

Artificial Intelligence-based Predictive Analysis of Self-Advancing Goaf Edge Support (SAGES) for Improving Underground Mines Safety

Ankit Singh (✉ ankit.2015DR1075@me.ism.ac.in)

Indian Institute of Technology (Indian School of Mines): Indian Institute of Technology
<https://orcid.org/0000-0002-3705-0330>

Aman Mittal

Indian Institute of Technology (Indian School of Mines): Indian Institute of Technology

Gunjan Haldar

Indian Institute of Technology (Indian School of Mines): Indian Institute of Technology

Dheeraj Kumar

Indian Institute of Technology (Indian School of Mines): Indian Institute of Technology

Research Article

Keywords: Analytical, Operational and Artificial Intelligence, Intelligent fault detection, Fault Analysis, Diagnostics and Monitoring, Machine Learning, Predictive Analysis, Internet-of-Things and Cyber-physical systems, Underground coal mines safety

Posted Date: February 1st, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1310552/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Artificial Intelligence-based Predictive Analysis of Self-Advancing Goaf Edge Support (SAGES) for Improving Underground Mines Safety

Ankit Singh^a, Aman Mittal^b, Gunjan Haldar^b, Prof. Dheeraj Kumar^a

^a *Department of Mining Engineering*

^b *Department of Mechanical Engineering,*

Indian Institute of Technology (ISM), Dhanbad 826004, India

ABSTRACT

An efficient and robust communication channel is a critical factor in underground coal mines to ensure the safety of the working environment. Through this paper, we explore how the functionalities and operational and analytical efficiency of SAGES can be analyzed utilizing historical data acquired during the depillaring operations carried out in underground coal mines using SAGES for safety and productivity during deployment with the incorporation of IoT with Artificial Intelligence. Applying predictive machine learning of convergence, load at withdrawal, and duration of yielding, in hrs will automate the analysis of the after-effects of depillaring operations. This will ultimately help in the maintenance of components of SAGES and measures to be taken on the safety of the workplace. Linear estimators, gradient boosting, and clustering methods are employed to detect the output properties of the deployed SAGES. Further, we illustrate and analyze a fine-tuning approach for supervised and unsupervised algorithms subsequently, which are analyzed and the performance of various estimator models in our experiments are compared. To end with, we perform an ensemble of the models for each target variable.

Keywords – Analytical, Operational and Artificial Intelligence, Intelligent fault detection, Fault Analysis, Diagnostics and Monitoring, Machine Learning, Predictive Analysis, Internet-of-Things and Cyber-physical systems, Underground coal mines safety

1. INTRODUCTION

Self-Advancing (mobile) Goaf Edge Support (SAGES) [Singh et al., 2016] have been developed by IIT (ISM), Dhanbad in collaboration with Jaya Bharat Equipment Pvt. Ltd., Hyderabad in the EOI (Expression of Interest) project of Ministry of Coal, Govt. of India for the depillaring operations undertaken in underground coal mines. SAGES (see Figure 1) have been developed for depillaring operations in underground coal mines to avoid labor-intensive and time-consuming processes in erecting wooden props as a support system at goaf¹ edges (working front) for protection of roof during depillaring operations. Goaf edges of the depillaring area in an underground coal mine are inherently high-stress zones and the support systems erected at such places should be capable of providing adequate support to the roof (see Figure 2). In the caving system of depillaring, it is very important that the initial, periodic, and main falls of roof take place timely inside the goaf to avoid building up of the roof pressure to an unsafe limit. This is also equally important that all such falls are restricted inside the goaf edges and do not extend up to the working area. Goaf edge supports play a very vital role in achieving these objectives by working as "Breaker Line Supports". However, care is to be taken, as regards to the selection of support and its capacity, which would help in creating a stiffer and operationally effective breaker line to facilitate regular fall in the goaf resulting in better strata management. Goaf edge supports also work as a substantial physical barrier between the supported area of extraction and the goaf and, thus discourage the persons from entering the goaf.

¹ Goaf: The part of a mine from which the coal has been mined (taken out) and the space filled up with caved rock



Figure 1: SAGES in fully raised position deployed in mines

SAGES when deployed in underground coal mines are operated using a radio remote control which eliminates the danger to workers from being exposed to hazardous roof conditions associated with setting and withdrawal of conventional roof supports to carry out the operations.



Figure 2: SAGES deployed for field trial at an underground mine in RK7 underground coal mine, Singareni Colliery Company Limited, India²

One of the critical contributing factors of accidents is the lack of predictive tools and methods of failure of roof and sides, thereby leading to failure to withdraw or removal of workers before the actual failure. The historical data are collected from the deployed SAGES machines in the underground coal mines between the period of 2017-2018. The historical data are going to be used for the prediction of slices³ (see Figure 3 that shows seam called 29L where the shaded area was a slice which became goaf after coal extraction and SAGES is deployed to support the unstable roof for the further extraction of coals from the rest of the 29L slices) during the extraction of SAGES using newly proposed AI models. Through these predictive analytical results, required steps can be undertaken before

² Picture used from SAGES project industry partner website <http://www.jaya-bharat.com/>

³ Slices are generated when a coal seam is divided/cut into suitable thickness for mining.

deployment of the SAGES through which, safety in underground coal mines can be improved. Following predictions are going to be performed using the proposed model (objectives):

- a. Maximum load at the withdrawal of SAGES after completion of a slice.
- b. Maximum convergence between canopy and base of the SAGES at time of withdrawal of the SAGES.
- c. Prediction of yielding of the SAGES during extraction of a slice and duration of yielding.

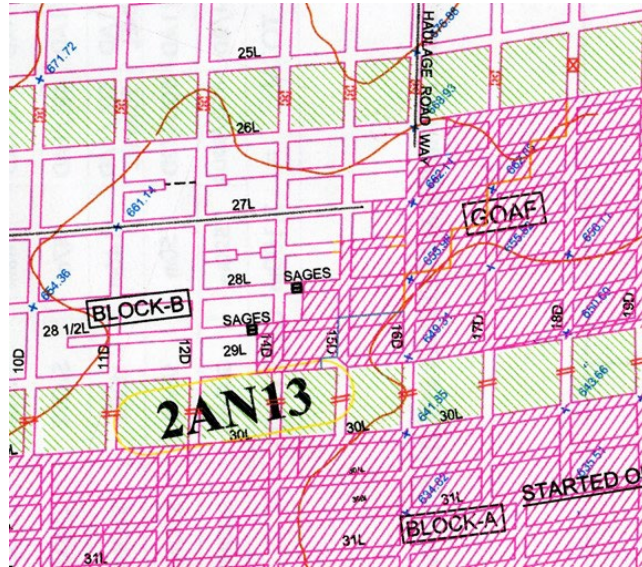


Figure 3: Schematic diagram of working area (Panel 2AN13) showing extraction of coal in coal mines⁴

2. RELATED WORK

There have been various proposals and developments for advancement in automated technologies using AI in the mining industry for improving efficiency, hazardous exposure reduction, and improving safety. The research described in Hyder et. al, 2019 shows that Artificial Intelligence can be incorporated in mineral processing, hazard detection, exploration, and accident analysis. Coal excavation automation [Nalbantov et. al., 2010] wherein the shearer loader has been integrated with a pattern recognition system which detects the correct position of the coal to be excavated, hence automating the process of mining. The pattern recognition system consists of two mechanisms, each being noise-elimination mechanism and coal-bed boundary detector. This data-mining approach is based on image recognition.

Safety being the primary goal, Matloob et. al., 2021 describes how AI and machine learning can provide an effective risk assessment to highlight probable hazards during mining and with other economic benefits. It prescribes formal and informal approaches for risk assessment which includes methods such as fault tree analysis, hazard identification and ranking, workplace risk assessment, and control but does not showcase an approach on how data preprocessing and analysis can be automated through which useful insights can be extracted. Janusz et. al., 2017 proposes an approach on how to assess and predict hazardous seismic events in coal mines which compares conventional methods of seismic and seismoacoustic to an ensemble of diverse regression models called the seismic hazard assessment model wherein batch data is aggregated and split into static and time-series data. Features are extracted from the time series data and presented as input to different regression models such as Support Vector Machines (SVM), regression trees, and generalized linear models but do not propose a solution for automation of data processing and analytics for improving mines safety.

Research undertaken in Fu et. al., 2020 explains how deep learning models such as convolutional neural networks, long short-term memory, deep belief networks, and deep reinforcement learning to carry out operations

⁴ Courtesy: RK7 Incline Mine, Srirampur Area, SCCL, Telangana, India

such as drilling, haulage, blasting, mineral processing, and communication but does not account for the bottleneck due to unlabeled image data for the development of sensors and deep learning models.

Although developments and proposals have been made in the field of automating the operation of mining, however, these approaches do not account for the automation of data analysis, processing, and proficient communication of the data recorded during mining operations carried out in underground coal mines. Coal excavation automation automates the process of discovering coal-bed boundaries through image processing but does not account for how human labor can be minimized as well as does not describe an approach on how the recorded data help deduce other mechanical properties during mining operations. Analysis methods can be divided into statistical analysis [Cunningham et. al., 1995] and model-based diagnosis. Both categories of analytics depend on the actual data recorded from XR5⁵ data loggers installed in the SAGES. Whilst model-based analytics can provide an intuition of behavioral variation of data, statistical methods on the other hand are better for anomaly detection and data wrangling.

3. BACKGROUND CONCEPT

3.1 Proposed model of mines IoT System [A. Singh et. al. /Ad Hoc Networks 78 (2018) 115-129]

SAGES are used in the goaf area in the underground coal mines to support the unstable roof of the location where coal has been freshly extracted with the help of depillaring operations. As shown in Figure 4, the WSN (Wireless Sensors Network) repeaters and the underground base station receive the status of the SAGES sensors which is transferred to the base station. With the help of IoT, results of workmen health monitoring, mine environment health monitoring, strata behavior monitoring is recorded and transferred to the base station subsequently, which are then transferred to the cloud applications assuming ground base station has access to the Internet. This data is preprocessed using ETL (Extract, Transform and Load) data pipeline in the cloud system. The preprocessed data is used as an input for statistical analysis, big data mining, and machine learning.

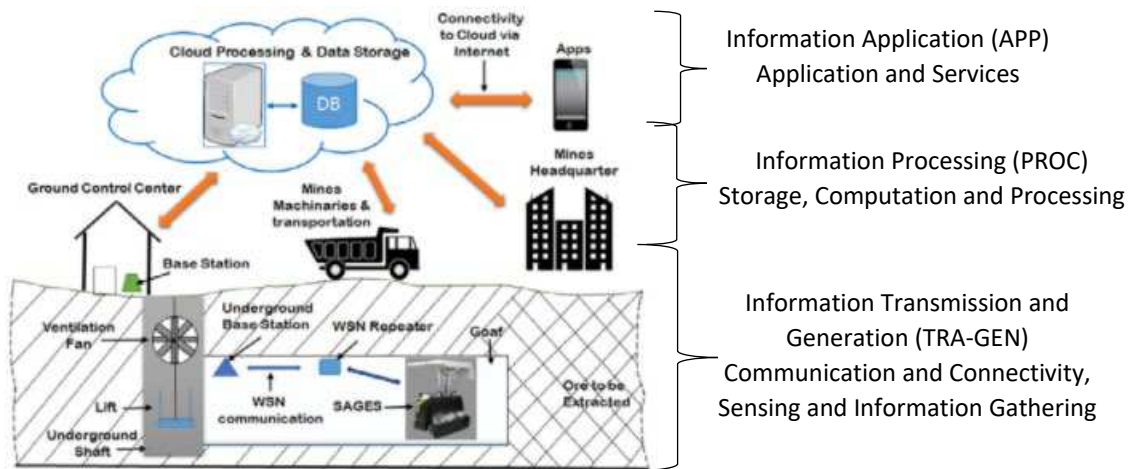


Figure 4: Underground Mine Communication IoT infrastructure mapped to IoT building blocks (A. Singh et. al)

By embedding machine learning with IoT systems, the processing of information (PROC) extracted from the SAGES sensors and data loggers will be more efficient and faster. The absence of artificial intelligence generates processing constraints as IoT nodes has limited processing power due to which occurrence of resource exhaustion attack is inevitable which in turn degrades the quality of IoT network. The data analysis structure will contain automated online monitoring, data transmission, and processing as well as analysis and reporting of production forecasting, thus helping in mine planning and design. By adapting and developing technologies for safe mining and proficient data processing and analysis, we can proceed towards the smart-SAGES goal.

