Real-time Incident Detection in Geothermal Drilling Operations using Machine Learning Solutions

Aira Aspiras[1], Sadiq Zarrouk[1], Ralph Winmill[2] and Andreas Kempa-Liehr[1]

[1] University of Auckland Geothermal Institute, Grafton, Auckland, New Zealand (Aotearoa)

[2] Contact Energy Ltd, Wairakei, Taupo, New Zealand (Aotearoa)

aira.aspiras@auckland.ac.nz

Keywords: *Geothermal drilling, Incident Detection, Risk Reduction, Machine Learning*

ABSTRACT

Geothermal energy, while a reliable low-carbon resource, represents only a small fraction of global renewable capacity due to high upfront costs and resource risks. Drilling wells accounts for a substantial portion of geothermal project costs; successful drilling is thus crucial for geothermal projects. Finishing wells on-time and on-budget has always been a challenge for operators and mostly has modest success rates.

Overall drilling efficiency can be improved by increasing the rate of penetration (ROP) and reducing non-productive times (NPT). Geothermal systems present unique challenges like fault structures, severe lost circulation, and high temperatures that lead to potentially expensive incidents. Leveraging machine learning technologies offers promise in addressing these drilling challenges and optimising operations.

The drilling incident detection system using systematic and automated feature engineering on time-series data was able to successfully predict incidence of hole-related NPT. This system can potentially be deployed and assist drilling personnel in making tactical decisions during drilling, in the future, thereby reducing operational risks and enhancing overall drilling efficiency and cost performance.

1. GEOTHERMAL DRILLING

Improving the overall drilling efficiency in geothermal projects involves two main strategies: enhancing the rate of penetration (ROP) and reducing non-productive times (NPT) [1]. ROP measures the rate at which the hole is drilled [2], while NPT refers to the time lost due to various problems encountered during drilling [3]. Geothermal drilling faces unique challenges compared to drilling in oil fields, as geothermal systems often feature fault structures, severe lost circulations, and high temperatures [4][5]. These characteristics increase drilling risks and contribute to potential NPT and incidents. Stuck pipe and loss of well control are among the most severe incidents, resulting in significant losses of time, capital, and even lives [5].

Stuck pipes incur significant rig time in freeing the pipe, and if unsuccessful, may end up with the whole BHA, including expensive tools, being left in the hole and subsequently losing the hole itself. Several stuck pipe mechanisms exist, such as solid-induced pack-off, differential sticking, wellbore geometry, and others [6,7].

This study's general objective is to improve the overall drilling performance by de-risking and optimising drilling operations through the development of an incident detection system that will serve as an early warning system that would assist the Drillers and Engineers in tactical and strategic decisions.

1.1 Drilling Data Acquisition

Drilling formation several kilometres deep into the earth requires relating and understanding several different parameters from the different surface and downhole instrumentations to determine the state of the hole and its general location at any given point. Rigs have surface instrumentations to determine parameters like weight on bit (WOB), hook load (HL), string rotation (RPM), rotary torque (TRQ), and standpipe pressures (SPP) while drilling. In addition to surface instrumentation, additional downhole data through measurement-while-drilling (MWD) sensors are now installed in the BHA for directional control and drilling data acquisition [7]. Over the years, more sensors have been added by and around the rig to acquire more data[8] with Mud Logging units becoming a mainstay drilling component to monitor both drilling- and geologic-parameters in real-time.

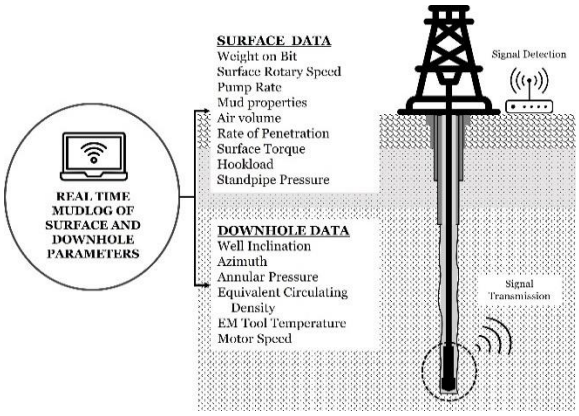


Figure 1 Real-time Mudlog captures both the surface measured data and downhole transmitted data from the MWD and PWD sensors.

Mud logs make it easier for drilling personnel to monitor trends indicating drilling efficiency and overall health of the drilled hole and the controllable surface parameters used at each point in time [7]. Drillers use trends from these parameters to decide whether to drill ahead, pause and extend circulation, conduct a wiper trip, or pull out to the surface [9]. Modern Mud logs are also recorded in digital format, so they are easily stored, retrieved, displayed at multiple locations, and connected to other drilling software, if necessary [7]. Drilling Engineers use these parameters in offset well analysis during the planning stage to prevent

drilling issues and provide guidance on how to address them if encountered as part of lessons learned [10].

During drilling, the data available to the drillers is a time-based drilling parameters transmitted real-time and not depth-based, as in the previous section. Drillers analyse these different data correlations and evaluate the best possible course of action almost instantaneously. Drilling data shown in real-time monitoring systems can be categorised into the controllable, input parameters and the drilling condition indicators or resultant drilling parameters (Figure 2).

