamos el siguiente código:

```
import seaborn as sns  
sns.pairplot(data);
```

Y nos muestra la siguiente matriz de dispersión donde a partir de las cuadrículas podemos ver las correlaciones entre las variables, se observa en la imagen correlaciones positivas bien definidas y de la misma manera también correlaciones negativas, los cuales tienen un sentido lógico al analizar cada una con detalle.

Tabla 3. 14. Diagrama de dispersión de las variables usadas en el modelo de aprendizaje automático.



Fuente: Elaboración propia.

En el análisis de las correlaciones nos podemos dar cuenta que para el caso del UCS su correlación con los parámetros de fragmentación nos indican un tamaño casi uniforme en los resultados de fragmentación, incluso variando el UCS, mientras que en la carga si hay un corto rango de valores pero que de igual manera se mantiene, aunque varíe el UCS.

CAPITULO IV.

TRATAMIENTO DE DATOS

El análisis de los datos nos permitió identificar el comportamiento estadístico que tienen cada uno de nuestros parámetros de manera individual y también en correlación con las otras variables, sabiendo esto podemos elegir las herramientas adecuadas para tratar nuestros datos a manera que estén listos para introducir en el modelo de aprendizaje automático que elijamos en este caso el de redes neuronales.

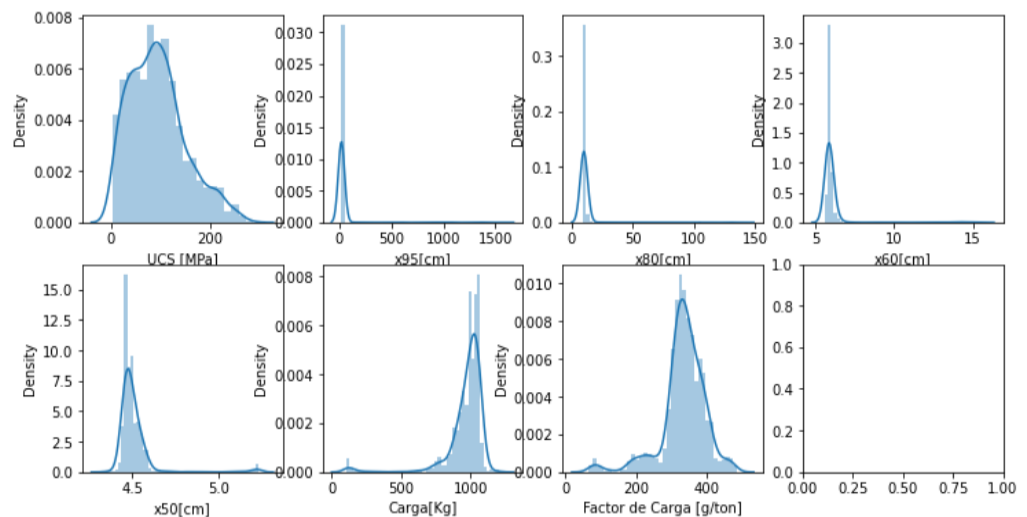
Los tratamientos de datos necesarios que se identificaron para la base de datos que se posee según el comportamiento de los datos fueron una transformación de estos para reducir el sesgo de las variables y una transformación para estandarizar el rango de los valores de las variables.

4.1 POWER TRANSFORMER POR MÉTODO BOX COX

Como se mencionó, una de las características que se ha observado en los datos de entrada es que estos poseen un fuerte sesgo en la mayoría de las variables, esto debe ser transformado antes de introducir las variables al modelo de aprendizaje automático. Para este fin utilizaremos el método de transformación por Box Cox que reducirá el sesgo que posee la distribución de cada una de las variables, generando una transformación en cada variable y variando su distribución como se muestra en la imagen.

Inicialmente la distribución de los datos es como se muestra en la imagen a continuación.

Tabla 4. 1. Histogramas de distribución de los parámetros del modelo antes de la transformación por el método de Box Cox.



Fuente: Elaboración propia.

Luego realizamos la transformación por el método de Box Cox para este fin usamos el siguiente código:

```
from sklearn.preprocessing import PowerTransformer
# extract features with skew
features = data[['UCS [MPa]',
x95[cm]','x80[cm]','x60[cm]','x50[cm]','Carga[Kg]']]

#instantiate
pt=PowerTransformer(method='box-cox', standardize = True,)
#Fit the data to the powertransformer
skl_boxcox=pt.fit(features)
#Lets get the Lambdas that were found
#print (skl_boxcox.lambdas_)
calc_lambdas = skl_boxcox.lambdas_
#Transform the data
skl_boxcox= pt.transform(features)
#Pass the transformed data into a new dataframe
df_features=pd.DataFrame(data=skl_boxcox, columns =['UCS [MPa]',
'x95[cm]','x80[cm]','x60[cm]','x50[cm]','Carga[Kg]'] )
# Pass to the original dataframe the transform columns
data.drop(['UCS [MPa]'], axis = 1, inplace= True)
data.drop(['x95[cm]'], axis = 1, inplace= True)
data.drop(['x80[cm]'], axis = 1, inplace= True)
data.drop(['x60[cm]'], axis = 1, inplace= True)
data.drop(['x50[cm]'], axis = 1, inplace= True)
data.drop(['Carga[Kg]'], axis = 1, inplace= True)

# Concatenar ambos dataframes
df_data = pd.concat([data, df_features], axis=1)
cols= df_data.columns.tolist()
# Pasar el último elemento al primero de la lista (2 veces)
cols =cols[-1:]+cols[:-1]
cols =cols[-1:]+cols[:-1]
cols =cols[-1:]+cols[:-1]
cols =cols[-1:]+cols[:-1]
```

```

cols =cols[-1:]+cols[:-1]
cols =cols[-1:]+cols[:-1]
#Sobreescribir
df_data = df_data[cols]
df_data

```

Obtenemos la base de datos transformada donde la distribución de los parámetros se ha transformado buscando una distribución normal, disminuyendo el sesgo para cada parámetro.

A continuación, se muestra la base de datos transformada.