⁵ <https://www.pace-sci.com/data-loggers-xr5.htm>

3.2 Data logging (acquisition) in SAGES

The SAGES is having inbuilt rock monitoring sensors and data loggers. The data logger is programmed for logging the sensor data at every minute. One of the sensors is a pressure transmitter fitted in the hydraulic legs of the support. The pressure sensors respond to the strata load building upon the canopy of the SAGES due to the convergence of the roof strata. The stability of the roof very much depends on its convergence. The layers of the strata above the roof separate from the main roof and put excessive load on the support when the convergence of the roof is increased. Ultimately, they fail in the face and working zones causing loss of the machine, men, and coal. Therefore, a draw wire type convergence sensor is fixed between the base and canopy of the SAGES. The data monitoring system continuously monitors the strata load and convergence at the working faces. It generates a warning when convergence, load, and combined load-convergence combined exceeds a prescribed limit for safe working. This monitoring system helps in-time decisions to avoid any eventuality and ensure a safe working place.

The new data logger is integrated and functional in the SAGES. It is deployed in the mines packed inside steel cover for the field trial (see Figure 5). Load and convergence observations of every minute are stored in data loggers which are afterward analyzed manually for the progressive development and maintenance of SAGES. It is an example of automation and information technology in mines. By adopting IoT (Atzori et. al., 2010) based information and communication system (A. Singh et. al., 2018) and merging digital technologies like artificial intelligence (machine learning), we can implement an efficient communication channel and data collection and analysis process to enable proficient decision making, hence increasing underground safety in mines.



Figure 5: XR5 Data Logger packed inside steel cover for the field trial

We are using the data from the XR5 data loggers that were installed in the initial field trial before the newly designed and developed IoT-enabled data logger (A. Singh et. al., 2018) for SAGES. Taking a close look at these XR5 data logger files, we find that there may exist an intercorrelation within these observations which could not be extracted manually. To deal with this issue, we formulate the predictive analysis task as an unsupervised and supervised machine learning problem and propose an approach using Linear Estimators, Gradient Boosting, and DBSCAN1D (Meglicki et. al, 2021) algorithms. Through the related works (see Section 2) and Data Analysis (see Section 4.2), we perceive that the proposed approach could precisely distinguish and calculate the load at withdrawal, convergence, and duration of yielding of four deployed SAGES 1, 2, 3, and 4 respectively and also develop a proficient and seamless processing and analysis of data with IoT.

3.3 Development Strategy

Generalized Linear Model (GLM) [Müller et. al., 2004] is a statistical method to calculate the relation between continuous and/or categorical variables. It is a flexible generalization of ordinary linear regression for analysis [Kumari et. al., 2018], thus allowing response variables to have an abnormal error distribution. GLM has been used for analytical purposes in various domains such as field research, operations research, and exploration as described in Wignall et. al., 1987.

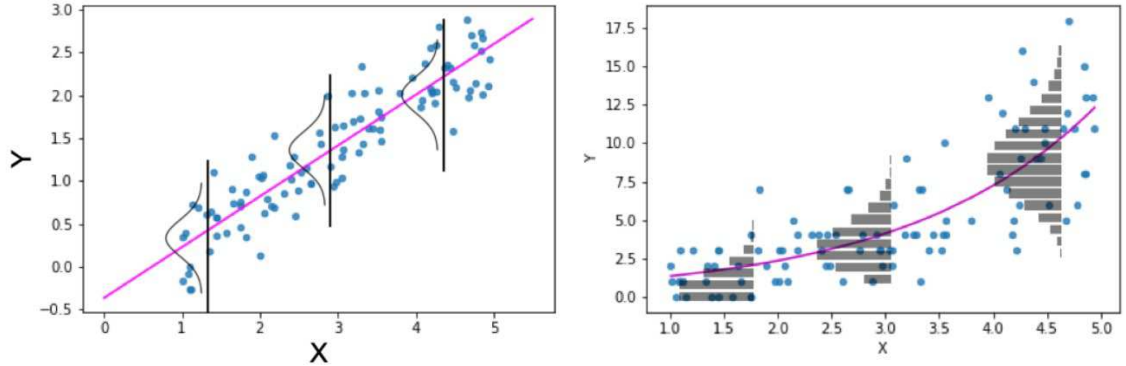


Figure 6: Representation of difference between (a) Linear Regression and (b) Generalized Linear Model

Data mining is a process that is used to extract information, patterns, and trends from a large dataset for analytical and decision-making purposes. As the data loggers installed in the SAGES provide us with unlabeled data for the strata load on Leg-1 and Leg-2 and absolute convergence recorded during deployment of the SAGES in underground coal mines, we employ unsupervised clustering algorithms to generate groups into a given number of clusters in presence of noise to mine information and trend which is followed in the load on the legs and convergence value during depillaring operations. Unsupervised methods such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [Babur et. al., 2015] has been used for unsupervised clustering. By using Euclidean distance, DBSCAN (see Figure 7) groups together points in an n-dimensional space to find associations and structures in data.

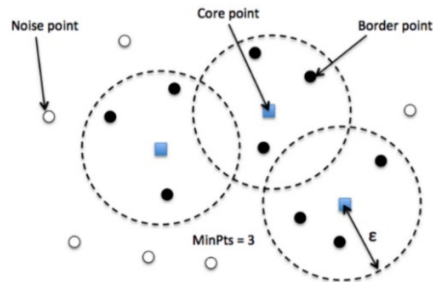


Figure 7: Segmented representation of DBSCAN algorithm

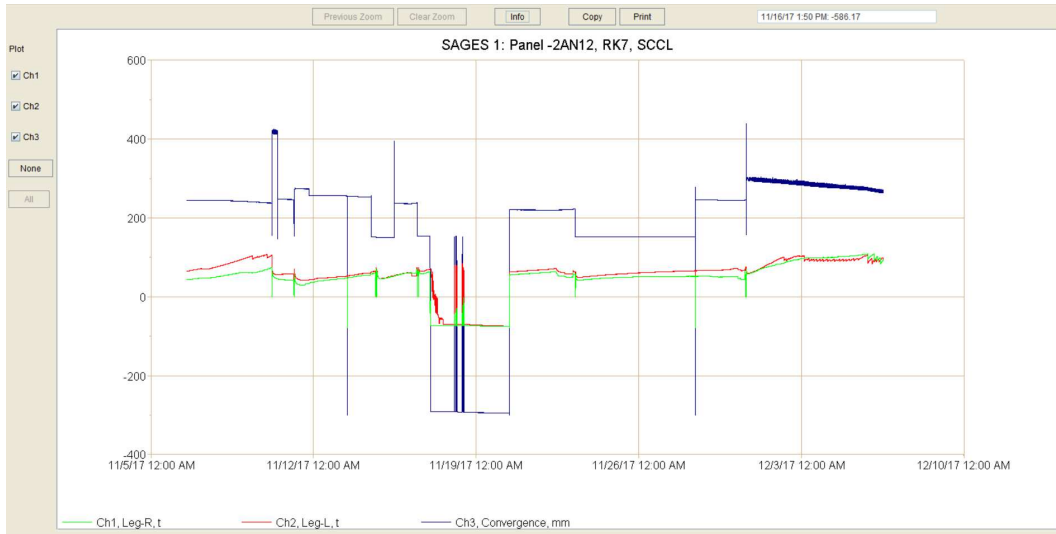


Figure 8: Representation of data acquired using XR5 data loggers installed in SAGES

With the facilitation of work carried out in A. Singh et. al., 2018 and Singh et. al., 2016, the recorded data was monitored using LogXR⁶ software (see Figure 8), plotting the data recorded in the XR5 data logger during deployment of the SAGES. Spatial clustering depends on the location, prominence as well as feature size of the points to be clustered. Other unsupervised algorithms, for instance, K-Means [Na et. al., 2010] cannot be used for analysis as clusters formed using K-Means must have the same feature size, absence of noise and it cannot work with points which may form clusters with a varying density as well as anomalous points will be assigned to the regular cluster which can create false trends. We can view from Figure 9, how the DBSCAN algorithm segregates noise and outliers in the data whereas the K-Means clustering algorithm does not and considers outliers/noise as “normal” data. Thus, working with spatial clustering in 1-D space will use an enhanced and extended version of the DBSCAN algorithm, that is, the DBSCAN1D algorithm [Meglicki et. al., 2021]. DBSCAN1D runs in $O(N)^7$ time instead of $O(N \log N)$ and does not depend on the ϵ -neighborhoods.

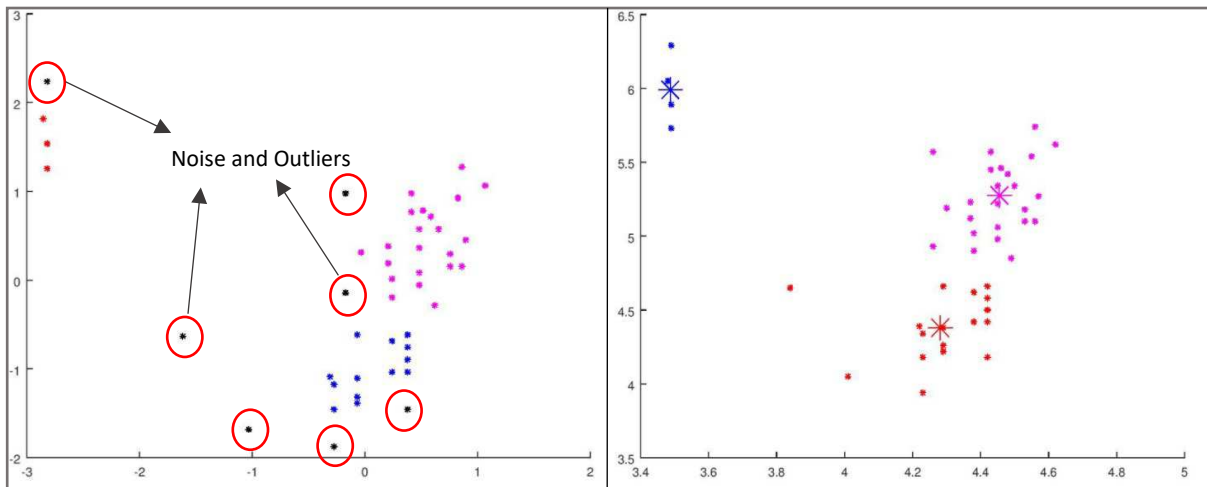


Figure 9: Representation of difference between (a) DBSCAN and (b) K-Means

4. METHODS

⁶ LogXR is a software used to plot data acquired in XR5 data loggers - <https://www.pace-sci.com/LogXR.htm>

⁷ $O(N)$ represents the time complexity of the algorithm, similarly $O(N \log N)$

4.1 Quantitative Methodology

The dataset has been collected from XR5 data loggers installed inside SAGES. These data files contain information about the strata load on Leg-1 and Leg-2, in ton being applied on the SAGES and convergence, in mm which was caused due to variation in load and other environmental factors. The acquired data consists of files from August 4, 2017, to January 25, 2018, for SAGES-1 & SAGES-2 and November 7, 2017, to February 15, 2018, for SAGES-3 and SAGES-4. The data consists of observations that have been procured every minute for the given duration. Features such as set load, withdrawal load, set-date, withdrawal date, location (face, block, level, slice, dip (categorical)), convergence, roof exposure and blasted holes were extracted at the time of deployment in underground coal mines and from the monitored raw data in XR5 data loggers grouped for each slice at every location in the coal mine which has been used to predict convergence, in *mm* and strata load at withdrawal on the SAGES.

4.2 XR5 Logger Data Analysis

We analyze the data obtained from XR5 data logger files extracted from SAGES through JointGrid plots. JointGrid plots are used for bivariate analysis and marginal univariate distribution analysis of the feature variables. From Figure 10, we observe that there is an approximately linear relationship between strata load on Leg-1 and Leg-2 of SAGES. Looking at the same plots for strata load on leg vs. convergence (mm), we can conclude that although we can see an increasing relation between the two, there is not much linearity and the value of absolute convergence, in *mm* varies irregularly with high values of strata load on Leg-1 and Leg-2 of SAGES.