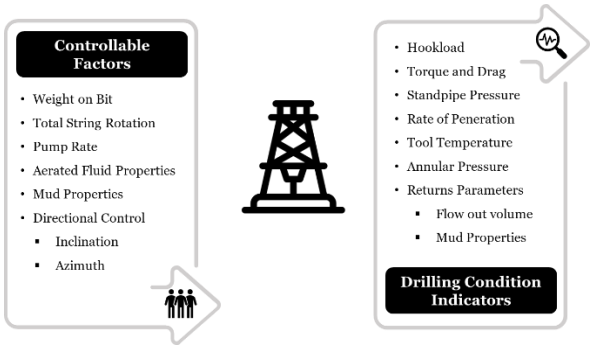


Figure 2 Drilling personnel are able to control the WOB, String Rotation, Pump rate, Mud and air parameters, and well path direction at the surface. Some parameters like torque ,drag and ROP, among others are used to evaluate drilling conditions

1.2 Data Used in the Study

Downhole and surface drilling data used in this study is captured at 12 datapoints per minute or 720 datapoints per hour. This data is automatically seen at surface in graphical and text forms through a visualisation software that is readily accessible not only to the rig, but also to office-based drilling personnel through the internet (Figure 1).

The drilling data comes from a set of deep geothermal wells in New Zealand and is focused only on drilling the reservoir section where severe losses and, thus, hole problems are often experienced − particularly, the 12 ¼″ hole section until well completion. For the purposes of this study and illustrations, the identified NPT is limited only to hole-related drilling issues like tight spots and stuck pipes, among others. The identification of NPT and PT datapoints are done by domain experts and accepted as "truth".

2. LITERATURE REVIEW

A machine learning program that detects incidents or troubles ahead of time is very valuable in any drilling operation, including geothermal. Abnormal drilling events that can potentially lead to significant lost rig time need to be minimised or eliminated. Having a model that can detect or predict these allows for applying mitigation measures in real time while drilling.

A few papers focus on generic incident detection, detecting multiple events, with some classifying them further, like studies conducted by Gurina et al [11–13]. Gurina et al.'s real-time analogues search model compares real-time MWD data and historical drilling database with 94 events from 80 wells in 19 oilfields to detect anomalies and identify their type using the Gradient Boosting classification model. The study found that MWD signals that undergo clustering and aggregated statistics as input to the classifier model showed the best results [12].

Similarly, Antipova et al. developed a machine learning model based on Gradient Boosting Decision Trees [14]. It allows the detection of anomalies using real-time drilling data within a 4-hour sliding window by automatically extracting patterns and feeding them into the model. The model then calculates the probability per accident and raises a warning if it crosses an established threshold. The database consisted of 80+ incidents from 20+ wells. Types of accidents identified are stuck, fluid show, seal, washout, break of drillstring and Mud loss. Data preprocessing was done to the raw file: (1) "cleansing" of data, which is the elimination of factors that reduce the quality of data (i.e. duplicates, abnormal values, gaps, noise), (2) simple "feature extraction" - statistical mean, standard deviation, and the difference between mean value during 1st and 2nd part of the considered interval around the accident, then (3) "optimisation", which includes dimension reduction and elimination of insignificant features.

A succeeding study by Gurina et al. detects pre-accident patterns with probabilities within 6 hrs before the accident by applying the Bag-of-features (BoF) method a wavelet representation of multi-variate time-series data to forecast [11]. The data was a database of 125 past drilling accidents from 100 wells. Dataset was divided into small segments and transformed into wavelets, which were then clustered using K-means algorithm. Parameters for this type of feature generation needed to be selected properly: wavelet type, level of decomposition, and the number of clusters. To compare the output for this model, other models were developed: Convolution Neural Network (CNN) model was developed using Stochastic gradient descent (SGD) optimiser and cross-entropy as loss functions and the Breakdown model developed by Antipova et al. [14]. BOF outperformed both models. It was able to forecast 70% of drilling accidents with a false positive rate of 40%.

A number of incident detection studies focus on stuck pipes, which can easily lead to potentially costly incidents. Variations in stuck pipe-related studies include predicting stuck pipe occurrences, estimating stuck pipe probabilities, and identifying the type of stuck pipe mechanism.

Shadizadeh et al. [15] produced one of the earliest works in drilling ML by using ANN to predict stuck pipe occurrences in Iran's oil fields to estimate stuck pipe risk during the well planning procedure. Two hundred seventy-five (275) cases with 115 stuck pipes were used for this study, where the events were divided into dynamic and static conditions. Various drilling data and Mud properties were initially considered, but after domain expert feature generation and selection, the final input features were reduced to six (6), and these were fed into a neural network. Results showed accuracy for all cases to be >75%.

Two studies by Al-Baiyat et al. [16] and Jahanbakshi et al. [17] on stuck pipe prediction used SVM and ANN as the main models with variants in the transfer functions or kernel used. SVM outperformed ANN in both studies. Another commonality between these studies is the lack of feature generation and selection prior to fitting the models and the lack of cross-validation in the test validation. Al-Baiyat et al. used drilling parameters to predict stuck pipe occurrences

and further classified as per cause of stuck pipe - differentially stuck or mechanically stuck [16]. Data comprised 48 datasets for training and 18 datasets for testing from a published work by Murillo et al. [18]. Jahanbakshi et al. [17] predicted drill pipe sticking events in horizontal and sidetracked wells in Iran's oil fields, using data from 63 wells drilled in the Persian Gulf with a total of 214 datasets, which was divided into 70% training and 30% validation sets.