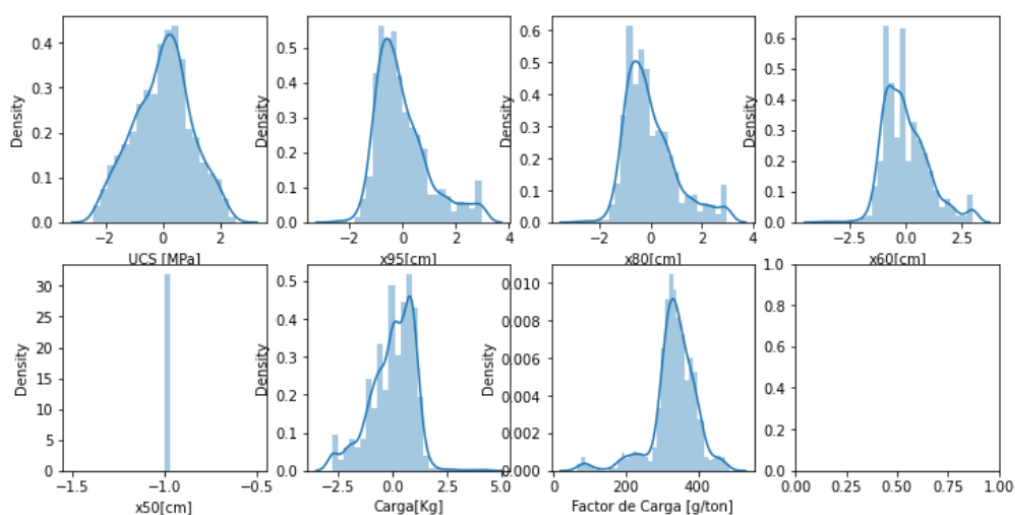
Tabla 4. 2. Base de datos resultante de la transformación de los parámetros por el método de Box Cox.

	UCS [MPa]	x95[cm]	x80[cm]	x60[cm]	x50[cm]	Carga[Kg]	Factor de Carga [g/ton]
0	-2.313471	0.808063	0.683853	0.211109	-1.0	-0.237857	315.0
1	-1.190551	-0.632172	-0.665568	-0.697486	-1.0	0.866984	329.0
2	-1.028778	-0.821443	-0.874427	-0.996308	-1.0	1.181857	325.0
3	-0.265004	-0.145014	-0.133864	-0.088416	-1.0	0.109646	311.0
4	0.347680	-0.821443	-0.874427	-0.996308	-1.0	1.181857	338.0
...
1080	0.935548	-0.563157	-0.590619	-0.603093	-1.0	0.766578	356.0
1081	-0.006568	-0.906320	-0.956267	-1.101599	-1.0	1.347208	364.0
1082	0.923550	-0.720805	-0.768327	-0.894006	-1.0	1.021825	361.0
1083	1.522164	-0.462675	-0.469602	-0.511338	-1.0	0.604139	348.0
1084	1.057641	-0.462675	-0.469602	-0.511338	-1.0	0.604139	349.0

Fuente: Elaboración propia.

Volvemos a realizar el plot de la distribución de los parámetros y observamos la diferencia, después de la transformación los parámetros tienen una distribución más normalizada.

Tabla 4. 3. Histograma de distribución de los parámetros después de la transformación por el método de Box Cox.



Fuente: Elaboración propia.

4.2 ESTANDARIZACIÓN

¿Por qué escalar, estandarizar o normalizar? Muchos algoritmos de aprendizaje automático funcionan mejor o convergen más rápido cuando las características están en una escala relativamente similar y / o cerca de la distribución normal. La estandarización es una técnica de escalado que asume que sus datos se ajustan a una distribución normal (Raschka, Sebastian & Mirjalili, Vahid 2017). Ejemplos de tales familias de algoritmos incluyen: regresión lineal y logística, k-vecinos más cercanos, redes neuronales, support vector machine con funciones de kernel de polarización

radial, análisis de componentes principales (PCA), y análisis discriminante lineal (LDA).

La estandarización es el proceso de transformar los datos a una media de 0 (cero) y una desviación estándar de 1 (uno). Importante para corregir la diferencia de escala entre los parámetros.

Esto se logra a partir de la función 'StandardScaler' de la librería scikitlearn, que estandariza una característica restando la media y luego escalando a la varianza unitaria. La varianza unitaria significa dividir todos los valores por la desviación estándar.

Al hacer la estandarización sobre los parámetros independientes estos datos poseen una escala más uniforme. Además, la estandarización es más adecuada para técnicas que asumen una distribución gaussiana en las variables de entrada, por este motivo antes de estandarizar hemos realizado la transformación de Box Cox que convierte la distribución de los datos a una distribución gaussiana.

Los parámetros a estandarizar son los inputs y no el output por lo que primero se debe separar las variables input de la variable target y para este fin usamos la función ".values", luego indicamos a "X" como el conjunto de los 6 parámetros de entrada con el siguiente código:

```
df_data
array = df_data.values
X = array[:,0:6]
X
```

"X" nos muestra el siguiente array de los inputs

```
array([[ -2.313,  0.808,  0.684,  0.211, -1.  , -0.238],
       [-1.191, -0.632, -0.666, -0.697, -1.  ,  0.867],
       [-1.029, -0.821, -0.874, -0.996, -1.  ,  1.182],
       ...,
       [ 0.924, -0.721, -0.768, -0.894, -1.  ,  1.022],
       [ 1.522, -0.463, -0.47 , -0.511, -1.  ,  0.604],
       [ 1.058, -0.463, -0.47 , -0.511, -1.  ,  0.604]])
```

Para indicar el output usaremos la variable “Y” con el siguiente código a manera análoga que con los inputs

```
Y = array[:,6]
Y
```

“Y” nos muestra el array del output

```
array([315., 329., 325., ..., 361., 348., 349.])
```

Luego de haber separado los inputs como “X” procedemos a estandarizarlos con el siguiente código:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler().fit(X)
rescaledX = scaler.transform(X)
np.set_printoptions(precision=3)
```

Imprimimos los parámetros inputs estandarizados como se muestra a continuación.