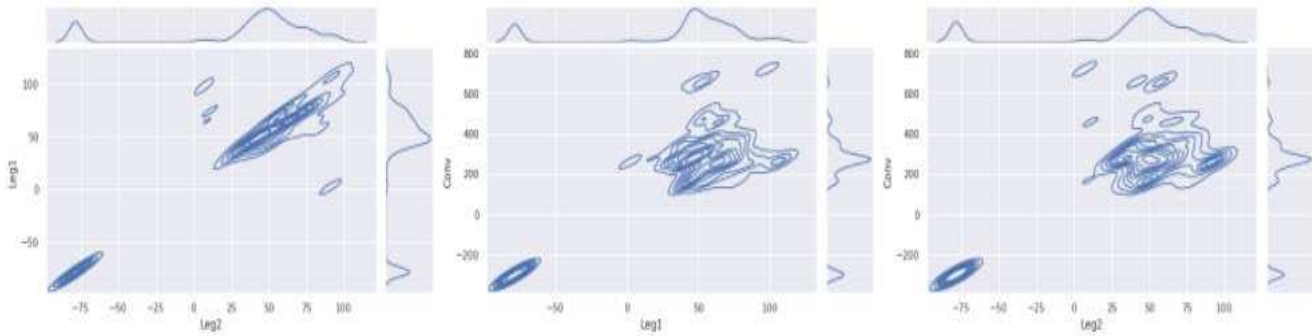


Figure 10: JointGrid plots for SAGES-2

From the residual⁸ plots in Figure 11, we can observe that linear regression models are not a good fit when there is a transition of SAGES among slices, where the values of strata load and convergence tend to deviate significantly from the $y=0$ line. Therefore, we try to fit a Generalized Linear Model (GLM) on the data points through we can observe the regression lines and it fits with our conclusions made from the JointGrid plots in Figure 10. GLM generalizes linear regression by allowing the linear model to relate to the response variable via a link function⁹ g and by allowing the magnitude of the variance of each measurement to be a function of its predicted value, precisely, GLM is determined by link function g and the variance function $v(\mu)$ and x . The architecture/network structure of the model can be understood from the below equations.

$$Y_i \sim F_{EDM}(\cdot | \vartheta, \phi, w_i)$$

$$\mu_i \leftarrow E[Y_i | x_i] \leftarrow g^{-1}(x_i' \beta)$$

$$f_{EDM}(y | \vartheta, \phi, w) \leftarrow c(y, \phi, w) \exp((y\vartheta - b(\vartheta)) / w)$$

$$\text{Var}[Y_i | x_i] \leftarrow (\phi / w_i) v(\mu_i)$$

$$v(\mu) \leftarrow b''(\vartheta(\mu))$$

⁸ Residual plot represents how much a regression line vertically misses a data point

⁹ It maps a non-linear relationship in a data onto a linear one

GLM network structure

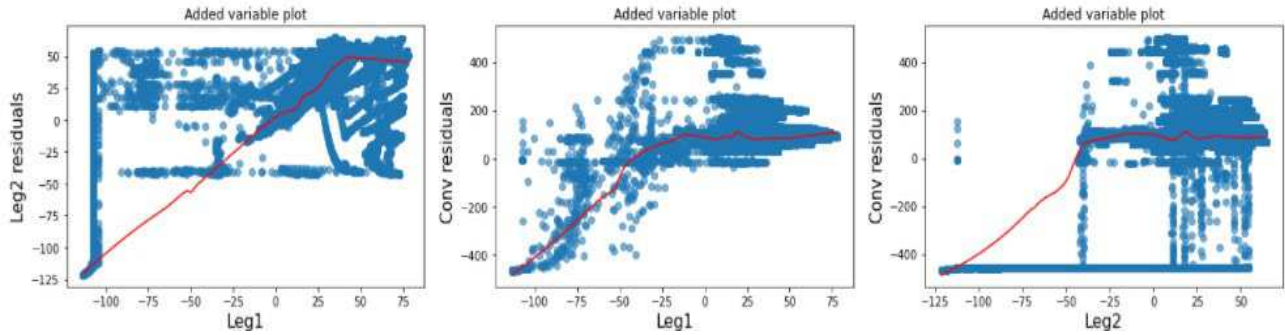


Figure 11: GLM Residual plots for SAGES-1

Depillaring operations using SAGES-1 and SAGES-2 were carried out for six slices, whereas using SAGES-3 and SAGES-4, the operations were carried out for four slices. By employing the K-Means clustering algorithm on raw data to cluster strata load on Leg-1 and Leg-2 as shown in Figure 12, we can observe that the K-Means algorithm cluster strata load values in the XR5 data loggers in correlation to the number of slices for which the particular SAGES is used. The number of clusters is equal to the number of slices for which SAGES were used. We can also observe that the K-Means clustering algorithm spatially generates clusters but also considers the noise and outliers, which in the case of DBSCAN cannot be observed (see Section 3.3).

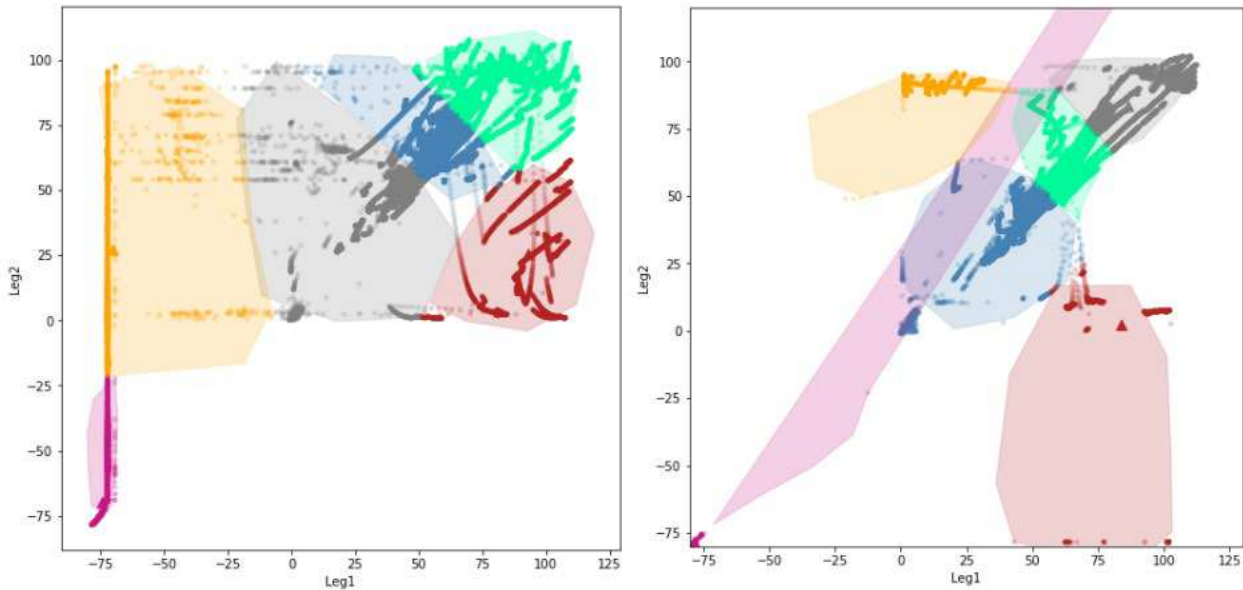


Figure 12: Representation of spatial clustering of XR5 logger data using K-Means clustering

We check the continuous variance of the values of load on Leg-1 and Leg-2 and compare it with the variation of convergence in the SAGES. Figure 13 explains the variability of the features which concludes that the similarity in the variation of each feature is ~95% to others for all days and months of deployment of SAGES and thus, we use a single unsupervised machine learning algorithm to calculate the duration of yielding in SAGES. The similarity in variation in all the three properties arises due to deployment at the same location with the depillaring driving factor being the same with the strata load on Leg-1 and Leg-2 having a causal effect on convergence.

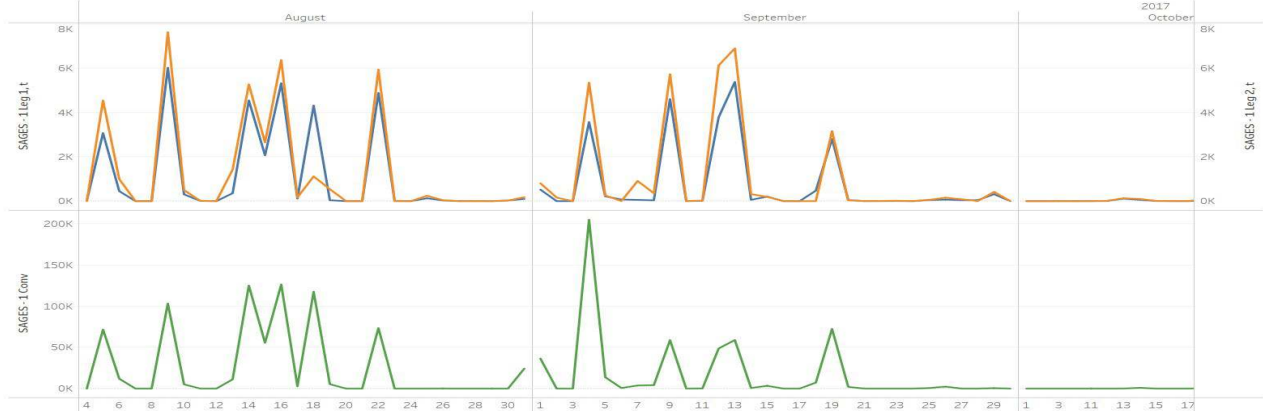


Figure 13: Comparison of variation of Strata Load on Leg-1 and Leg-2 with Convergence

4.3 Base Architecture

The base architecture is used to extract task-specific features which are passed on to task-specific modules during implementation. For predicting withdrawal load and convergence, we extracted block and SAGES-wise data. Since SAGES-1 and SAGES-2 are deployed at the same location at rise and dip sides respectively, set-load, convergence, withdrawal load, face value will be affected by each other's influence. Percent extraction of the rib, %, and roof and side condition were not included in the prediction for both as they contained only a constant numerical value. Support yielding duration, *in hrs*, was not used as a feature due to the unavailability of proper values. A similar approach was followed for predicting target features for SAGES-3 & SAGES-4. The final data for predictive analysis of convergence in SAGES has a shape of (62, 13) for both rise and dip sides. The data for predictive analysis of strata load at withdrawal on SAGES has a shape of (62, 15). We will be using Conditional Generative Adversarial Networks (CTGAN¹⁰) (Xu et. al., 2019) for modeling the probability distribution of rows in tabular format and generating realistic synthetic data of 1000 rows.

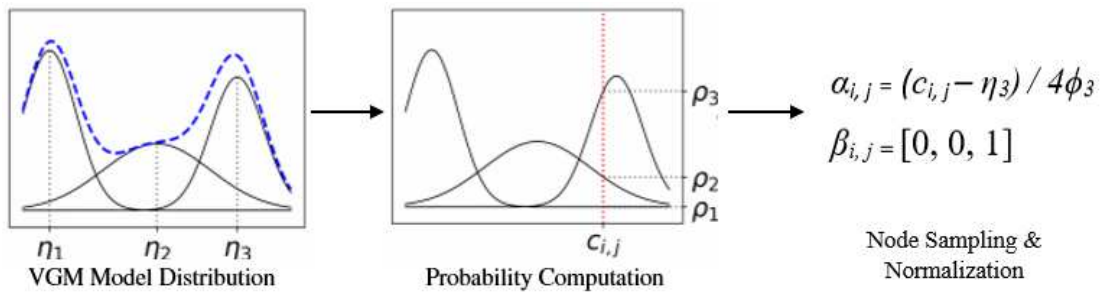


Figure 14: Mode-Specific Normalization [Xu et. al., 2019]

Each continuous value is represented using a one-hot vector indicating the mode and a normalized scalar indicating the value within the model. Figure 14 shows an example of Mode Specific Normalization (MSN) where η_1 , η_2 , and η_3 are the three modes which is leveraged to deal with columns having non-Gaussian and multimodal distributions. For each column Variational Gaussian Mixture (VGM) model estimates the number of modes to fit on the Gaussian mixture. Probability from each node is calculated and the highest probability density is sampled to normalize the value. Values in a column are represented as a one-hot vector for each column processed independently, which indicates the mode.