Rostami and Manshad [19] developed the same models of ANN and SVM for stuck pipe prediction two years later but using more elaborate methodology that we are familiar with today incorporating feature selection and hyperparameter tuning. After removing outliers, feature selection was done by finding features with the highest correlation to the output. The models were then "tuned" to find the optimal hyperplanes for the SVM, and performance is now evaluated with 5-fold cross-validation. The study showed that both models performed satisfactorily on the unseen data and had no significant difference in predicting stuck pipe occurrences.

Abbas et al. [20] developed a stuck pipe incidence model caused by mechanical sticking using ANN and SVM, with the latter performing better. The dataset used consisted of 1010 datasets from drilled wells in Southern Iraq. The model used the fscaret R-package for feature selection, resulting in 25 features being used as inputs to the models - wellbore inclination, rate of penetration, lithology type, and BHA length being the most relevant inputs. No additional features were generated. The best classification performance result for the ANN model used all-forward backpropagation with the 'tansig' transfer function while for the SVM was the model which used gaussian kernel function (all results above 80%).

The study by Elmousalami and Elaskary [21] evaluated more than 10 machine learning models in predicting stuck pipe incidents using surface drilling data from 103 wells comprising 26 stuck pipe events in the Gulf of Suez. No feature generation or selection was used to preprocess the input drilling parameters. The best prediction model, Decision Trees Ensemble, was used as the Optimal classifier in the optimisation methodology for the Stuck pipe mitigation system.

Khan et al. [22] published a study comparing machine learning classifiers, ANN and SVM and their variants, for detecting stuck pipe incidents using 268 datasets. Various combinations of base model, network structure, activation function, and kernel functions were run and used to predict outcomes. It was found that the best performing model for each algorithm was the ANN-Logistic Activation function and SVM-Radial Basis Function (RBF) kernel (C=10, sigma=0.1), in which the former was chosen as the best model, although SVM performance was still deemed acceptable.

Tsuchihashi et al. [23] used a different approach called 3D Convolutional Neural Network (CNN) on depth-domain data to detect the early signs of stuck pipes. The novelty of this research is that it focuses on detecting the pre-cursors of stuck pipe incidents and transforming time-series raw data into depth domain data. Thirty (30) field well data were used, including 6 stuck pipe incidents in the validation set. The model was able to detect early signs in 3 cases on the said set. Feature engineering using Gradient-weighted Class Activation Map (Grad-CAM) was attempted in this study but did not yield any meaningful improvement in the prediction.

From the incident detection studies available, majority used direct drilling parameters or manual feature engineering techniques like domain expert generated drilling indices and simple statistical descriptions of the dataset. Very few used automatic feature engineering techniques, and none used the systematic and extensive feature extraction and generation approach called tsFresh. While the studies applied training and testing datasets ratio as in typical ML methodologies, with some incorporating cross validation, none explored a holistic validation approach that can be understood and applied in actual practical real-time drilling scenario. As shown in the studies, there is a great potential in incident prediction, however, results for the evaluation of the different algorithms are inconsistent and more robust hyperparameter tuning procedure need to be developed to ensure quality of models. The numerical models that will be developed will address some of these gaps while exploring how geothermal data responds to ML models.

3. NUMERICAL MODEL DEVELOPMENT

Numerical models are useful in seeking solutions to complicated sets of mathematical models that are difficult to solve directly through analytical means, as is the case for machine learning. Overall, this study aims to develop simple and practicably useable machine learning model to enable the researcher and, eventually, the Drilling Engineers and Supervisors on site to be able to predict potential NPT and incidents in real time drilling data.

In these models, features are extracted from the generated windows and are used to predict the next event, evaluated using time-series split technique. Learning from the different ML studies reviewed, the methodology described in Figure 3 is deemed to be the most appropriate for the incident detection model and will be used for this study. The type of model used in this study is supervised machine learning method, which learns from labelled datasets to predict targets.

The following tools were used to develop the machine-learning models:

- Matplotlib (Version 3.8.3) is a Python-based, open-source library that provides simple and easy to use generic data visualization tools [24]. All visualizations in this project use Matplotlib and Seaborn library packages.
- Pandas (Version 2.2.1) is an open-source Python-based that provides a comprehensive set of tools for efficient data manipulation for use in widescale data analysis [25].
- Python (Version 3.12.2) is an open-source, object-oriented programming language which has built-in data structures and supports several modules and packages, encouraging program modularity and code reuse [26].