```
print(rescaledX[:,:])

[[ -2.313  0.808  0.684  0.211  0.  -0.238]
 [-1.191 -0.632 -0.666 -0.698  0.  0.867]
```

```
[-1.029 -0.821 -0.874 -0.996 0.  1.182]
...
[ 0.924 -0.721 -0.768 -0.894 0.  1.022]
[ 1.522 -0.463 -0.47  -0.511 0.  0.604]
[ 1.058 -0.463 -0.47  -0.511 0.  0.604]]
```

El resultado de esta transformación finalmente nos permite tener a todas las variables input a una misma escala, y con una distribución con poco sesgo. Siendo estas condiciones las adecuadas para empezar a realizar el modelado con los algoritmos de aprendizaje automático.

4.3 TRAIN- TEST - SPLIT

Una vez que las variables de la base de datos ya se encuentran listas para ser introducidas en un modelo mediante algoritmo de machine learning se procede a realizar el remuestreo de la base de datos para obtener una mejor representatividad de la predicción.

La función Train-Test-Split permite dividir la base de datos en conjunto de entrenamiento y validación en una relación 80 - 20.

El conjunto de entrenamiento será el 80% del total de datos en la base de datos ya que es necesario tener la mayor cantidad de datos posibles para el aprendizaje, y el 20 % servirá para la validación del modelo entrenado.

Para la división de la base de datos usamos el siguiente código:

```
validation_size = 0.20
seed = 7
```

```
X_train, X_validation, Y_train, Y_validation = train_test_split (rescaledX, Y,  
test_size=validation_size, random_state=seed)
```

4.4 MÉTRICAS

Al ser un problema de regresión la métrica que se usará es aquella asociada a medir la diferencia entre el valor predicho y el valor real, ejemplo de estas métricas son el error cuadrático medio y error medio absoluto.

CAPITULO V.

MODELADO CON REDES NEURONALES

El modelo de redes neuronales que se implementó en la presente tesis busca predecir el valor del factor de carga a partir de los datos de entrada (inputs) que son el target de fragmentación, el UCS del terreno y la carga de explosivo. De esta manera se genera un aprendizaje automático de predecir cuál es el factor de carga que genera las condiciones de fragmentación deseada, bajo las condiciones del terreno que se presenten y con una carga de explosivo que aprovecha al máximo el espacio perforado de cada hole (con taco crítico).

A continuación, se explicará el funcionamiento de la red neuronal.

Las redes neuronales tienen 3 partes principales y cada parte está compuesta de nodos. Estas son la capa de entrada, las capas ocultas y la capa de salida.

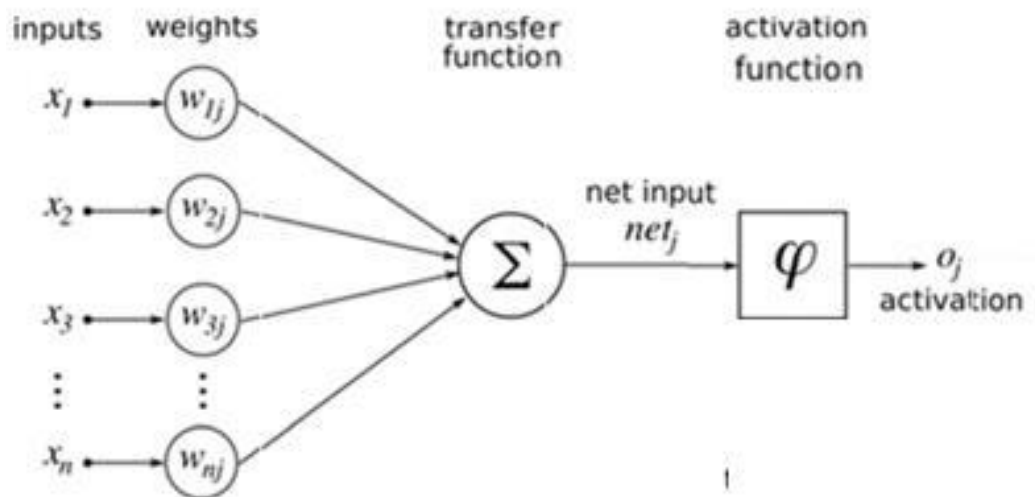


Figura 5. 1. Proceso esquematizado de las redes neuronales artificiales.

Fuente: Tomado de "<https://medium.com/@ricardojmv85/entendiendo-las-redes-neuronales-artificiales-1e0f4523eae6>."

En este sentido en la capa de entrada se colocaron las variables (inputs) de target de fragmentación, UCS de la roca, carga del explosivo todas estas son variables del tipo numérico mientras que en la capa de salida está el factor de carga que será predicho por la red neuronal artificial. El objetivo es al tener el valor el factor de carga derivado de esto obtener el valor del burden y espaciamiento que ira variando segun las condiciones del terreno (para este modelo representado por la variable UCS).

Las capas ocultas contienen la arquitectura de la red neuronal.

5.1 REDES NEURONALES ARTIFICIALES

5.1.1 PERCEPTRON

Dentro de los tipos de redes neuronales artificiales encontramos al perceptrón siendo el más simple de los tipos de redes neuronales artificiales.

Es un modelo de una simple neurona que puede ser usado para problemas de clasificación binaria, a partir de esta se pueden crear modelos más complejos de redes neuronales.

El algoritmo del perceptrón está inspirado en la forma del procesamiento de información de una neurona humana. Una neurona recibe señales inputs desde sus dendritas, las cuales son transformadas en señales eléctricas hacia la neurona y luego son transmitidas hacia otras células. De manera similar, el perceptrón recibe señales de entrada de ejemplos de datos de entrenamiento que ponderamos y combinamos en una ecuación lineal llamada activación que genera el valor del output o predicción.

$$activation = bias + \sum_{i=1}^n weight_i * x_i$$

5.1.2 DESCENSO DE GRADIENTE ESTOCÁSTICO

La gradiente del descenso es el proceso de minimizar una función siguiendo la pendiente o gradiente de esa función. El perceptrón usa el gradiente de descenso estocástico que actualiza los pesos de la ponderación en la red neuronal. En cada iteración del gradiente del descenso el peso se actualiza usando la siguiente ecuación:

$$w = w + (tasa\ de\ aprendizaje) * (valor\ esperado - valor\ predicho) * x$$

Donde w es el peso que se está optimizando, la tasa de aprendizaje es una tasa de aprendizaje que debe configurar, (por ejemplo, 0.01), (esperado-predicho) es el error de predicción para el modelo en los datos de entrenamiento atribuido al peso y x es el valor de entrada. Es de esta manera que se calibra la mejor ponderación de los inputs o variables de entrada obteniendo la predicción con el error mínimo.

5.1.3 ALGORITMO DE BACKPROPAGATION

La propagación hacia atrás de errores o retropropagación es un método de cálculo del gradiente utilizado en algoritmos de aprendizaje supervisado utilizados para entrenar redes neuronales artificiales. El método emplea un ciclo propagación – adaptación de dos fases. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas siguientes de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Sin embargo, las neuronas de la capa oculta solo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total.

La importancia de este proceso consiste en que, a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas de tal

modo que las distintas neuronas aprenden a reconocer distintas características del espacio total de entrada. Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento.

La red retropropagada trabaja bajo aprendizaje supervisado y por tanto necesita un conjunto de entrenamiento que le describa cada salida y su valor de salida esperado.

5.1.4 FUNCIÓN DE PÉRDIDA

Una función de pérdida, o Loss function, es una función que evalúa la desviación entre las predicciones realizadas por la red neuronal y los valores reales de las observaciones utilizadas durante el aprendizaje. Cuanto menor es el resultado de esta función, más eficiente es la red neuronal. Su minimización, es decir, reducir al mínimo la desviación entre el valor predicho y el valor real para una observación dada, se hace ajustando los distintos pesos de la red neuronal.

Podemos tener en la función de pérdida al error lineal o error local que como acabamos de indicar, el error de aprendizaje, también llamado error local, se calcula obteniendo la diferencia entre el valor real que hay que predecir y el valor predicho por la neurona artificial.

$$\text{Error} = \text{Prediccion_real} - \text{Prediccion_realizada}$$

Este es el error que buscaremos minimizar durante los aprendizajes. Este error se puede considerar como error local porque se centra en una observación dada comparando el valor real y el valor predicho. También se tiene al error cuadrático medio MSE o error global

El error cuadrático medio es una función de error global del aprendizaje. Es decir, que esta función nos permitirá conocer de manera global el porcentaje de error cometido por nuestra neurona.

5.2 PREDICCIÓN VERSUS DATA DE VALIDACIÓN