For handling categorical values in the dataset, CTGAN introduces training by sampling and conditional generators approach to resample the training data and deal with imbalance labels so that all the categorical features get optimum

¹⁰ CTGAN is a collection of Deep Learning based Synthetic Data Generators for single table data, which can learn from real data and generate synthetic clones with high fidelity

chance to be included in the sample from which GAN learns. To capture the linear and non-linear relationships and complexities in the data, CTGAN uses generator and critic models to learn the conditional distribution of the real data which contain a fully connected hidden network with activation function of LeakyReLU to capture correlations and non-linear interactions.

$$\begin{aligned}
 h_0 &\leftarrow z \oplus \text{cond} \\
 h_1 &\leftarrow h_0 \oplus \text{ReLU}(\text{BN}(f^0(h_0))) \\
 h_2 &\leftarrow h_1 \oplus \text{ReLU}(\text{BN}(f^1(h_1))) \\
 D_l &\leftarrow \text{Gumbel}_{0,2}(g^D_l(h_2)), l = 1, 2 \dots ND \\
 C_j &\leftarrow \tanh g^C_j(h_2), j = 1, 2 \dots NC
 \end{aligned}$$

CTGAN network structure

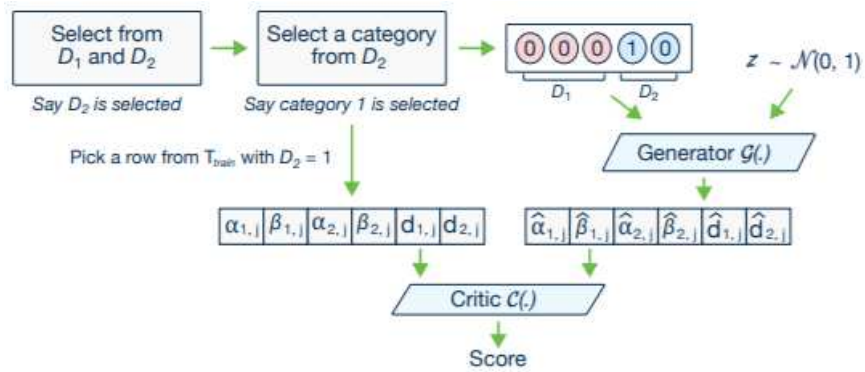


Figure 15: CTGAN Model [Xu et. al., 2019]

The features from data recorded from the XR5 data logger installed in SAGES are extracted and the timestamp objects are converted into DateTime features. The complete non-uniform time series data is resampled into a 1-minute window for which the most recent values were chosen as features that were returned as float. The variation of the strata load on Leg-1 and Leg-2 of the SAGES mimics that of the signal as shown in Figure 16. Peaks of these signals were marked by finds all local maxima by a simple comparison of neighboring values in a 1-D array. A subset of peaks can be selected by filtering their prominence, distance, height, and width. The parameters below were found by manual trial and experimentation. The parameters are as follows:

- Distance is the required minimum horizontal distance in the sample between the neighboring peaks in the signal with a value greater than or equal to 1.
- Width is the required minimum width of the peaks in the sample of the signal with numerical values or a 2-element sequence or an array of the dimension of the signal.
- Prominence is the required minimum prominence of the peaks in the signal sample filtering out the peaks whose prominence is not in the given range. Its values can be numerical values or a 2-element sequence or an array of the dimension of the signal.
- Height is the required minimum height of the peaks in the signal sample with numerical values or a 2-element sequence or an array of the dimension of the signal.

Distance	1
Width	7
Prominence	{P = (0.2, 5) x = Load}
Height	Mean

Table-1: Parameters to find peaks

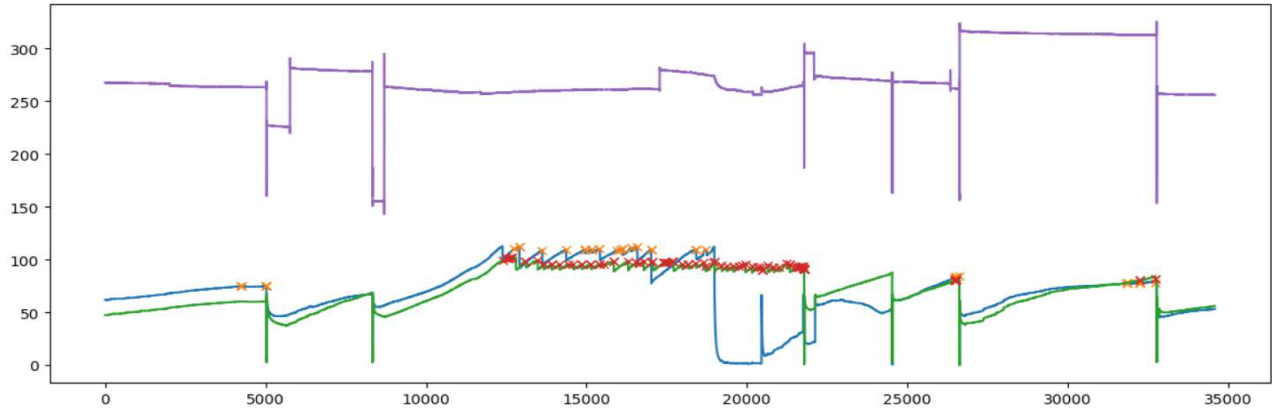


Figure 16: Highlighted prominent peaks

4.3.1 Conditional – GAN (CTGAN) similarity statistics between real and synthetic data

The reflection on the statistical scores of the CTGAN generated synthetic data has been used as the pivot point for the similarity between real data and synthetic data. Similarity score gives the aggregate of metrics mentioned in Table-2. We can observe that basic statistics and mean correlation between the two datasets are significant with a similarity score equal to 0.8220.

Basic Statistics	0.9774
Correlation Column correlations	0.7781
Mean correlation between real and synthetic data	0.9462
1 – MAPE ¹¹ Estimator results	0.5864
Similarity Score	0.8220

Table-2: Similarity statistics between real and CTGAN resampled data

From the plots representing the similarity between CTGAN generated synthetic data and real data (see Figure 17), we can observe the similarity in the distribution of variables such as strata load at withdrawal at rise side, Face value, in m , setting load at dip side and dip value. Dip value, setting load at dip side and strata load at withdrawal at rise side in both original and synthetic resampled data follow a remarkably similar distribution. Face value, in m in both datasets, have distinguishable features which result in a decrease of similarity score to 0.8220. Through the observations (see Table-2 and Figure 17), we can validate the usage of CTGAN resampled data.

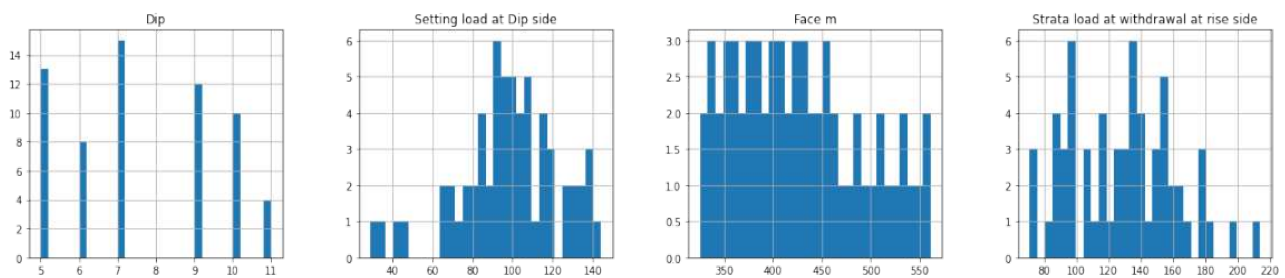


Figure 17(i): Plots representing distribution of (a) Dip (b) Setting load at Dip side (c) Face, m (d) Strata load at Withdrawal at Rise side for original data

¹¹ Mean Absolute Percentage Error

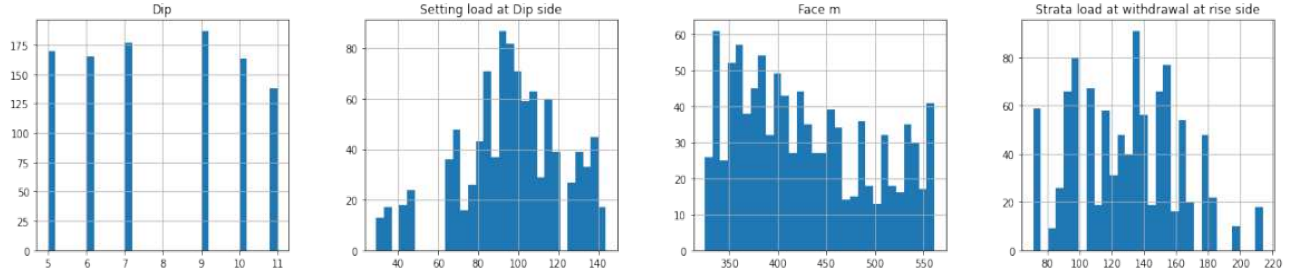


Figure 17(ii): Plots representing distribution of (a) Dip (b) Setting load at Dip side (c) Face, m (d) Strata load at Withdrawal at Rise side for CTGAN generated data

5. IMPLEMENTATION OF AI MODEL FOR SAGES

The development of the three separate modules are based on three defined objectives objective (see Section 1):

- Strata Load at withdrawal on SAGES
- Convergence in SAGES
- Duration of yielding in SAGES

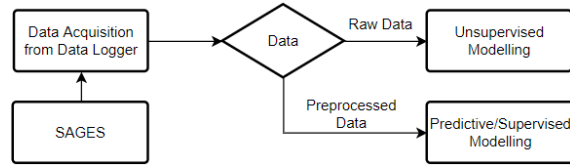


Figure 18: Proposed System

5.1 Load at withdrawal on SAGES

We segment the CTGAN resampled dataset of shape (1000, 15) to predict two variables, Strata Load at Withdrawal at Rise Side (includes SAGES-1 and SAGES-3) and Strata Load at Withdrawal at Dip Side (includes SAGES-2 and SAGES-4) which was passed on to Generalized Linear Estimator tuned manually by minimizing the RMSE (Root Mean Squared Error). We train a Generalized Linear estimator with a Poisson distribution for five-folds as the dataset contains non-negative responses. The basic network of a Generalized Linear Estimator is the same as shown in the GLM network structure (see Section-3.3 and Section-4.2).

By using the Poisson distribution family with GLE, the model defines the dependency between the response and the covariates present in the data which is fit on the data by maximizing the penalized likelihood, represented by

$$\text{Penalization} \leftarrow \frac{1}{2}(1 - \alpha)\|\beta\|_2^2.$$

The network structure of Poisson regression used in Generalized Linear Estimators can be expressed as:

$$\hat{y} = e^{x^T \beta + \beta_0}$$

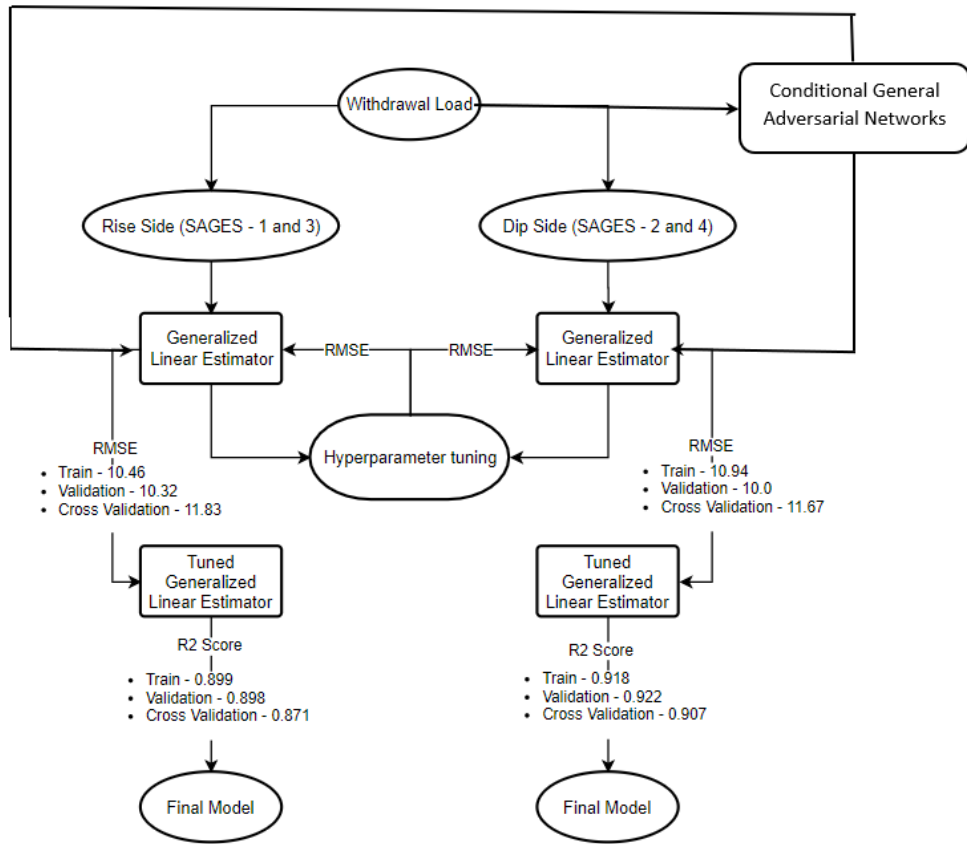
$$\max_{\beta, \beta_0} \frac{1}{N} \sum_{i=1}^N (y_i(x_i^T \beta + \beta_0) - e^{x_i^T \beta + \beta_0}) - \lambda(\alpha\|\beta\|_1 + \frac{1}{2}(1 - \alpha)\|\beta\|_2^2)$$

$$D = -2 \sum_{i=1}^N (y_i \log(y_i/\hat{y}_i) - (y_i - \hat{y}_i))$$

Poisson Regression network

Analysis of strata load on Leg-1 and leg-2 on SAGES show that one leg has a causal effect on the other, which conveys significant linearity (see Figure 10). From the GLM residual plots (see Figure 11), it is understandable that linear regression as a model cannot be used, instead, a Generalized Linear Estimator model has a substantially improved fit between Load on Leg-1 and Leg-2. We split the resampled data into a train to test ratio of 85:15 which

gives us immense data for cross-validation¹² training as well as for testing. By splitting the data into five-folds, we reduce overfitting in the data while training. We train two separate hyper tuned models each for Rise Side and Dip Side of deployment to contain the data's interactivity.