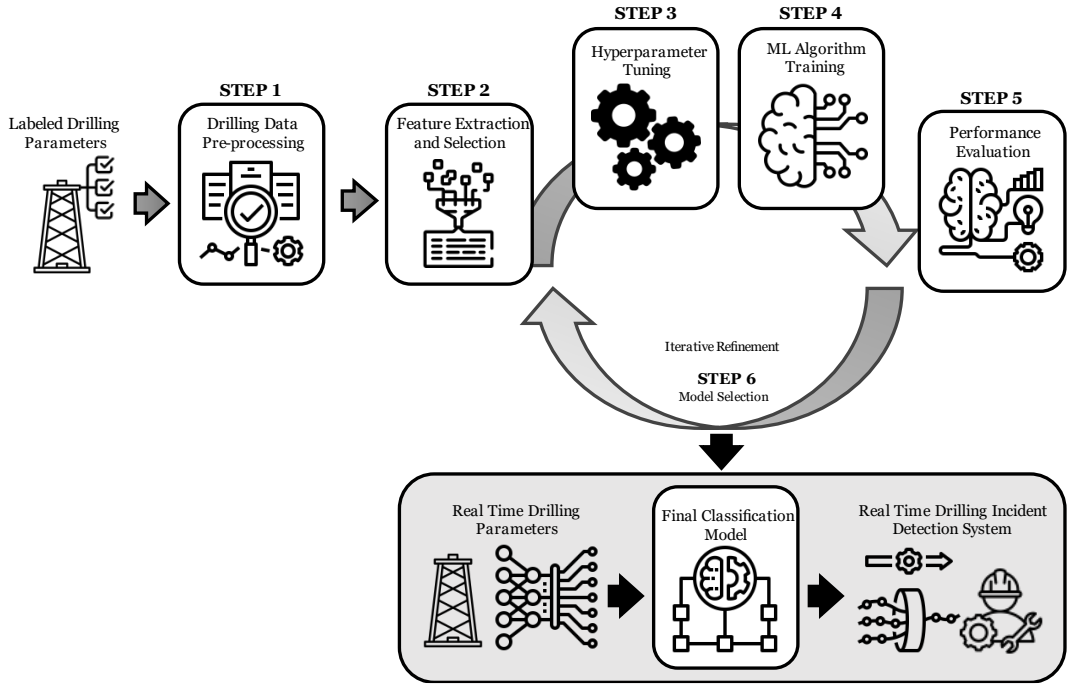


Figure 3 Machine Learning requires extensive data pre- processing techniques prior loading into the selected ML model, after which the model needs to be cross validated for fitness

- Sci-kit Learn (Version 1.4.1)is an open-source Python-based machine learning library that provides built-in machine learning algorithms and models, integrated with tools for model fitting, data preprocessing, model selection, and model evaluation, among others [27].
- Tsfresh (Version 0.20.2) or Time Series Feature Extraction on the basis of Scalable Hypothesis tests (TSFRESH) is a Python-based machine learning library package which supports systematised automatic feature extraction and selection [28].

3.1 Drilling Data-preprocessing

3.1.1 Data Quality Control and Outlier Removal

Raw drilling data comes with noise and outlier datapoints and so pre-processing is necessary. Typical drilling parameters used in the existing studies and the ranges of acceptable values set by Drilling Subject Matter Expert (SME) is shown in Table 1. Presented in Table 1 are the codes, units, minimum, and maximum data for each drilling parameter for standardization. Any values out of the range presented is removed from the dataset and replaced by the previous "good" value in every cell. Gaps in the dataset are also addressed similarly, with previous "good" data filling in the missing cells. Datasets that are not in the proper units are to be converted based on the reference below.

3.1.2 Data Preparation

Data may need to be prepared depending on the objective of the machine learning problem. In most forecasting or prediction studies involving time-series data [11,12,14], windows of determined time duration are created. As illustrated in Figure 4, the window sets can either be done sequentially or rolled such as the one followed in this study. Windows of size 2 hours were created and rolled every hour using TSFRESH. A total of 523 windows were created for this well with 22 time-series features.

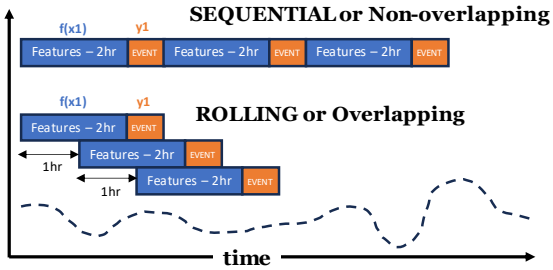


Figure 4 Data preparation alternatives for time-series data in predicting event outcomes.

3.2 Feature Extraction and Selection

Features are the inputs by which the ML models are trained; thus, the choice of features directly affects the performance of the models in terms of accuracy, speed of runs, and software requirement.

In contrast to naïve feature engineering and manual feature engineering, automatic feature engineering refers to the process of automatically generating complex features, especially time-series data from input dataset using computational algorithms without manual intervention, as this process can oftentimes be time consuming [27,28]. Some feature extraction libraries exist for this purpose - TSFresh [27], Catch22 [29], hctsa [30], AutoML [31], FeatureTools[32], FeatureWIZ [33], PyCaret [34].

Table 1 Drilling Parameter Range set by Drilling SME used in the study

Code	Drilling Parameters	Units	Min	Max
Bit_Depth	Present Bit Depth	m	0	--
Hole_Depth	Maximum Hole Drilled	m	0	--
Hook	Average Hookload	lbs	0	500 000
WOB	Average Weight on Bit	lbs	--	130 000
RPM_Surface	Average Surface Rotation	rpm	0	250
Flow_Out	Average Pump Flow out	lpm	0	10 000
SPP	Standpipe Pressure	psi	0	5 000
Flow_In	Pump Flow In	lpm	0	5 000
ROP	Instantaneous Rate of Penetration	mph	0	200
TRQ_ave	Average Surface Torque	ft-lbs	0	40
TRQ_max	Maximum Surface Torque	ft-lbs	0	40
Mud_In	Average Mud Weight In	ppg	8.3	15.0
Mud_Out	Average Mud Weight Out	ppg	8.3	15.0
Temp_In	Mud Temperature In	°C	0	200
Temp_Out	Mud Temperature Out	°C	0	200
Ann_Pres	Annular Pressure	psi	0	5 000
Int_Pres	Internal Pressure	psi	0	5 000
ECD	Equivalent Circulating Density	ppg	0	20
MWD_Temp	Measurement-While-Drilling Tool Temperature	°C	0	200
MLP_Pres	Air Injection Pressure	psi	0	5 000
Air_Scfm	Air Injection Volume	scfm	0	5 000
Block_Pos	Block Position	m	0	40