A continuación se muestra la predicción del factor de carga obtenido mediante redes neuronales artificiales versus el valor del factor de carga obtenido experimentalmente, se compararon mediante una gráfica donde la línea de color rojo indica el valor real del factor de carga de los holes usados para la validación (valor experimental) y la línea de color azul es el factor de carga predicho para cada hole usado para validación, se observa que se ha minimizado el error en la predicción, y los valores son muy precisos, el error obtenido para la predicción usando la métrica del error medio absoluto es de 16.2, este se obtiene como resultado de la diferencia absoluta promedio entre la variable experimental y la variable predicha.

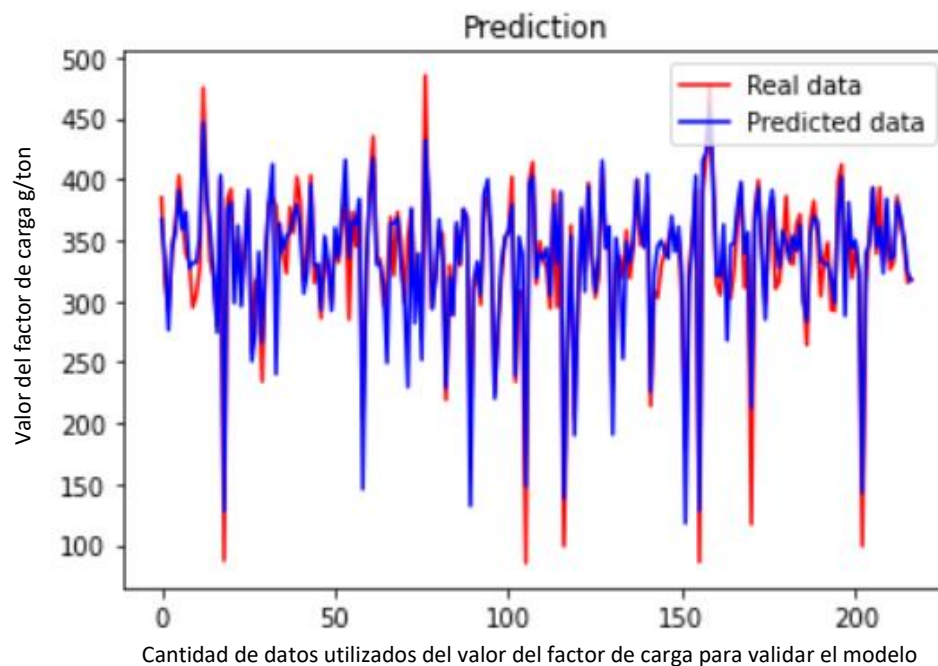


Figura 5. 2. Predicción del factor de carga para los holes de los datos de validación (gráfico de validación del modelo predictivo).

Fuente: Elaboración propia.

Los resultados de la predicción versus la data experimental para la validación como se ve en el gráfico son muy cercanos, estos resultados se obtienen después de hacer el entrenamiento y como resultado de ejecutar el modelo de redes neuronales artificiales sobre los datos de validación.

A continuación, se realizó una nueva predicción usando el modelo de redes neuronales artificiales esta vez sobre todos los holes de una malla de voladura. Y nuevamente se observa que los resultados de los valores del factor de carga predicho y los experimentales son muy cercanos.

Con lo cual podemos concluir que el modelo de redes neuronales artificiales incluso para nuevos valores de factor de carga de predicción genera resultados muy precisos.

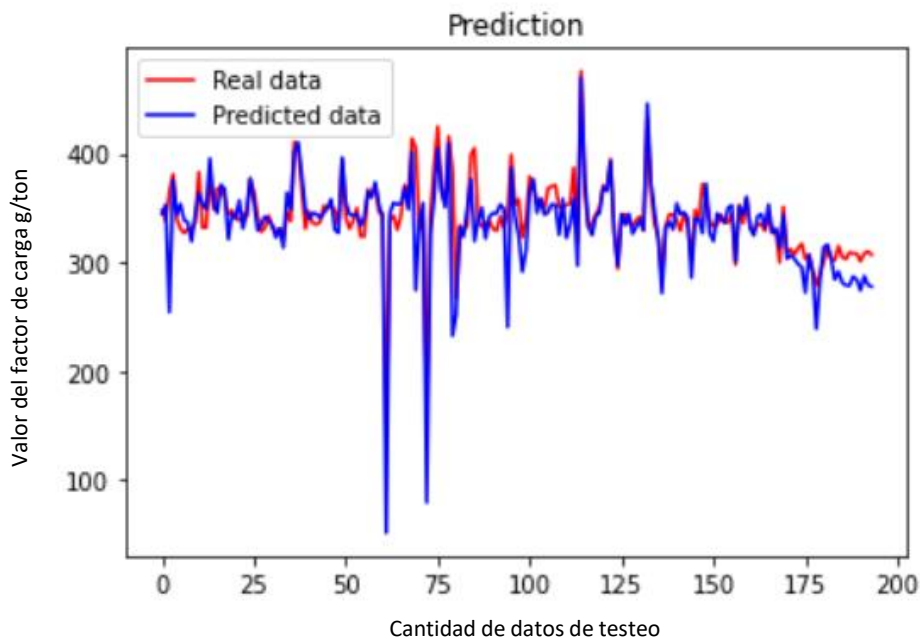


Figura 5. 3. Predicción del factor de carga para holes de una nueva malla con datos no entrenados.

Fuente: Elaboración propia.

Sintetizando lo escrito líneas arriba y para describir mejor la información que nos brinda la figura 5.3.; por ejemplo, si se cuenta con 1000 datos, de los cuales 800 datos serán usados para crear el modelo y 200 datos serán utilizados para validar el modelo. Entonces, una vez realizado el modelo predictivo y realizada la validación de este, se procede a realizar un testeo, es decir predecir sobre nuevos datos que no participaron en ninguna etapa del proceso, y si tenemos que el testeo resulta correcto entonces el modelo predictivo funciona, es decir lo real y que se predice están coincidiendo.

CAPITULO VI.

IMPLEMENTACIÓN EN LOS PROCESOS DE VOLADURA

La red neuronal artificial que se ha implementado permite predecir el valor del factor de carga en todo el terreno en función del UCS de las rocas, la fragmentación deseada y la carga de explosivo que se usa. Esto nos da la ventaja de controlar en los datos de entrada la cantidad de explosivo a usar por ejemplo podemos ingresar como input una carga de explosivo de taco crítico de manera que se aproveche al máximo el espacio en cada hole, las condiciones del terreno son una variable independiente que depende de la zona y no de la operación, esta no la manipulamos, y la fragmentación que es una variable que podemos controlar en esta variable se introduce como input cuál es el tamaño de fragmento que deseamos obtener, finalmente como target está el valor del factor de carga que está asociado a cada unidad de espacial de nuestro banco donde se hará la voladura.

Con los valores predichos del factor de carga en todo el terreno del banco de voladura es que se obtiene la recomendación de los burden y espaciamiento para el diseño de malla y la perforación de los holes.

Luego, dado que el terreno no es uniforme en resistencia mecánica los valores de factor de carga variarán para cada cambio de respuesta mecánica del terreno complicando esto el diseño de la malla para voladura. Por este motivo, para poder realizar un diseño de malla factible desde el punto de vista técnico es que se realiza la clusterización de los valores de la respuesta mecánica del terreno representados por la variable UCS, de esta manera se define al terreno como un número determinado de grupos litológicos que se tomarán a criterio del usuario especialista.

6.1 VARIABLES USADAS

- Variable independiente: UCS
- Variable dependiente: grupo litológico

UCS asociado al comportamiento geológico del terreno, se clusteriza en grupos que representen los principales grupos litológicos.

Nota: Estas variables son utilizadas solo para el proceso de implementación del modelo predictivo en una unidad minera.

6.2 CLUSTERIZACIÓN DEL TERRENO EN GRUPOS LITOLÓGICOS.

El proceso de clusterización permite generar un diseño viable de ser ejecutado desde el punto de vista técnico. La clusterización del terreno en grupos litológicos se realiza a partir del uso de un algoritmo de aprendizaje no supervisado, específicamente el algoritmo de “K means” que clusterizará las respuestas mecánicas y de esta manera obtendremos un número limitado y menor de respuestas mecánicas del terreno, así como sus factores de carga asociados. Se debe tener en cuenta el expertise de un minero para indicar al algoritmo de aprendizaje no supervisado el número de clusters a considerar en su modelo.

Finalmente definimos el diseño de malla para cada respuesta mecánica obtenida después de la clusterización. Este diseño de malla cumple las condiciones de fragmentación deseada y usa la carga de explosivo con taco crítico en cada hole. En la imagen a continuación se muestra la comparación entre el mapa de calor para un terreno con UCS variando para cada unidad de área y el mapa de calor del mismo terreno después de realizar la clusterización con K-means, habiendo definido un valor de UCS para cada área clusterizada, así como su respectivo factor de carga predicho.



Figura 6. 1. Imagen que esquematiza el comportamiento unitario por celdas del factor de carga recomendado en un mapa de calor resultado de la predicción con redes neuronales artificiales.

Fuente: Elaboración propia.

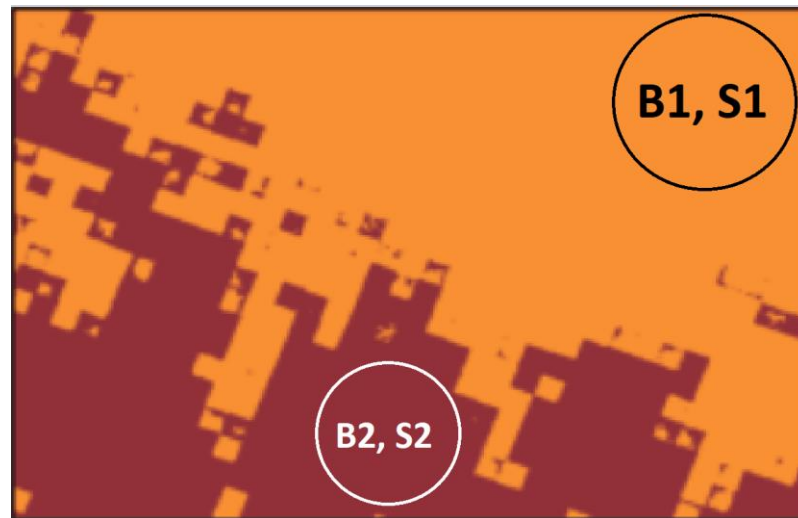


Figura 6. 2. Imagen que esquematiza la clusterización del factor de carga en un mapa de calor resultado de la predicción con las redes neuronales artificiales. Se observan los burden y espaciamiento que se recomienda para cada grupo litológico.