Withdrawal Load at Rise side	
Family	Poisson
Link	Log
Regularization	Ridge (lambda = 0.022201)
Total predictors	50
Active Predictors	50
Iterations	4
Withdrawal Load at Dip side	
Family	Poisson
Link	Log
Regularization	Ridge (lambda = 0.02822)
Total predictors	50

¹² Cross Validation is a technique to ensure robustness of the machine learning model by splitting the data into subsamples for specified iterations, also known as Rotation Estimation or Out-of-Sample testing

Active Predictors	50
Iterations	4

Table-3: Generalized Linear Estimator parameters for predictive analysis of Load at Withdrawal

To minimize the risk of overfitting, we use regularization techniques. In ridge regularization, a small penalty is added to alter the cost function which is equivalent to the square of the magnitude of the coefficients. It uses a shrinkage parameter λ that minimizes their value.

$$\min \left(\sum_{i=1}^n 1 * \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ji} \right)^2 + \lambda \sum_{i=1}^n \beta_j^2 \right)$$

; where p is the total number of predictors

Ridge Regularization

Assumptions –

- It was assumed that set load, convergence, load at withdrawal are correlated and had interactivity strength.
- Similar interactivity strength was observed in set date and withdrawal date at both dip and rise sides of underground coal mines.
- Null values in Blasting to Induce Caving were assumed to be equal to 0.

5.2 Convergence in SAGES

For the predictive analysis of Convergence, at both Rise and Dip sides, we segment the data with different target features and create a train-test splitting in a ratio of 85:15, and three separate models were incorporated on this data – Generalized Linear Estimator, Gradient Boosting Estimator with default parameters and Gradient Boosting Estimator with hypertuned parameters – the outputs of which were stacked to create a final ensemble model.

- **Generalized Linear Estimator model**

Generalized Linear Estimators used for the predictive analysis of Convergence in SAGES has been trained for six-folds and follow the same concept of usage, architecture, and network (see Section-5.1), but with Elastic Net regularization as that of GLE used for the predictive analysis of Load at Withdrawal on SAGES.

Convergence at Rise side	
Family	Poisson
Link	Log
Regularization	Elastic Net (alpha = 0.25; lambda = 0.5736)
Total predictors	47
Active Predictors	13
Iterations	4

Convergence at Dip side	
Family	Poisson
Link	Log
Regularization	Elastic Net (alpha = 0.25; lambda = 1.5925)
Total predictors	47
Active Predictors	7
Iterations	2

Table-4: Generalized Linear Estimator parameters for predictive analysis of Convergence

Elastic Net regularization [Zou et. al., 2004] is used to create a convex combination of Ridge and Lasso regularization. During regularization, the L1 penalty forms a sparse model whereas the quadratic part of the penalty stabilizes L1 as well as eliminates the limit of variable selection and promotes grouping effect which easily identifies variables using correlation, subsequently enhancing the sampling procedure. We can describe the geometry of Elastic Net in a 2-dimensional space with $\lambda_1 = \lambda_2$. The geometry of the elastic net explains the grouping and the sparsity of L1 penalty.

$$\hat{\beta} \equiv \underset{\beta}{\operatorname{argmin}} (\|y - X\beta\|^2 + \lambda_2 \|\beta\|^2 + \lambda_1 \|\beta\|_1)$$

$$J(\beta) = \lambda_2 \|\beta\|^2 + \lambda_1 \|\beta\|_1$$

Elastic Net Regularization and Penalty

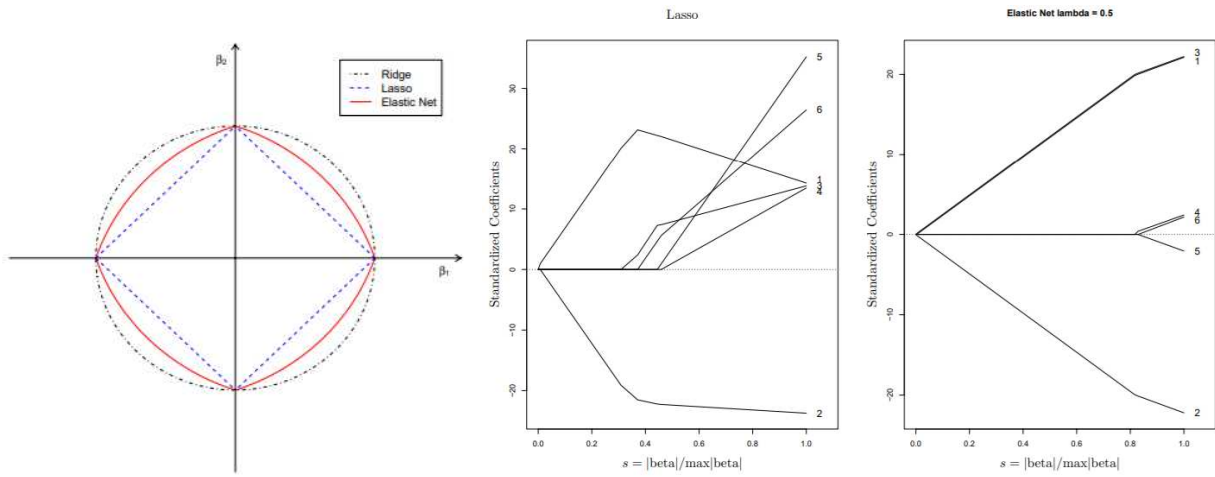


Figure 20: (a) Geometry of Elastic Net (b) Comparison of Lasso and Elastic Net regularization [Zou et. al., 2004]

• **Gradient Boosting Estimator model**

We train this model for the predictive analysis of Convergence in SAGES with cross-validation of six-folds, as the data contains highly unscaled information, with least linearity as shown through JointGrid plots for Load on legs vs. Convergence in SAGES. It is based on the idea of Gradient Boosting Modelling (GBM).

$$f(x) = f^{(M)}(x) = \sum_{m=0}^M f_m(x) = f(x) + \sum_{m=1}^M \eta \rho_m \phi_m(x)$$

GBM resulting model equation

Boosting, also known as additive modeling is a method of combining multiple simple or weak models to generate a single composite model, which finally uses gradient descent to minimize the loss. Decision trees are used as weak learners by transforming the data into a tree representation. Each node and leaf of the tree represents the attribute and the label respectively. Cumulatively, each weak predictor corrects its predecessor's error. The ensemble consists of N trees. *Tree_1* is trained using the feature matrix X and the labels y . The predictions labeled $y1(\hat{hat})$ are used to determine the training set residual errors $r1$. *Tree_2* is then trained using the feature matrix X and the residual errors $r1$ of *Tree_1* as labels. The predicted results $r1(\hat{hat})$ are then used to determine the residual $r2$. The process is repeated until all the N trees forming the ensemble are trained (see Figure 21).

Convergence at Dip side		Convergence at Rise side	
Number of Trees	100	Number of Trees	100
Number of Internal Trees	100	Number of Internal Trees	100
Model Size (Bytes)	11384	Model Size (Bytes)	11150
Minimum Depth	2	Minimum Depth	2
Maximum Depth	4	Maximum Depth	4
Mean Depth	3.34	Mean Depth	2.86
Minimum Leaves	4	Minimum Leaves	4
Maximum Leaves	5	Maximum Leaves	5
Mean Leaves	4.37	Mean Leaves	4.2

Table 5: GBM parameters

- Hypertuned Gradient Boosting Estimator model**

This model has the same architecture as the Gradient Boosting Estimator used with default parameters and has been trained for six-folds. Hyperparameter tuning is an essential step as it directly impacts the performance of the machine learning model. By manual hypertuning the model hyperparameters, we can observe the behavior of individual parameters. The hyperparameters used in this particular model are as follows:

Stopping Rounds	25
Stopping Metric	RMSE
Rate of column sampling	0.65
Sample Rate	0.65
Distribution	Poisson
Number of folds	6

Table-6: Hyper tuning of GBM model

Model Parameters (GBM)	GBM Parameter values	
	Convergence at Rise Side	Convergence at Dip side
Number of Trees	61	71
Number of Internal Trees	61	71
Model Size (Bytes)	5846	6857
Minimum Depth	1	1
Maximum Depth	3	3
Mean Depth	1.95082	1.985915
Minimum Leaves	2	2
Maximum Leaves	4	4
Mean Leaves	2.95082	2.985915

Table-7: GBM parameters for predictive analysis of Convergence

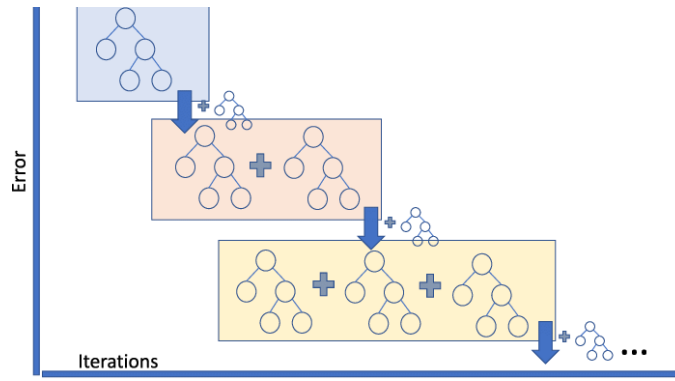


Figure 21: Schematic representation of Gradient Boosting regression in regards to algorithm iterations [Baturynska et. al., 2021]

- **Ensembling**

Separate results for convergence in SAGES at dip side and rise side are predicted using each model. Predictions are stacked horizontally for the dip side and rise side respectively to create two datasets with target variable as Convergence for the respective segments.

To develop an ensemble composite model, we use Random Forest Regressor and XGBoost regressor for Convergence at rise side and dip side respectively. Random Forest [Ali et. al., 2012] is a meta-estimator¹³ that uses bagging¹⁴ technique in which the models are developed parallelly with no interaction between the individual weak learners. In the case of regression, it outputs the mean prediction of the individual trees. It adds additional randomness to the model by searching for the best feature among a subset of features while splitting a node. These models were chosen based on experimentation.

By using the base structure of gradient boosting trees, XGBoost [Chen et. al., 2016] is a boosting technique that minimizes a regularized (L1 and L2) objective function that combines a convex loss function and a penalty term for model complexity i.e., the regression tree functions. XGBoost approaches the process of sequential tree building using parallelized implementation due to the interchangeable nature of loops while building the base learners. This nesting of loops limits parallelization because the inner loop must be executed before the execution of the outer loop. Therefore, the order of loops is interchanged using the initialization and sorting of parallel threads to improve run-time.

	glm	gbm	tuned_gbm	real		glm	gbm	tuned_gbm	real
0	3.675279	2.293820	2.717159	2	0	5.761838	5.067831	5.610388	5
1	7.720132	10.414952	7.893065	11	1	11.636723	22.783902	21.335473	23
2	4.690973	3.544496	3.695012	3	2	7.455836	13.947068	12.859149	14
3	4.206223	5.983175	5.252626	6	3	6.267170	2.236805	2.910139	1
4	4.470981	4.037067	4.163822	4	4	6.654325	3.087079	2.536840	3

Table-8: Stacked predictions and the target variable

¹³ Meta-estimator is a parameter which takes another estimator as a parameter

¹⁴ Bagging is a bootstrap aggregation method used for ensemble learning to reduce variance in a noisy dataset

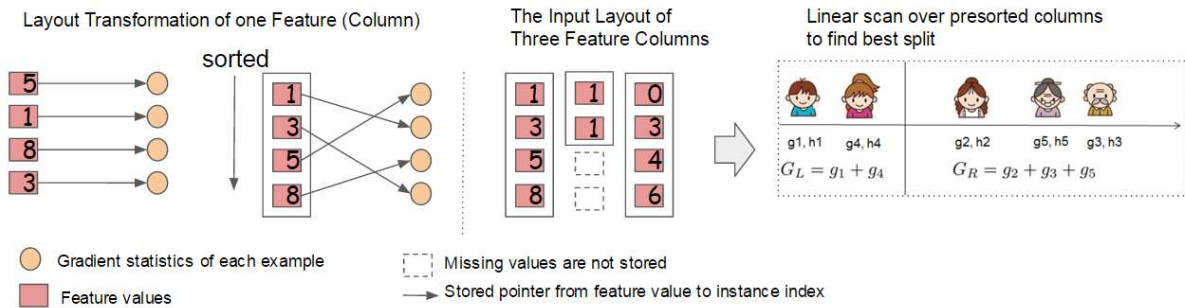


Figure 22: Block Structure for parallel learning [Chen et. al., 2016]

To implement the ensemble models, the data is scaled such that the mean is equal to 0 and the standard deviation is equal to 1. Feature scaling is used to normalize the range of independent variables used for predictive analysis of the data. The scaled data is fit into the models and the respective RMSE scores are compared.

$$z = (x - \mu) / \sigma$$

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

Standardization

base_score	booster	learning_rate	objective	n_estimators
0.5	gbtree	0.1	reg:linear	100
max_depth	reg_lambda	reg_alpha	colsample_bytree	subsample
3	1	0	1	1

Table-9: XGBoost parameters for ensembling

Assumptions –

- Through data analysis, it was found that set date and withdrawal date did not play an important role in the prediction of convergence, *mm*, and thus, were not included in the predictor features.
- It was assumed that Load at withdrawal at dip and rise side and set load at dip and rise side are correlated and had interactivity strength.
- Null values in Blasting to Induce Caving were assumed to be equal to 0.