The Model is designed to be a machine learning model that serves as decision support system. It shall extract features from the 2hr windows and use those features to predict the succeeding events for potential hole-related issues. Figure 5 shows the procedure and actual results of one of the wells.

Each window was then passed on for automatic feature extraction using TSFRESH's built-in feature generation techniques (i.e. maximum, standard deviation, variance, and entropy), resulting to 17 226 features. False discovery rate (FDR) level was maintained at default value of 0.2. Relevant features were identified through their p-values using the transformer package, resulting in a final selection of 4130 features. These features will be the ones used to fit several machine learning models.

3.3 Hyperparameter Tuning

Several models were investigated based on a combined assessment from the literature review, as well as ease of use and computing resource requirements. Each of the ML algorithms come with hyperparameters that govern the training process and therefore need to be adjusted or calibrated to improve the performance of the models.

- Decision Trees (DT) is one of the most popular supervised machine learning techniques that can be used for classification and regression techniques due to their simplicity. Like a tree, it will start from a root node and will be repeatedly split into two subsets until a Boolean outcome at the leaf node, which represents the outcomes, is achieved [36].

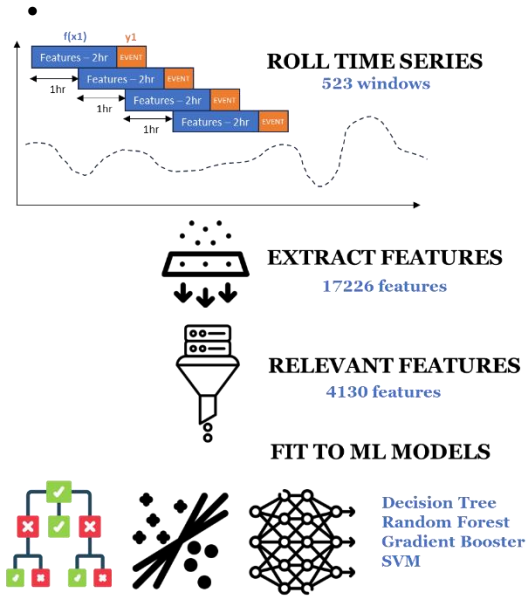


Figure 5 Automatic Feature extractions and Selections conducted using TSFRESH. Relevant features were used to predict events

- Random Forest (RF) is a learning algorithm based on Decision Trees and is thus also used extensively for classification and regression analysis. It extends the performance of DTs by forming multiple trees and making predictions through majority voting or averaging predictions from the individual trees. Random Forests address the overfitting that commonly plagues Decision Trees and are able to handle high-dimensional data and large datasets [37].
- Gradient Boosting (Gboost) is another popular Ensemble method. It is a learning algorithm that combines predictions of weak learners to create a strong learner and is useful for both classification and regression. The iterative process gradually produces a strong, robust predictive model. It is primarily used in supervised machine learning where the model is trained on labelled dataset to make predictions or infer relationships between features and targets [38].
- Adaptive Boosting (AdaBoost) was introduced by Freund and Schapire [39], which trains a series of weak learners iteratively. Each weak learner focuses on the predictions that were misclassified and assigns higher weights to these, forcing the subsequent weak learners to focus more on them. Finally, all weak learners are combined through a weighted majority voting scheme to produce the final strong classifier.
- Support Vector Machines (SVM) algorithm constructs an optimal hyperplane to perform classification and regression analysis by maximising the support vectors or those datapoints that are closest to the hyperplane [40]. The idea of SVM can be extended into non-linearly separable patters by transforming the input data into a higher-dimensional space using kernel functions, where it may be easier to find a separating hyperplane.

Hyperparameters tuning can be done by domain experts, literature reviews, or by iterative trial-and-error, choosing the combination with the best outcomes.

3.4 Machine Learning Training

Models are made to learn on the Training set only and the performance of the predicted results is assessed using the previously unseen Testing set. In these models, time-series split was used to handle time-dependent data to provide a more realistic split as it ensures that the model is not training on a future set of data, preventing a "leak" into past data.

3.5 Performance Evaluation

To fit the model, several metrics need to be considered to evaluate the appropriateness of the model used. When discussing performance metrics, especially in the context of machine learning and data analysis, it's crucial to understand its strengths, limitations, and appropriate usage. Several metrics are derived from Confusion Matrix [41,42]. Confusion matrices are useful; however, it is easier to present performance metrics as single numbers for easy reference and comparison.