Fuente: Elaboración propia.

En la segunda imagen se observa los burden y espaciamiento para cada una de las dos respuestas mecánicas; es así que tenemos un B1 (burden uno), S1 (espaciamiento uno) para la primera respuesta mecánica, y un B2 (burden dos), S2 (espaciamiento dos) para la segunda respuesta mecánica del terreno respectivamente.

Una consideración importante es definir la cantidad de grupos litológicos asociados al grupo de respuestas mecánicas esto en base a criterios geológicos, porque en base a estos se generan las recomendaciones de burden y espaciamiento.

6.3 COMPARATIVA RED NEURONAL ARTIFICIAL VERSUS MÉTODO TRADICIONAL DE VOLADURA.

A continuación, se realiza una comparación en el uso de recursos al usar una malla de voladura realizada por los métodos tradicionales y otra usando las consideraciones de la red neuronal artificial.

6.3.1 MÉTODO TRADICIONAL

Empecemos con la determinación del diseño de malla usando el burden por espaciamiento generado por un método tradicional, para este fin a manera general se siguen los siguientes pasos:

- *Se evalúan las características del explosivo.*
- *Se evalúan los valores de taco máx. y min (crítico).*
- *Se evalúan las características de roca.*
- *Se evalúa la fragmentación teórica.*
- *Se define el diseño en burden x espaciamiento (Ejemplo 7x7 trabada).*
- *Se evalúa el factor de carga teórico.*
- *Se evalúa la ubicación teórica de pozos.*
- *Se determina el factor de carga real pozo a pozo.*
- *Se determina la ubicación de pozos real y largo real.*
- *Se determina la carga del explosivo recomendada para cada pozo.*

Supongamos que nuestra malla es cuadrada, el Gblast en base a la información de las condiciones geomecánicas de la roca y cálculos de distribución de fragmentación, nos entrega el valor de la carga de explosivo en cada hole.

$$FC(g/ton) = W_{exp} / B * E * H * \rho_{roca}$$

Donde:

W_{exp} = peso del explosivo (g)

B: Burden (m)

E: Espaciamiento (m)

H: Altura del banco (m)

ρ_{roca} : Densidad de la roca (ton/m³)

6.3.2 PROCESAMIENTO MÓDULO OPTIMIZACIÓN DE DISEÑO USANDO RED NEURONAL ARTIFICIAL

Para el método con red neuronal artificial se predice el valor del factor de carga teniendo como input el valor de la carga del explosivo usando taco crítico, el valor del target de fragmentación, y el UCS de la roca en el área de voladura.

A continuación, realizaremos la comparativa entre diseño de malla para voladura basado en las redes neuronales artificiales versus el método tradicional, debemos estimar el burden por espaciamiento, supongamos que se trabajara bajo una malla cuadrada, las variables que se repiten en ambos modelos son las siguientes:

- El mismo UCS
- El mismo target de fragmentación.

Mientras que las variables que cambian son:

- La Carga de explosivo
- El Factor de carga

Para la comparativa, se introducirá en la red neuronal artificial los inputs de las condiciones de una malla de voladura ejecutada por el método tradicional, modificando el valor de la carga del explosivo cambiando este por el valor de carga para un taco crítico supuesto, ya que no se posee el valor de la altura del banco para tener el valor real del taco crítico.

Teniendo los resultados de una voladura ejecutada, se observa cual es el factor de carga recomendado al ejecutar la red neuronal artificial y se realiza la comparación

A continuación, se muestran las condiciones iniciales de un pozo al haber usado el método tradicional.

Hole, UCS [MPa], x95 [cm], x80 [cm], x60 [cm], x50 [cm], Carga [Kg], Factor de Carga [g/ton]

P1004, 65.05, 18.68, 10.25, 5.95, 4.542, 941.0, 340.0

Las condiciones iniciales indican un valor de 941 Kg de carga de explosivo, donde para los siguientes cálculos asumimos los siguientes valores:

Carga lineal de explosivo = 100.4 Kg/m

Densidad de explosivo = 1.32 gr/cc

Densidad roca = 2.6

Con lo cual se obtiene cual es la longitud equivalente a los kilogramos de carga de explosivo usados:

941.0 Kg de explosivo = Longitud de carga de 9.37 metros

Los datos mostrados son las condiciones que enmarcan lo recomendado para la voladura al usar el método tradicional de voladura.

Dado que no se conoce la altura del banco, asumamos la carga máxima considerando taco crítico es de 1041 kg, 100 kg más que el usado en el método tradicional siendo un equivalente a 10.368 m de explosivo. En este primer paso estamos recuperando aproximadamente 1m perforado de pozo al usar el taco crítico, 1 metro perforado que no se estaba aprovechando y se estaba rellenando con material de relleno.

A continuación, procedemos a ejecutar el algoritmo de redes neuronales artificiales a fin de obtener el valor del factor de carga recomendado dado el cambio en la carga de explosivo, pero manteniendo el target de fragmentación.

Donde al ejecutar la red neuronal artificial considerando los inputs mencionados anteriormente se obtiene un factor de carga de 328.70456. Comparando el factor de carga inicial versus el generado por la red neuronal artificial tenemos:

FC tradicional: 340.0

FC ANN: 328.70456

Veamos el siguiente ejemplo de aplicación usando los resultados obtenidos donde se obtendrá los valores del burden por espaciamiento recomendado para cada caso.

Inicialmente por el método tradicional tenemos un factor de carga de 340.

$$FC = 340(g/ton) = \frac{941000 g}{x1^2 * H * d}$$

Por el método de redes neuronales artificiales el valor del factor de carga recomendado es de 328.7

$$328.7(g/ton) = \frac{1041000 g}{x2^2 * H * d}$$

Para efectos de cálculo y comparar los resultados tomamos H = 15m como altura del banco y d = 2.6 kg/m³ como densidad de la roca y obtenemos que:

$$x1^2 = 70.96 \text{ y } x2^2 = 81.20$$

$$x1 = 8.42 \text{ y } x2 = 9.011$$

Si nuestro terreno considerando UCS uniforme posee área de 1000 m^2 tendríamos:

Opción Tradicional: $1000/70.96 = 14.09$ pozos, x $15 \text{ m c/u} = 211.3 \text{ m}$ perforados

Opción ANN: $1000/81.20 = 12.31$ pozos, x $15 \text{ m c/u} = 184.7 \text{ m}$ perforados.

Se ha reducido la perforación en un **12.5 %**.