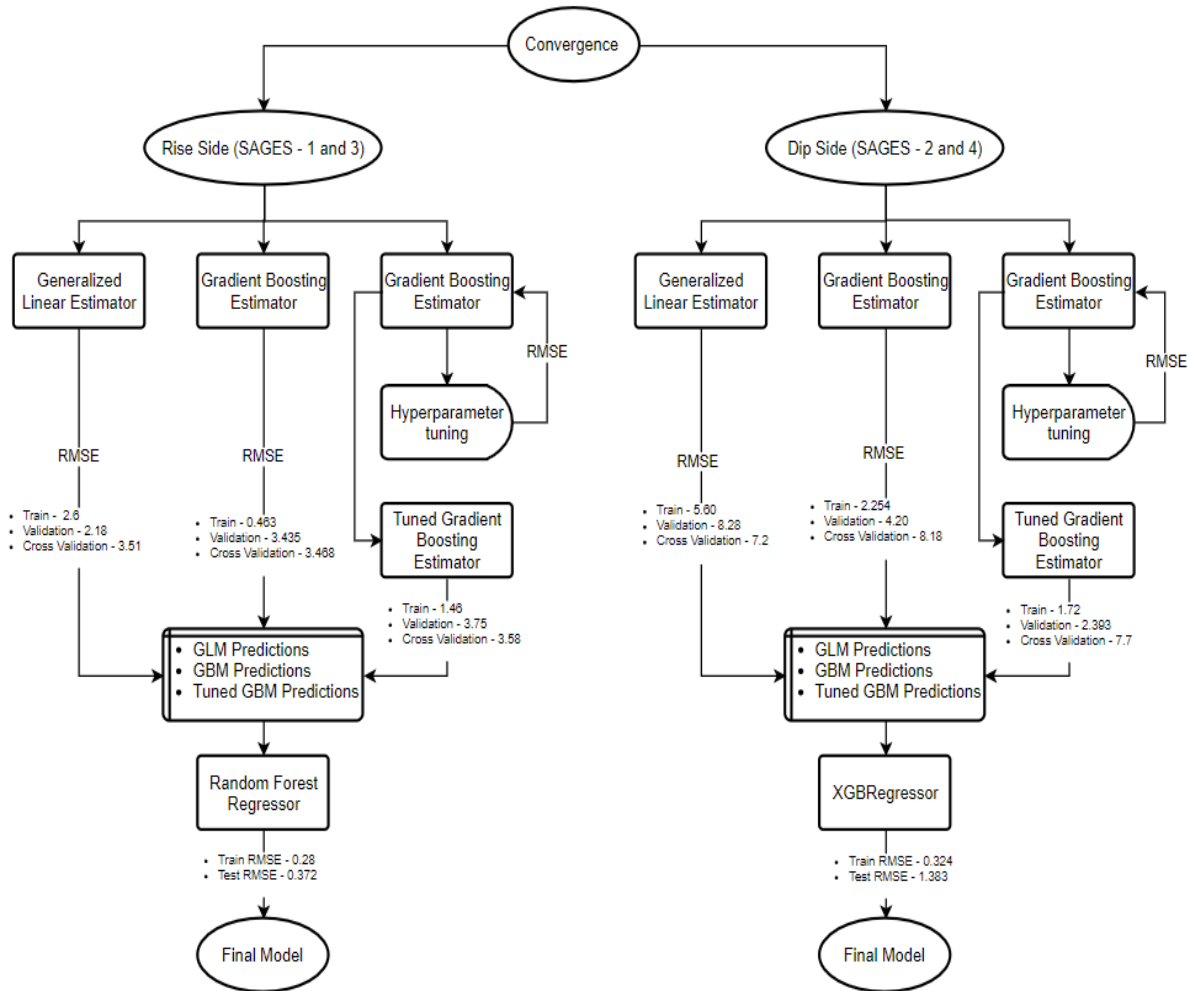


Figure 23: Module Architecture for predictive analysis of SAGES

5.3 Duration of Yielding in SAGES

We use the peaks found in feature extraction as a parameter to Density-Based Spatial Clustering of Applications with Noise in 1-Dimension (DBSCAN1D) [Meglicki et. al., 2021] for unsupervised machine learning to calculate the duration of yielding in SAGES. Density-Based Clustering refers to unsupervised learning methods that identify distinctive groups/clusters in the data, based on the idea that a cluster in data space is a contiguous region of high point density, separated from other such clusters by contiguous regions of low point density. DBSCAN1D is a base algorithm for density-based clustering in a single dimension (see Section-3.3). It can discover clusters of different shapes and sizes from a large amount of data, which may contain noise and outliers.

DBSCAN1D algorithm requires 2 parameters – *eps* and *min_samples*.

- *eps* (ϵ) is a measure of distance used to locate the neighborhood of points. The value of *eps* (ϵ) is chosen through the nearest neighbors algorithm plot as shown below. By choosing the value at the elbow, we get the efficient and effective *eps* (ϵ) value. *eps* (ϵ) value may differ depending on the data.
- *min_samples* is the minimum threshold of the number of points clustered together to be considered dense. Its value should be greater than (dimension of data + 1) or (2 * dimension of data)

Pipeline Functions – Data Formation

Require: *.txt file extracted from XR5 data loggers installed in SAGES

Function convert (*file*)

```

if provided file is a text file
  txt ← file opened in read mode
  lines ← [L94 to LN] for L in txt
  df ← map to DataFrame ← split_clean(lines)
  df ← headers ← ['Date', 'Timestamp', 'Leg-1', 'Leg-2', 'Conv']
return df

```

```

Function split_clean(string)
string ← split(',') ← replace({' ': '', '\n': ''})
return string

```

Pipeline Function – Data Preprocessing

Require: timeframe == '1min'
 DataFrame

```

Function ConvResamp (df, timeFrame)
df.loc[:, 'Datetime'] ← to_datetime(str(df['Date']) + ' ' + str(df['Timestamp']))
df ← index ← 'Datetime'
conv ← dict ← {'Timestamp', 'Leg-1', 'Leg-2', 'Conv'} ← value ← 'last' for each key in dict

df ← resample(timeFrame) ← aggregate(conv)
NaN values are dropped
df ← headers ← ['Timestamp', 'Leg-1', 'Leg-2', 'Conv']
df[['Leg-1', 'Leg-2', 'Conv']] ← df[['Leg-1', 'Leg-2', 'Conv']] ← float
return df

```

Pipeline Function – Plotting Peaks

Require: DataFrame

```

Function plotting_peaks(df)
p1, pts1 ← finding_peaks(df, 'Leg-1')
p2, pts2 ← finding_peaks(df, 'Leg-2')
p3, pts3 ← smoothing_conv(df)

Plot ConvResamp(df, '1min')['Leg-1']
Plot (p1, Array(ConvResamp(df, '1min')['Leg-1'])[p1])

Plot ConvResamp(df, '1min')['Leg-2']
Plot (p2, Array(ConvResamp(df, '1min')['Leg-2'])[p2])

Plot ConvResamp(df, '1min')['Conv'] ← savgol_filter

```

DBSCAN1D is suitable for spatial clustering of different densities and sizes in 1-dimension in large datasets. We develop the algorithm to create spatial clusters of prominent peak points found in the Load and Convergence variation of SAGES recorded during deployment. SAGES when undergoing yielding produce a dense cluster of peak point patterns at various stages of monitoring due to the application of pressure at valves. Using DBSCAN1D, we can create clusters of prominent peak points generated during yielding and assign them adjacent labels. The resultant dataset consists of the prominent peak values, cluster labels, DateTime of the respective peak, and uniform step-minute value of the respective peak point. In the resultant dataframe, we convert the DateTime index to durations for the respective clusters. From Figure 25 which displays variation and spatial clustering plot, we can interpret the total number of clusters without noise as well as the respective peak points marked where yielding takes place in SAGES. Below is the duration table of the clusters.

Pipeline Function – Finding Peaks and Edges

Require: DataFrame
 Feature column names in ['Leg-1', 'Leg-2', 'Conv']
 distance == 1

```
width == {1; feature in ['Leg-1', 'Leg-2']
          7; feature == 'Conv'}
prominence = [0.5, 25]
```

```
Function finding_peaks(df, feature)
peaks, pts ← ConvResamp(df, '1min')[feature], distance ← distance, width ← width,
            height ← mean(feature), prominence ← prominence ← find_peaks
return peaks, pts
```

```
Function smoothing_conv(df)
peaks, pts ← ConvResamp(df, '1min')['Conv'], distance ← distance, width ← width,
            height ← mean('Conv') ← find_peaks
return peaks, pts
```

	Values	Cluster	Peaks
Datetime			
2017-12-09 11:47:00	224.9	9	24
2017-12-09 11:52:00	224.9	9	29
2017-12-09 11:56:00	224.9	9	33
2017-12-09 12:00:00	224.9	9	37
2017-12-09 12:04:00	224.9	9	41
...
2018-01-04 05:43:00	271.8	8	37100
2018-01-04 06:27:00	271.8	8	37144
2018-01-04 07:06:00	271.8	8	37183
2018-01-04 08:27:00	271.8	8	37264
2018-01-04 08:33:00	271.8	8	37270

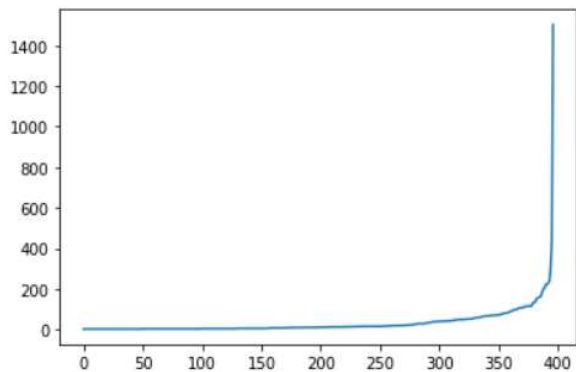


Figure 24: (a) Resultant dataframe by clustering **(b)** Nearest neighbor algorithm plot

Assumptions –

- A minimum of three prominent peaks are present during yielding in the variation plot of XR5 data files.
- Convergence either increases or decreases during yielding. (Observed)
- The difference in environmental conditions can lead to noise.