For this study, the metrics used are:

- Accuracy,
- Area under Receiver Operating Characteristic Curve (ROC_AUC),
- F1 score, and
- MCC score

4. SUMMARY OF RESULTS AND DISCUSSION

Results of the different models assessed are shown in Figure 6. If only accuracy is the basis for the performance, it shows that all models very much predict NPT accurately. Accuracy is calculated as the ratio of the number of correct predictions to the total number of predictions made, [43]. It is simple and intuitive, but it also has limitations, particularly on imbalanced datasets like this project. NPTs form <15% of the total dataset; thus, predicting all events as PT would yield an 85% accuracy score, which is misleading. In addition, it treats all misclassifications equally, but in this case, a false negative is more critical than a false positive.

Another metric that is usually used is the ROC_AUC. The area under the ROC curve is a plot of the true positive rate (TPR) on the Y axis and the false positive rate (FPR) on the X axis. ROC_AUC measures how well a classifier performs in comparison to a false classification [44]. Compared to Accuracy, it performs better with unbalanced datasets but still treats misclassifications equally [45]. Based on the results, all models appear to be able to predict NPT properly which agrees with the Accuracy results.

Alternative metrics are needed to provide a more comprehensive evaluation of model performance, especially for highly imbalanced data set. F1 score is one of the widely used metrics for imbalanced datasets. It indicates the classifier's ability to correctly predict instances of the positive class [44]. A score higher than 0.5 is necessary to be able to achieve better than guess work. Now, the results shows that AdaBoosted Decision Tree, Adaboosted Random Forest, GBoost, and XGBoost presents more acceptable results for all 4 wells.

Another similar metric for highly unbalanced dataset is MCC. Brian Matthews introduced a new correlation coefficient called the Matthews Correlation Coefficient (MCC) which provides a balanced measure that considers true positives, true negatives, false positives, and false negatives. It is a robust metric for binary classification and imbalanced datasets, although it is difficult to interpret intuitively. MCC ranges from -1 to +1, where 0 represents random prediction, +1 indicates perfect prediction, and -1 indicates an inverse relationship between prediction and target [46]. As seen in the results, AdaBoosted Decision Tree, Adaboosted Random Forest, and GBoost present acceptable results for all 3 wells, with AdaBoosted Decision Tree and Gboost both being the best models.

The only caveat in this process is that the feature extraction step takes a very long time and requires heavy computing power. Using 32GB RAM on the reservoir section of 1 well dataset 6 hours of runtime. However, this is not a concern for deployment as the relevant features identified in the process will be the inputs used for the models.

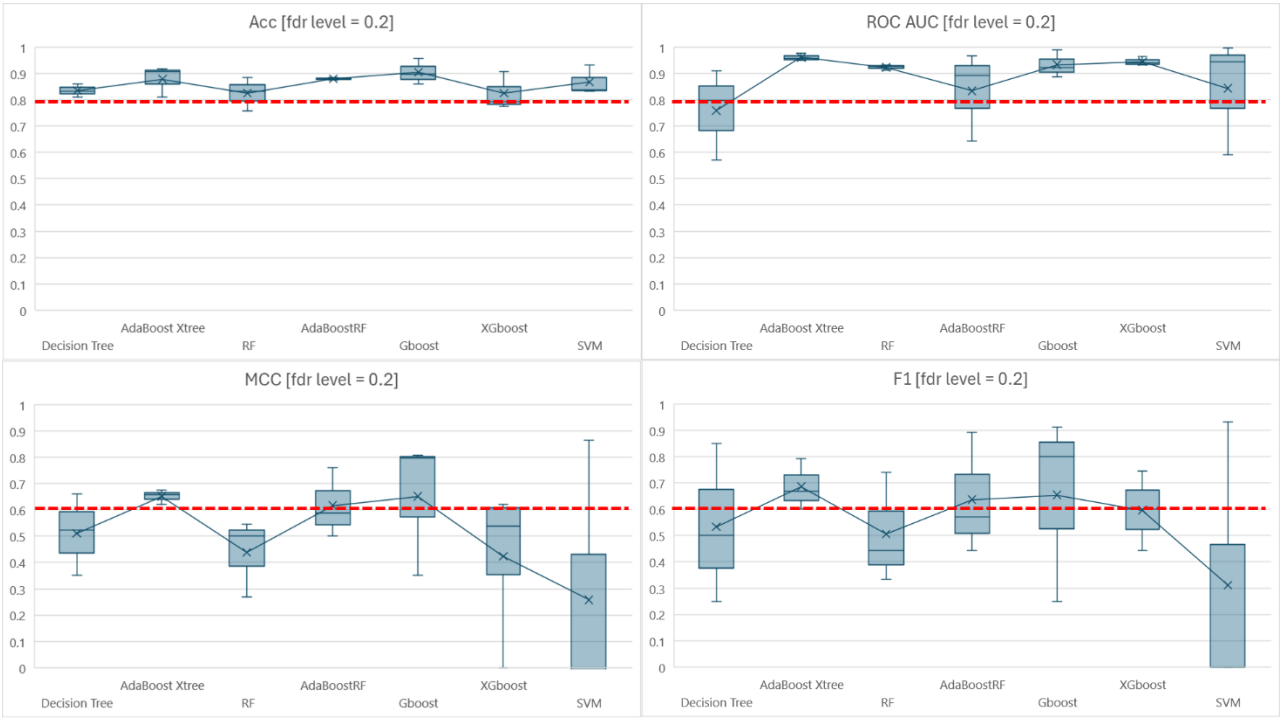


Figure 6 Results of the algorithms tested using different performance metrics show that Decision Tree outperforms all others.