Esta mejora en la perforación del diseño de malla impactará sobre los costos operacionales de perforación y voladura generando así una mayor rentabilidad en el proceso.

CONCLUSIONES

- En la voladura actual la carga de explosivo en los pozos perforados no siempre aprovecha la máxima su capacidad de carga considerando el taco crítico generando un sobredimensionamiento de los recursos utilizados.
- De acuerdo con la obtención de resultados mostradas en el gráfico 6.2, se demuestra que se logró que el método de redes neuronales artificiales propuesto prediga la ubicación óptima de los pozos de voladura para cada grupo litológico, aprovechando la carga de explosivo usando taco crítico.
- En base al apartado 6.3.2, se llega a concluir que mediante el método de redes neuronales artificiales propuesto se obtuvo un diseño de malla, el cual genera un ahorro del 12.5% en la perforación de pozos con los mismos valores de fragmentación.
- Al utilizar el método de redes neuronales artificiales propuesto se genera un diseño de malla con optimización en los metros perforados para voladura,

reduciendo costos operativos en el proceso de perforación y voladura de rocas.

- El método propuesto muestra una gran adaptabilidad a las necesidades del usuario, por lo cual su utilidad es para cualquier unidad minera, siendo un aporte a la minería en general.
- Este modelo de redes neuronales artificiales ofrece una mejora alternativa para los métodos de voladura actuales debido a una mejor precisión, y aprovechamiento de recursos.
- Los grupos litológicos son clusterizados usando algoritmos de clusterización de aprendizaje no supervisado como el K-means

RECOMENDACIONES

- Se debe considerar el criterio geológico para definir la cantidad de grupos litológicos al momento de usar los algoritmos de clusterización.
- Se recomienda seleccionar solo un target de fragmentación específico para ejecutar el modelo de redes neuronales artificiales y predecir el factor de carga del que se obtendrá el burden y espaciamiento para el diseño de malla de voladura.
- Se recomienda actualizar la red neuronal artificial con más datos y ver la respuesta de las predicciones con un mayor número de datos de entrenamiento, a fin de mejorar aún más el modelo predictivo.
- Se recomienda ver el comportamiento de otras variables asociadas a la voladura y generar nuevos modelos de predicción que podrían dar nuevos resultados y mejoras en el diseño de voladura.

BIBLIOGRAFÍA

- Artificial neural network approach to predict blast-induced ground vibration, airblast and rock fragmentation, Tiile, Raymond N. (2016)
- Python Machine Learning, Machine Learning and Deep Learning with Python, scikitlearn and TensorFlow, Raschka, Sebastian & Mirjalili, Vahid (2017)
- Machine Learning Algorithms From Scratch (Brownlee, Jason (2017))
- Prediction of Rock Fragmentation in Open Pit Mines, using Neural Network Analysis (K. Oraee (2006))
- D. Michie, D.J. Spiegelhalter, C.C. Taylor (eds). Machine Learning, Neural and Statistical Classification, 1994
- R. Rojas. *Neural Networks: A Systematic Introduction*, Springer, 1996
- Oraee, K.; Asi, B. Prediction of Rock Fragmentation in Open Pit Mines, using Neural Network Analysis. In Proceedings of the Fifteenth International Symposium on Mine Planning and Equipment Selection (MPES 2006), Torino, Italy, 19–22 September 2006.
- Shi, X.; Zhou, J.; Wu, B.B.; Huang, D.; Wei, W. Support vector machines approach to mean particle size of rock fragmentation due to bench blasting prediction. *Trans. Nonferr. Met. Soc. China* 2012, 22, 432–441. [CrossRef]
- Sayadi, A.; Monjezi, M.; Talebi, N.; Khandelwal, M. A comparative study on the application of various artificial neural networks to simultaneous prediction of rock fragmentation and backbreak. *J. Rock Mech. Geotech. Eng.* 2013, 5, 318–324.

- Saadat, M.; Khandelwal, M.; Monjezi, M. An ANN-based approach to predict blast-induced ground vibration of Gol-E-Gohar iron ore mine, Iran. *J. Rock Mech. Geotech. Eng.* 2014, 6, 67–76.
- Enayatollahi, I.; Bazzazi, A.A.; Asadi, A. Comparison between neural networks and multiple regression analysis to predict rock fragmentation in open-pit mines. *Rock Mech. Rock Eng.* 2014, 47, 799–807.
- Tiile, R.N. Artificial Neural Network Approach to Predict Blast-Induced Ground Vibration, Airblast and Rock Fragmentation. Master's Thesis, Missouri University of Science and Technology, Rolla, MI, USA, 2016.
- Castillo-Martinez, A.; Ramon Almagro, J.; Gutierrez-Escolar, A.; Del Corte, A.; Castillo-Sequera, J.L.; Gómez-Pulido, J.M.; Gutiérrez-Martínez, J.M. Particle Swarm Optimization for Outdoor Lighting Design. *Energies* 2017, 10, 141.
- Dhekne, P.; Pradhan, M.; Jade, K.; Mishra, R. Boulder prediction in rock blasting using artificial neural network. *ARPN J. Eng. Appl. Sci.* 2017, 12, 47–61.
- Asl, P.F.; Monjezi, M.; Hamidi, J.K.; Armaghani, D.J. Optimization of flyrock and rock fragmentation in the Tajareh limestone mine using metaheuristics method of firefly algorithm. *Eng. Comput.* 2018, 34, 241–251.
- Lawal, A.I.; Idris, M.A. An artificial neural network-based mathematical model for the prediction of blast-induced ground vibrations. *Int. J. Environ. Stud.* 2020, 77, 318–334.
- Kanchibotla, S.S. Optimum blasting? Is it minimum cost per broken rock or maximum value per broken rock? *Fragblast* 2003, 7, 35–48.

- Cunningham, C. The Kuz-Ram Model for production of fragmentation from blasting. In Proceedings of the Symposium on Rock Fragmentation by Blasting, Lulea, Sweden, 23–26 August 1983.
- Kuznetsov, V. The mean diameter of the fragments formed by blasting rock. *Sov. Min.* 1973, 9, 144–148.
- Ruder, S. An overview of gradient descent optimization algorithms. arXiv 2016, arXiv:1609.04747.
- Silva, J.; Amaya, J.; Basso, F. Development of a predictive model of fragmentation using drilling and blasting data in open pit mining. *J. S. Afr. Inst. Min. Metall.* 2017, 117, 1089–1094.
- Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* 1989, 2, 303–314.
- Hecht-Nielsen, R. Counterpropagation networks. *Appl. Opt.* 1987, 26, 4979–4984.