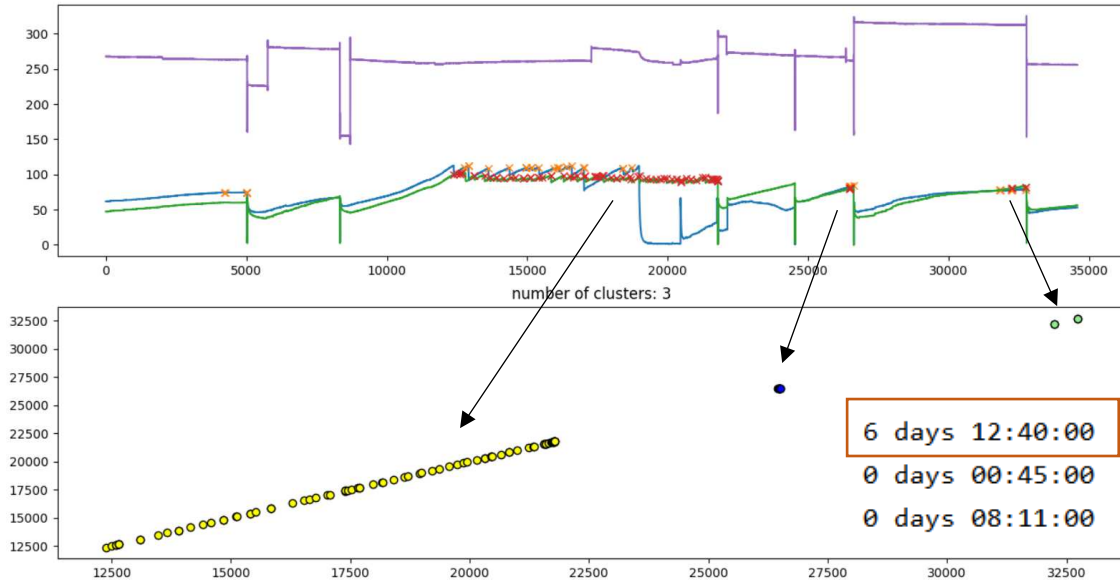


Figure 25: Prominent peak and Cluster representation with a 1-minute step, output

Pipeline Function – Calculating Duration of Yielding

Require: DataFrame

```
eps = [500, 1000]
min_samples = [3, 7]
```

Function ai_model_cluster(df, eps, min_samples)

```
p0, pts0 ← finding_peaks(df, 'Leg-2')
p1, pts1 ← finding_peaks(df, 'Leg-1')
db ← DBSCAN1D(eps ← eps, min_samples ← min_samples)
label ← db.fit_predict(p0)
create mask ← bool(db.labels_) ← map with {0, 1}
mask[db.core_samples_indices_] ← True
num_clusters ← Length(set(labels)) - (1 if -1 in labels else 0)
unique_labels ← set(labels)
# Give a list of colors for different clusters ← colors ← List
```

```
for k, col in zip(unique_labels, colors)
```

```
    if k == -1
        col ← 'k'
    class_member_mask ← (labels == k)
    xy ← p0.reshape(-1, 1)[class_member_mask & mask]
    Plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor ← col, markeredgecolor ← 'k', markersize ← 6)
```

```
df_ ← DataFrame ← {'Values': ConvResamp(df, '1min')['Leg-2'].iloc[peaks], 'Clusters': db.labels_,
                  'Datetime': ConvResamp(df, '1min')['Leg-2'].index, 'Peaks': peaks}
```

```
df_ ← index ← 'Datetime'
```

```
df_ ← rows having 'Cluster' > -1
```

```
for i in set(df_.Cluster)
```

```
    print ← df_[df_['Cluster'] == i].iloc[-1].name - df_[df_['Cluster'] == i].iloc[0].name
```

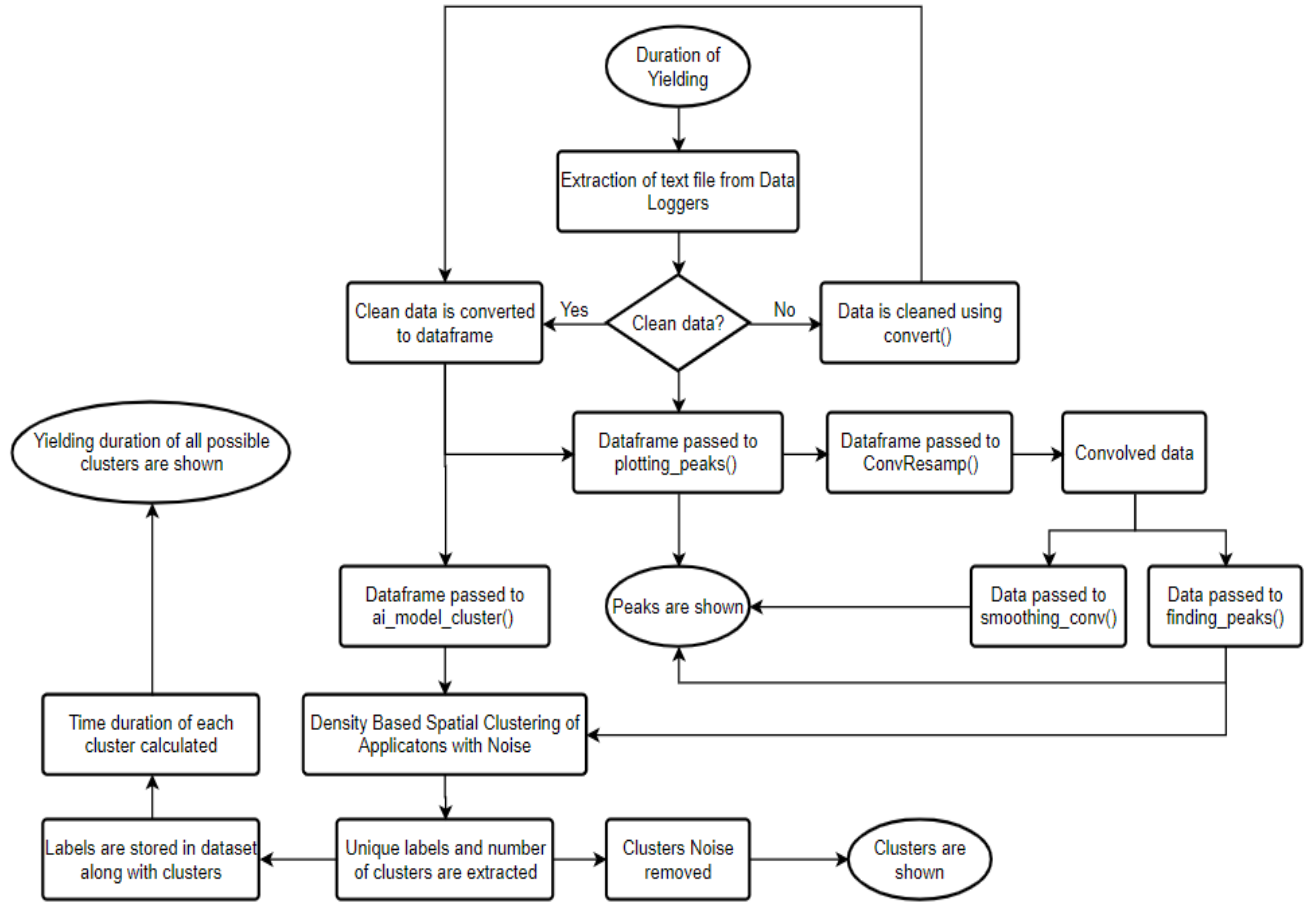


Figure 26: Module Architecture for predictive analysis of Duration of Yielding in SAGES

6. RESULTS

6.1 Model Comparison

Model development for the predictive analysis of strata load at withdrawal in SAGES uses only Generalized Linear Estimators (GLE) fit on Conditional GANs resampled data, each for the segmented locations. Similar variance in performance metric score has been observed for the predictive analysis of strata load at withdrawal at both rise and dip side. The key metrics for the purpose of machine learning model selection are Mean Absolute Error (MAE), R2-score as these metrics provide better insights for regression-based tasks.

Machine Learning Model (SAGES)	Load at Withdrawal (Rise Side)	
	MAE	R2-Score
Gradient Boosting Estimator (GBE)	33.007	-0.926
Generalized Linear Estimator (GLE)	23.103	0.472
Generalized Linear Estimator + CTGAN	7.729	0.872

Table-10: Performance metric for predictive analysis of Strata Load at Withdrawal in SAGES at Rise Side

MAE	R2	AIC
8.485	0.905	6744.197

Table-11: GLE + CTGAN performance model score for Strata Load at Withdrawal (Dip Side)

By comparing the different models, including composite and ensembles, we can observe that ensemble models, that are Random Forest Regressor and XGBoost Regressor perform significantly better compared to individual learners. Therefore, for the final model, we will be using the models for end-stage predictive analysis of Convergence monitored in SAGES.

Machine Learning Model (SAGES)	Convergence (Rise Side)			Convergence (Dip Side)		
	MAE	RMSE	MSE ¹⁵	MAE	RMSE	MSE
Generalized Linear Estimator	2.376	3.086	11.097	4.080	6.213	50.604
Gradient Boosting Estimator	2.785	3.503	12.574	5.830	7.757	65.818
Tuned GBM	2.67	3.510	12.712	4.220	6.153	56.59
Ensemble		0.372 (XGBoost)			1.38 (Random Forest)	

Table-12: Performance metric for predictive analysis of Convergence in SAGES

Feature importance plots signify the percentage probability, or the relative importance of feature variables used by the machine learning model for the prediction of the target variable. It is calculated as the decrease in the impurity of the node in a boosting tree weighted by the probability of reaching the respective node. From Figure 27 and Figure 28, we can observe that slice category number, strata load at withdrawal at both rise and dip side and the setting load at dip side play an important role for the predictive analysis of convergence, in *mm*. Features such as SAGES number, Level, Block lie on the lower side of the variable importance plot, which signifies that the features are highly uncorrelated.

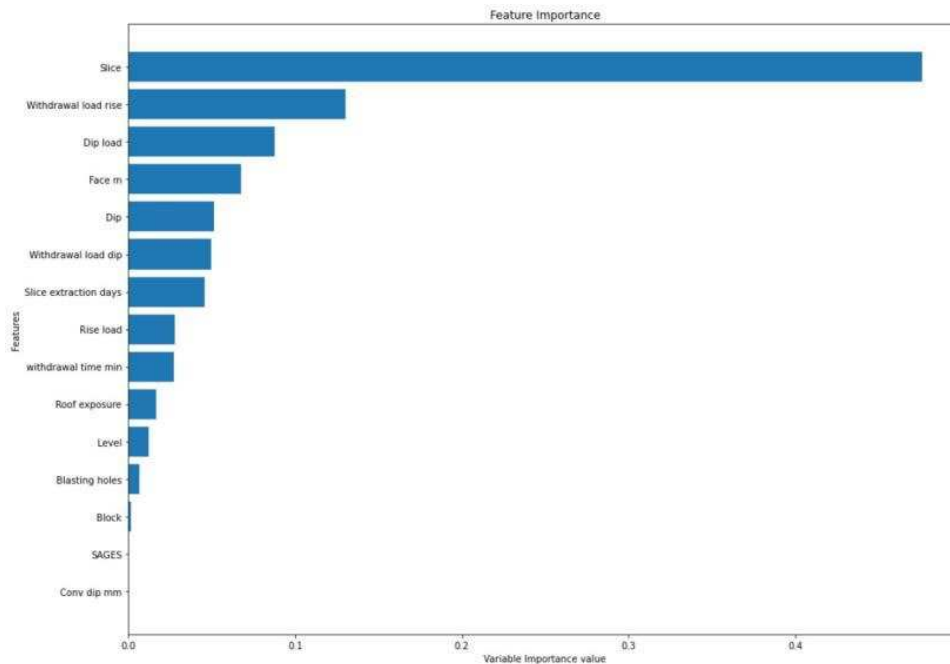


Figure 27: Feature importance plots for predictive analysis of Convergence, in *mm* using tuned GBM (Rise Side)

¹⁵ MSE – Mean Squared Error

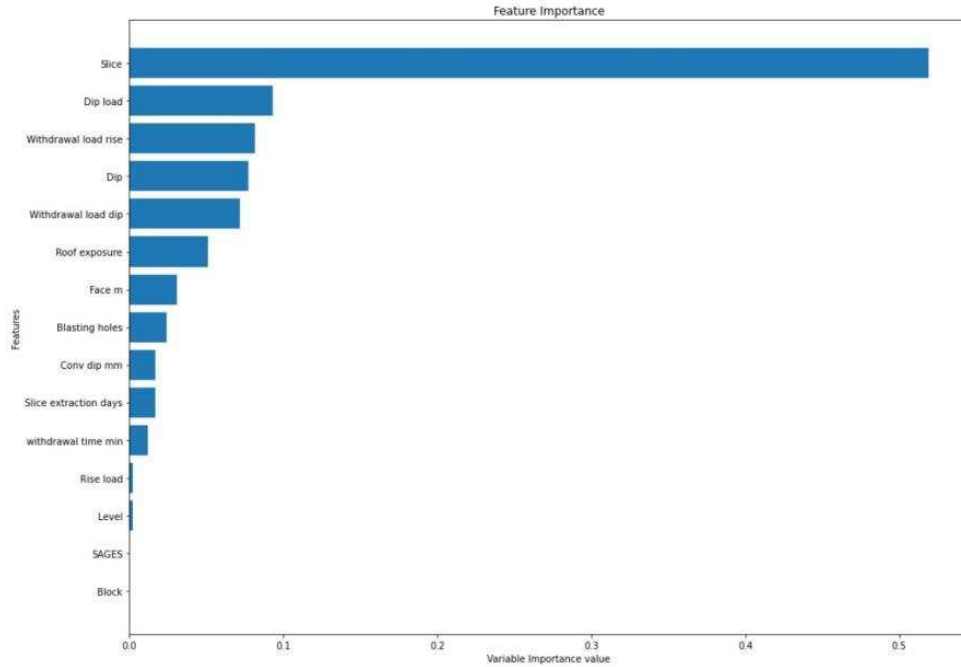


Figure 28: Feature importance plots for predictive analysis of Convergence, in *mm* using GBM (Rise Side)

SHAP dependency plots (see Figure 29) show how much a particular variable has affected the target variable. SHAP values allow us to decompose any prediction into the sum of effects of each feature value. Strata load at withdrawal (dip side) in SAGES have a significant positive effect on Convergence, in *mm* followed by slice category number indicating a sparse but significant dependency from Convergence value in SAGES.