5. CONCLUSION

Several models for incident detection related to hole problems were developed and tested using various machine learning algorithms, including Decision Tree, Random Forest, Gradient Boosting, and Support Vector Machine (SVM). The models were developed using automated feature engineering on a rolling time series using time-series split. Using the four (4) performance metrics, Decision Tree and Adaboosted Random Forest, were deemed acceptable with the former being the best model out of all algorithms.

Based on the outcomes of models developed with small-scale data, there is huge potential for utilizing machine learning to create real-time incident detection systems. These systems can assist drilling personnel in making tactical decisions in geothermal drilling, thereby reducing operational risks and enhancing overall drilling efficiency and cost performance.

An extensive literature review on drilling applications of machine learning was conducted and it is observed that there is a noticeable research gap in the geothermal industry with only three (3) out of 190 studies relating to geothermal. Extensive data collection is essential for any big data analysis, and there has been a significant increase in data acquisition in geothermal drilling over the years. This presents an opportunity to use ML to address the industry's unique drilling challenges like severe losses, high temperatures, and tectonically active geographical locations.

There is great potential in incident prediction based on the reviewed studies; however, results for the evaluation of the different algorithms are inconsistent implying that the ML models for drilling are still in its development phase. Also, the majority of the existing literature used minimal to no feature engineering techniques - none of the studies used systematic and extensive feature extraction and generation approach. While the studies applied training and testing datasets ratio as is typical in ML methodology, none explored a holistic validation approach that is practical in an actual real-time drilling scenario. The real-time incident detection system that will be developed will address some of these gaps while exploring how geothermal data responds to ML models.

REFERENCES

[1] Ouahrani L, Haris AN, Suluru S, Chiha A, Al Fakih A. Invisible Lost Time measurement and reduction contributes to optimizing total well time by improving ROP and reducing flat time. All Days, SPE; 2018. https://doi.org/https://doi.org/10.2118/192319-MS.

[2] API Recommended Practice 65-1. Cementing Shallow-water Flow Zones in Deepwater Wells. Washington, DC: 2018.

[3] Dew LL, Childers MA. MODU Drilling Rig Downtime: An Objective Analytical Approach. SPE/IADC Drilling Conference, SPE; 1989. https://doi.org/https://doi.org/10.2118/18665-MS.

[4] Gehringer M, Loksha V. Geothermal Handbook: Planning and Financing Power Generation. Washington, DC: 2012.

[5] Carson CC. Geothermal Drilling Problems and their Impact on Cost. Am Assoc Pet Geol Bull 1982;66.

https://doi.org/https://doi.org/10.1306/03B5A3B3-16D1-11D7-8645000102C1865D.

[6] Nmegbu CGJ, Ohazuruike L V. Wellbore Instability in Oil Well Drilling: A Review. International Journal of Engineering Research and Development 2014;10:11–20.

[7] Finger J, Blankenship D. Handbook of Best Practices for Geothermal Drilling. Albuquerque: 2012. https://doi.org/https://doi.org/10.2172/1325261.

[8] Varhaug M. THE DEFINING SERIES - Mud logging. Oilfield Review 2015.

[9] Blue D, Blakey T, Rowe M. Advanced Mud Logging: Key to Safe and Efficient Well Delivery. Day 3 Wed, May 08, 2019, OTC; 2019. https://doi.org/https://doi.org/10.4043/29469-MS.

[10] Standards New Zealand. NZS 2403:2015 Code of practice for deep geothermal wells. Wellington, NZ, New Zealand: Standards New Zealand; 2015.

[11] Gurina E, Klyuchnikov N, Antipova K, Koroteev D. Forecasting the abnormal events at well drilling with machine learning. Applied Intelligence 2022;52. https://doi.org/https://doi.org/10.1007/s10489-021-03013-x.

[12] Gurina E, Klyuchnikov N, Zaytsev A, Romanenkova E, Antipova K, Simon I, et al. Application of machine learning to accidents detection at directional drilling. J Pet Sci Eng 2019;184. https://doi.org/https://doi.org/10.1016/j.petrol.2019.106519.

[13] Gurina E, Klyuchnikov N, Antipova K, Koroteev D. Making the black-box brighter: Interpreting machine learning algorithm for forecasting drilling accidents. J Pet Sci Eng 2022;218. https://doi.org/https://doi.org/10.1016/j.petrol.2022.111041.

[14] Antipova K, Klyuchnikov N, Zaytsev A, Gurina E, Romanenkova E, Koroteev D. Data-Driven Model for the Drilling Accidents Prediction. SPE Annual Technical Conference and Exhibition, vol. 2, SPE Annual Technical Conference and Exhibition; 2019, p. 195888. https://doi.org/https://doi.org/10.2118/195888-MS.

[15] Shadizadeh SR, Karimi F, Zoveidavianpoor M. Drilling Stuck Pipe Prediction in Iranian Oil Fields: An Artificial Neural Network Approach. Iranian Journal of Chemical Engineering 2010;7.

[16] Al-Baiyat I, Heinze L. Implementing Artificial Neural Networks and Support Vector Machines in Stuck Pipe Prediction. SPE Kuwait International Petroleum Conference and Exhibition, 2012.