- C. V. Cunningham: Fragmentation estimations and the Kuz-Ram model - Four years on. Proceedings of the second international symposium on rock fragmentation by blasting, Keystone, Colorado, 1987, 475-487.
- I. Onederra, S. Esen, and A. Jankovic: Estimation of fines generated by blasting - applications for the mining and quarrying industries. *IMM transactions, Mining Technology*, Vol 113, 2004, No.4:237-247.
- F. Ouchterlony: The Swebrec function: linking fragmentation by blasting and crushing. *IMM transactions, Mining Technology*, Vol 114, March 2005, No1:A29-A44.

GeoBlast S:A, Final Report PRO–ASP–246/07-E, “Modelamiento de la Fragmentación Resultante de Tronadura en Roca Primaria”. Emitido para: Proyecto Expansión Andina, CODELCO, Abril 9, 2008.

Enayatollahi, Iman; Aghajani Bazzazi, Abbas; Asadi, Ahamad, 2014. Comparison between Neural Networks and Multiple Regression Analysis to Predict Rock Fragmentation in Open-Pit Mines. *Rock Mechanics and Rock Engineering*, Volume 47, Issue 2, pp. 799 – 807.

Khandelwal, Manoj and Singh, T.N., 2009. Prediction of blast-induced ground vibration using artificial neural network. *International Journal of Rock Mechanics and Mining Sciences*, Volume 46, Issue 7, pp. 1214 – 1222.

M. Hajihassani; D. Jahed Armaghani; H. Sohaei; E. Tonnizam Mohamad; A. Marto., 2014. Prediction of airblast-overpressure induced by blasting using a hybrid artificial neural network and particle swarm optimization. *Applied Acoustics*, Volume 80, pp. 57 – 67.

Monjezi, M; Bahrami, A; Yazdian Varjani, A., 2010. Simultaneous prediction of fragmentation and flyrock in blasting operation using artificial neural networks. *International Journal of Rock Mechanics and Mining Sciences*, Volume 47, Issue 3, pp. 476 – 480.

Neaupane, Krishna Murari, and N. R. Adhikari, 2006. Prediction of tunneling-induced ground movement with the multi-layer perceptron. *Tunnelling and Underground Space Technology* 21.2, 151-159.

Monjezi, M., Rizi, S. H., Majd, V. J., and Khandelwal, M., 2014. Artificial neural network as a tool for backbreak prediction. *Geotechnical and Geological Engineering*, 32(1), 21-30.

- P.K. Simpson, 1990. *Artificial Neural Systems. Foundation, Paradigms, Applications, and Implementations.* Pergamon Press, New York, NY.
- Yang Y, Zang O., 1997. A hierarchical analysis for rock engineering using artificial neural networks. *Rock Mech Rock Eng* 30:207-222.
- Bahrami, A., Monjezi, M., Goshtasbi, K., & Ghazvinian, A., 2011. Prediction of rock fragmentation due to blasting using artificial neural network. *Engineering with Computers*, 27(2), 177-181.
- Sayadi, A., Monjezi, M., Talebi, N., and Khandelwal, M., 2013. A comparative study on the application of various artificial neural networks to simultaneous prediction of rock fragmentation and backbreak. *Journal of Rock Mechanics and Geotechnical Engineering*, 5(4), 318-324.
- Kanchibotla S., 2001. Optimum blasting? Is it minimum cost per broken rock or maximum value per broken rock? *Proceedings of Explo, Australasian institute of mining and metallurgy.*
- Hustrulid W., 1999. *Blasting principles for open pit mining, vol1.* A.A. Balkema, Rotterdam.
- Michaux S, Djordjevic N., 2005. Influence of explosive energy on the strength of the rock fragments and SAG mill throughput. *Miner Eng* 18:439–448.
- Kuzu, C; Fisne, A; Ercelebi, S.G, 2009. Operational and geological parameters in the assessing blast induced airblast-overpressure in quarries. *Applied Acoustics*, ISSN 0003-682X, Volume 70, Issue 3, pp. 404 – 411.
- Monjezi, M., Amiri, H., Farrokhi, A., & Goshtasbi, K., 2010. Prediction of rock fragmentation due to blasting in Sarcheshmeh copper mine using artificial

neural networks. *Geotechnical and Geological Engineering*, 28(4), 423-430.

Wu, Chengqing and Hao, Hong, 2007. Modeling of simultaneous ground shock and airblast pressure on nearby structures from surface explosions. *International Journal of Impact Engineering*, ISSN 0734-743X, Volume 31, Issue 6, pp. 699 – 717.

Persson PA, Holmberg R, Lee J. *Rock blasting and explosives engineering*, 1994. In: editors, C.R.C. Press; pp. 375–86, 515.

Dusenberry, Donald, 2010. *Handbook for Blast Resistant Design of Buildings*, pages 8-9.

Tawadrous AS., 2006. Evaluation of artificial neural networks as a reliable tool in blast design. *Int Soc Explos Eng*, 1,1–12.

Rezaei, M., Monjezi, M., Moghaddam, S. G., & Farzaneh, F., 2012. Burden prediction in blasting operation using rock geomechanical properties. *Arabian Journal of Geosciences*, 5(5), 1031-1037.

Sarkar, Kripamoy, Avyaktanand Tiwary, and T. N. Singh, 2010. Estimation of strength parameters of rock using artificial neural networks. *Bulletin of engineering geology and the environment*, 69.4, 599-606.

Mostafa TM., 2009. Artificial neural network for prediction and control of blasting vibrations in Assiut (Egypt) limestone quarry. *Int J Rock Mech Min Sci* 46(2):426–431.

U. Langefors, B. Kihlstrom, 1963. *The modern technique of rock blasting*. Wiley, New York.

Monjezi, M., M. Ghafurikalajahi, and A. Bahrami, 2011. Prediction of blast-induced ground vibration using artificial neural networks. *Tunnelling and Underground Space Technology* 26.1, 46-50.

U. Langefors, B. Kihlstrom, 1963. *The modern technique of rock blasting*. Wiley, New York.

Kuznetsov V. M., 1973. The mean diameter of the fragments formed by blasting rock. *Soviet Mining Science*, 144-148.

ANEXOS

Anexo 1.

"Resumen Perfil de Pozo"

"Machine": Código de la máquina operadora

"Banco": Código del banco perforado

"Malla": Código de la malla

"Hole": ID del pozo perforado

"Tpo. Perforac.(Seg)": Código del tipo de perforación

"Profundidad (Mts)": Profundidad del pozo perforado

"Diseño Prof.(Mts)": Profundidad del pozo en el diseño

"Este": Coordenada este del pozo perforado

"Diseño Este": Coordenada este del pozo en el diseño

"Norte": Coordenada norte del pozo perforado

"Diseño Norte": Coordenada norte del pozo en el diseño

"Elev. Collar":

"Banco Destino":

"WOB (Kn)":

"Torque (Kn/mts)":

"Air (Psi)":

"ROP (Mts/Hr)":

"RPM":

"BI (%)":

"Toma de Muestra":

"Prof. Muestreo (Mts)":

"Tipo Roca": Clasificación de roca por su resistencia mecánica

"Tipo Pozo" : Drill o Redrill