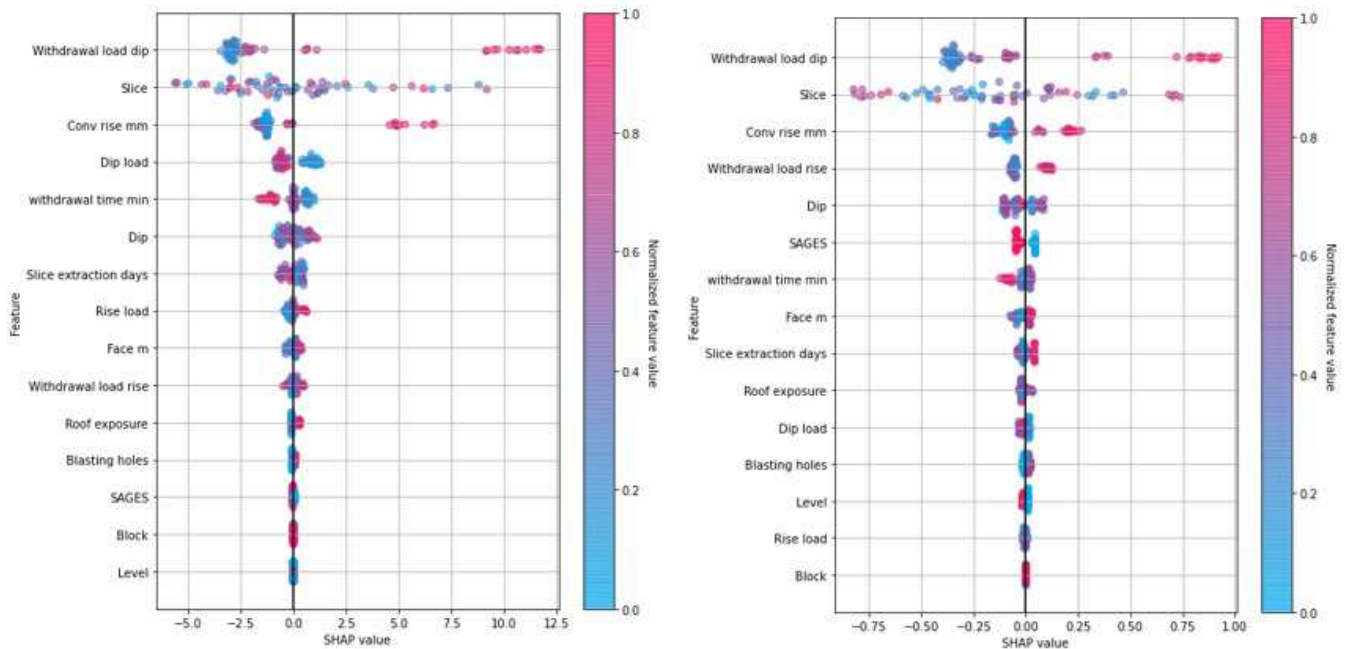


Figure 29: SHAP dependency plots for predictive analysis of Convergence, in *mm* (Dip Side)

7. CONCLUSION

This paper presents our approach for predicting Convergence, in *mm*, Strata Load at Withdrawal, in *ton*, and duration of yielding in SAGES deployed in underground coal mines. The predictive models were designed and developed using the following methods, Generalized Linear Estimator, Gradient Boosting Estimator, Density-Based Spatial Clustering of Applications with Noise in 1-Dimension (DBSCAN1D) for automating the analysis of the after-effects of depillaring operations. The models do not need to be trained for separate SAGES machines (four SAGES machines with number 1 to 4 deployed in the underground mines). The SAGES number has been feature engineered and merged to the dataset. Additionally, it was also observed that SAGES numbers were not predominantly correlated with other features, therefore, it was concluded to not develop separate models for each SAGES machine. The main features which were of importance are setting load, dip value, face value, slice number, etc. which when given as an input to the model retain the characteristics and properties. By incorporating Artificial Intelligence along with IoT in SAGES in the underground mines, a safe working environment could be ensuring for timely decision making. SAGES can be preset with the specifications recorded through predictive analysis which can help further in deducing other mechanical properties like pressure, temperature, setting load, and so on.

ACKNOWLEDGEMENT

This work was carried out under the guidance of Prof. Dheeraj Kumar and Prof. Upendra K. Singh, Department of Mining Engineering, Indian Institute of Technology (ISM), Dhanbad, and very special gratitude to them for reviewing the paper and giving essential tips to improve the paper's quality.

REFERENCES

- [1] Dr. Upendra K. Singh, Dheeraj Kumar, Shri N.V.N. Reddy. 2016. Development of Self-Advancing Mobile Goaf Edge Support (SAGES) for Depillaring Operation in Indian Coal Mines, *6th Asian Mining Congress, The Mining, Geological and Metallurgical Institute of India (MGMI), 2016*.
- [2] Ankit Singh, Dheeraj Kumar, Jürgen Hötzel, IoT Based information and communication system for enhancing underground mines safety and productivity: Genesis, taxonomy and open issues, *Ad Hoc Networks, Volume 78, 2018, Pages 115-129, ISSN 1570-8705, <https://doi.org/10.1016/j.adhoc.2018.06.008>*.
- [3] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, Kalyan Veeramachaneni. 2019. Modelling Tabular Data using Conditional GAN. *arXiv:1907.00503v2 [cs.LG] 28 Oct 2019*.
- [4] Martin Ester, Hans-Peter Kriegel, Jorg Sander, Xiaowei Xu. 1996. A Density Based Algorithm for Discovering Clusters. *KDD-96 Proceedings*.
- [5] Baturynska, Ivanna & Martinsen, Kristian. (2021). Prediction of geometry deviations in additive manufactured parts: comparison of linear regression with machine learning algorithms. *Journal of Intelligent Manufacturing*. 32. [10.1007/s10845-020-01567-0](https://doi.org/10.1007/s10845-020-01567-0).
- [6] Tianqi Chen, Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. *arXiv:1603.0.2754v3 [cs.LG] 10 Jun 2016*.
- [7] Ali, Jehad & Khan, Rehanullah & Ahmad, Nasir & Maqsood, Imran. (2012). Random Forests and Decision Trees. *International Journal of Computer Science Issues (IJCSI)*. 9.
- [8] Christine De Mol, Ernesto De Vito, Lorenzo Rosasco, Elastic-net regularization in learning theory, *Journal of Complexity, Volume 25, Issue 2, 2009, Pages 201-230, ISSN 0885-064X, <https://doi.org/10.1016/j.jco.2009.01.002>*. (<https://www.sciencedirect.com/science/article/pii/S0885064X0900003X>)
- [9] Zou, Hui & Hastie, Trevor. (2005). Regularization and variable selection via the elastic net (*vol B 67, pg 301, 2005*). *Journal of the Royal Statistical Society Series B*. 67. 768-768. [10.1111/j.1467-9868.2005.00527.x](https://doi.org/10.1111/j.1467-9868.2005.00527.x).
- [10] Meglicki, Bartosz. (2021). Linear time DBSCAN for sorted 1D data and laser range scan segmentation.

- [11] Atzori, Luigi & Iera, Antonio & Morabito, Giacomo. (2010). The Internet of Things: A Survey. *Computer Networks*. 2787-2805. [10.1016/j.comnet.2010.05.010](https://doi.org/10.1016/j.comnet.2010.05.010).
- [12] Government of India, Ministry of Coal, Research & Development. *Annual Report 2014-2015, 2015*.
- [13] Cunningham SJ. Machine learning and statistics: A matter of perspective. Working paper 95/11. Hamilton, New Zealand: University of Waikato, Department of Computer Science, 1995.
- [14] Müller, Marlene. (2004). Generalized Linear Models. DOI: [10.1007/978-3-642-21551-3_24](https://doi.org/10.1007/978-3-642-21551-3_24).
- [15] Wignall T.K., De Geoffroy J. (1987) Application of the General Linear Model (GLM) and Operations Research (OR) Models to the Optimization of Field Exploration and Development Planning. In: *Statistical Models for Optimizing Mineral Exploration*. Springer, Boston, MA. https://doi.org/10.1007/978-1-4613-1861-3_5
- [16] Kumari, Khushbu & Yadav, Suniti. (2018). Linear regression analysis study. *Journal of the Practice of Cardiovascular Sciences*. 4. 33. [10.4103/jpcs.jpcs_8_18](https://doi.org/10.4103/jpcs.jpcs_8_18).
- [17] S. Na, L. Xumin and G. Yong, "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm," 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, 2010, pp. 63-67, doi: [10.1109/IITSI.2010.74](https://doi.org/10.1109/IITSI.2010.74).
- [18] Babur, Iftikhar & Ahmad, Jawad & Ahmad, Bilal & Habib, Muhammad. (2015). ANALYSIS OF DBSCAN CLUSTERING TECHNIQUE ON DIFFERENT DATASETS USING WEKA TOOL. *Science International*. 27. 5087-5090.
- [19] A. Singh, U. K. Singh and D. Kumar, "IoT in mining for sensing, monitoring and prediction of underground mines roof support," 2018 4th International Conference on Recent Advances in Information Technology (RAIT), 2018, pp. 1-5, doi: [10.1109/RAIT.2018.8389041](https://doi.org/10.1109/RAIT.2018.8389041).
- [20] JunYing Chen, Zheng Qin and Ji Jia, 2008. A Weighted Mean Subtractive Clustering Algorithm. *Information Technology Journal*, 7: 356-360 DOI: [10.3923/itj.2008.356.360](https://doi.org/10.3923/itj.2008.356.360)
- [21] Satchidanandan Dehuri, Chinmay Mohapatra, Ashish Ghosh and Rajib Mall, 2006. A Comparative Study of Clustering Algorithms. *Information Technology Journal*, 5: 551-559 DOI: [10.3923/itj.2006.551.559](https://doi.org/10.3923/itj.2006.551.559)
- [22] Hyder, Zeshan & Siau, Keng & Nah, Fiona. (2019). Artificial Intelligence, Machine Learning, and Autonomous Technologies in Mining Industry. *Journal of Database Management*. 30. [10.4018/JDM.2019040104](https://doi.org/10.4018/JDM.2019040104).
- [23] Matloob, Sundas & Li, Yang & Khan, Khurram. (2021). Safety Measurements and Risk Assessment of Coal Mining Industry Using Artificial Intelligence and Machine Learning. *Open Journal of Business and Management*. 09. 1198-1209. [10.4236/ojbm.2021.93064](https://doi.org/10.4236/ojbm.2021.93064).
- [24] Nalbantov, Georgi & Smirnov, Evgueni & Nalbantov, Dilyan & Weiss, Gerhard & Nienhaus, Karl & Warcholik, Manuel & Mavroudis, Fiona. (2010). Image Mining for Intelligent Autonomous Coal Mining.. 17-23.
- [25] Janusz, Andrzej & Grzegorowski, Marek & Michalak, Marcin & Wróbel, Łukasz & Sikora, Marek & Ślęzak, Dominik. (2017). Predicting seismic events in coal mines based on underground sensor measurements. *Engineering Applications of Artificial Intelligence*. 64. 83-94. [10.1016/j.engappai.2017.06.002](https://doi.org/10.1016/j.engappai.2017.06.002).
- [25] Y. Fu, C. Aldrich, Deep Learning in Mining and Mineral Processing Operations: A Review, *IFAC-PapersOnLine*, Volume 53, Issue 2, 2020, Pages 11920-11925, ISSN 2405-8963, <https://doi.org/10.1016/j.ifacol.2020.12.712>.