[17] Jahanbakhshi R, Keshavarzi R, Shoorehdeli A, Emamzadeh A. Intelligent Prediction of Differential Pipe Sticking by Support Vector Machine Compared With Conventional Artificial Neural Networks: An Example of Iranian Offshore Oil Fields. SPE Drilling and Completion 2012.

[18] Murillo A, Neuman J, Samuel R. Pipe Sticking Prediction and Avoidance Using Adaptive Fuzzy Logic and Neural Network Modeling. 2009.

[19] Rostami H, Manshad AK, Wajnikonis CJ. A New Support Vector Machine and Artificial Neural Networks for Prediction of Stuck Pipe in Drilling of Oil Fields. J Energy Resour Technol 2014;136. https://doi.org/10.1115/1.4026917].

[20] Abbas AK, Almubarak H, Abbas A, Dawood J. Application of Machine Learning Approach for Intelligent Prediction of Pipe Sticking. Abu Dhabi International Petroleum Exhibition & Conference, 2019.

[21] Elmousalami HH, Elaskary M. Drilling stuck pipe classification and mitigation in the Gulf of Suez oil fields using artificial intelligence. J Pet Explor Prod Technol 2020;10. https://doi.org/10.1007/s13202-020-00857-w.

[22] Khan JA, Irfan M, Irawan S, Yao FK, Rahaman MSA, Shahari AR, et al. Comparison of Machine Learning Classifiers for Accurate Prediction of Real-Time Stuck Pipe Incidents. Energies (Basel) 2020;13. https://doi.org/10.3390/en13143683.

[23] Tsuchihashi N, Wada R, Ozaki M, Inoue T, Mopuri K, Bilen H, et al. Early Stuck Pipe Sign Detection with Depth-Domain 3D Convolutional Neural Network Using Actual Drilling Data. SPE Journal 2021.

[24] Hunter JD. Matplotlib: A 2D Graphics Environment. Comput Sci Eng 2007;9:90–5. https://doi.org/https://doi.org/10.1109/MCSE.2007.55.

[25] McKinney W. Data structures for statistical computing in Python. Proceedings of the Python in Science Conference, SciPy; 2010. https://doi.org/https://doi.org/10.25080/majora-92bf1922-00a.

[26] Van Rossum G, Drake F. Python 3 Reference Manual. Scotts Valley, California: CreateSpace; 2009.

[27] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 2011;12:2825–30. https://doi.org/https://doi.org/10.5555/1953048.2078195.

[28] Christ M, Braun N, Neuffer J, Kempa-Liehr AW. Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). Neurocomputing 2018;307:72–7. https://doi.org/https://doi.org/10.1016/j.neucom.2018.03.067.

[29] Wang C, Baratchi M, Bäck T, Hoos HH, Limmer S, Olhofer M. Towards Time-Series Feature Engineering in Automated Machine Learning for Multi-Step-Ahead Forecasting. The 8th International Conference on Time Series and Forecasting, Basel Switzerland: MDPI; 2022. https://doi.org/10.3390/engproc2022018017.

[30] Lubba CH, Sethi SS, Knaute P, Schultz SR, Fulcher BD, Jones NS. catch22: CAnonical Time-series CHaracteristics. Data Min Knowl Discov 2019;33:1821–52. https://doi.org/10.1007/s10618-019-00647-x.

[31] Fulcher BD, Jones NS. hctsa : A Computational Framework for Automated Time-Series Phenotyping Using Massive Feature Extraction. Cell Syst 2017;5:527-531.e3. https://doi.org/10.1016/j.cels.2017.10.001.

[32] Hutter F, Kotthoff L, Vanschoren J. Automated Machine Learning. Cham: Springer International Publishing; 2019. https://doi.org/10.1007/978-3-030-05318-5.

[33] Kanter JM, Veeramachaneni K. Deep feature synthesis: Towards automating data science endeavors. 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2015, p. 1–10. https://doi.org/10.1109/DSAA.2015.7344858.

[34] Seshadri R. GitHub - AutoViML/featurewiz: Use advanced feature engineering strategies and select the best features from your data set fast with a single line of code 2020. https://github.com/AutoViML/featurewiz.

[35] Moez Ali. PyCaret: An open source, low-code machine learning library in Python 2020. https://www.pycaret.org.

[36] Breiman L. Classification and regression trees. Routledge; 2017.

[37] Breiman L. Random forests. Mach Learn 2001;45:5–32.

[38] Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat 2001:1189–232.

[39] Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 1997;55:119–39.

[40] Christianini N, Shawe-Taylor J. Support vector machines and other kernel-based learning methods. Cambridge UP 2000.

[41] Mitchell TM. Machine learning. United States of America: The McGraw-Hill Companies; 1997.

[42] Zhou Z-H. Machine Learning. English Translation. Singapore: Springer Nature Singapore Pte Ltd; 2021.

[43] Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. Inf Process Manag 2009;45:427–37.

[44] Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. Inf Process Manag 2009;45:427–37. https://doi.org/https://doi.org/10.1016/j.ipm.2009.03.002.

[45] Provost F, Fawcett T. Robust classification for imprecise environments. Mach Learn 2001;42:203–31.

[46] Matthews BW. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta (BBA)-Protein Structure 1975;405:442–51. https://doi.org/https://doi.org/10.1016/0005-2795(75)90109-9.