

Article

Comparison of Machine Learning Classifiers for Accurate Prediction of Real-Time Stuck Pipe Incidents

Javed Akbar Khan ^{1,*}, Muhammad Irfan ^{2,*}, Sonny Irawan ³, Fong Kam Yao ¹, Md Shokor Abdul Rahaman ⁴, Ahmad Radzi Shahari ¹, Adam Glowacz ^{5,*} and Nazia Zeb ⁶

¹ Petroleum Engineering Department and Shale Gas Research Group, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia; yao.fong92@gmail.com (F.K.Y.); ahmadradzi.shahari@utp.edu.my (A.R.S.)

² College of Engineering, Electrical Engineering Department, Najran University, Najran 61441, Saudi Arabia

³ School of Mining & Geosciences, Nazarbayev University, Nur-Sultan City 010000, Kazakhstan; irawan.sonny@nu.edu.kz

⁴ Fundamental and Applied Science Department, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia; md_17009796@utp.edu.my

⁵ Department of Automatic, Control and Robotics, AGH University of Science and Technology, 30-059 Kraków, Poland

⁶ Department of Computer and Information Sciences, University of Management & Technology, Lahore 55150, Pakistan; nazia.zeb.001@post.umt.edu.pk

* Correspondence: javedakbar.khan@utp.edu.my (J.A.K.); miditta@nu.edu.sa (M.I.); adglow@agh.edu.pl (A.G.)

Received: 7 June 2020; Accepted: 15 July 2020; Published: 17 July 2020



Abstract: Stuck pipe incidents are one of the contributors to non-productive time (NPT), where they can result in a higher well cost. This research investigates the feasibility of applying machine learning to predict events of stuck pipes during drilling operations in petroleum fields. The predictive model aims to predict the occurrence of stuck pipes so that relevant drilling operation personnel are warned to enact a mitigation plan to prevent stuck pipes. Two machine learning methodologies were studied in this research, namely, the artificial neural network (ANN) and support vector machine (SVM). A total of 268 data sets were successfully collected through data extraction for the well drilling operation. The data also consist of the parameters with which the stuck pipes occurred during the drilling operations. These drilling parameters include information such as the properties of the drilling fluid, bottom-hole assembly (BHA) specification, state of the bore-hole and operating conditions. The R programming software was used to construct both the ANN and SVM machine learning models. The prediction performance of the machine learning models was evaluated in terms of accuracy, sensitivity and specificity. Sensitivity analysis was conducted on these two machine learning models. For the ANN, two activation functions—namely, the logistic activation function and hyperbolic tangent activation function—were tested. Additionally, all the possible combinations of network structures, from [19, 1, 1, 1, 1] to [19, 10, 10, 10, 1], were tested for each activation function. For the SVM, three kernel functions—namely, linear, Radial Basis Function (RBF) and polynomial—were tested. Apart from that, SVM hyper-parameters such as the regularization factor (C), sigma (σ) and degree (D) were used in sensitivity analysis as well. The results from the sensitivity analysis demonstrate that the best ANN model managed to achieve an 88.89% accuracy, 91.89% sensitivity and 86.36% specificity, whereas the best SVM model managed to achieve an 83.95% accuracy, 86.49% sensitivity and 81.82% specificity. Upon comparison, the ANN model is the better machine learning model in this study because its accuracy, sensitivity and specificity are consistently higher than those of the best SVM model. In conclusion, judging from the promising prediction accurateness as demonstrated in the results of this study, it is suggested that stuck pipe prediction using machine learning is indeed practical.

Keywords: artificial neural networks; drilling operation; machine learning classifiers; RBF Kernel function; stuck pipe; support vector machines; sensitivity analysis

1. Introduction

A stuck pipe can be defined as the inability to retrieve a drill string by means of pulling it out of the bore-hole even though the drill string does not lose its ability to move down, rotate and circulate partially or freely [1]. Other definitions of a stuck pipe include a pipe unable to be moved and where extra efforts such as jarring and spotting fluid placement have been started [2]. Studies have been pioneered by Hayward as far back as the year 1937 to investigate the cause and of solution for stuck pipes and casing [3]. Stuck pipes can have a huge impact on non-productive time (NPT), where drilling operation progress is hindered, resulting in higher well costs. NPT can be defined as an event described that causes the drilling operation to stop [4]. Some studies define NPT as time lost due to flat time that is caused by problems such as wellbore problems and equipment failures [5]. Flat time can be understood as time allocated to carry out activities that does not increase bore-hole depth. According to a study conducted by Saudi Aramco in the year 2012, stuck pipes account for 25% of the NPT or the equivalent of two rig-years' worth of additional cost every year [6]. During a drilling company operation in Latin America, NPT is used to benchmark the severity of stuck pipe incidents [7]. Stuck pipe mechanisms can be categorized into two major types in general: differential sticking and mechanical sticking. Mechanical sticking can be due to various issues such as pack-off and bridging as well as wellbore geometry [1]. These two mechanisms can be triggered by factors such as poor bore-hole cleaning, poor mud rheology, wellbore instability, improper drilling practices, the quality of the mud-cake and excessive reaming or back-reaming [8]. It is essential to understand the theory and factors involved that drive these mechanisms so that appropriate actions can be taken to reduce the risk of stuck pipes. Recently, computational models have been proposed for the early detection of stuck pipe incidents to minimize or avoid the NTP due to stuck pipes [9].

The stuck pipe data set used to train the predictive model is focused on stuck pipe incidents that have been recorded in the well drilling operations database. The database includes all the drilling operation data for the wells drilled from the year 2008 to the year 2018. The recording and storage of the data on the incidents of stuck pipes for this period are easily accessible as well as more complete and organized. Subject to the quality and completeness of the data, part of the data set is filtered during data cleaning and processing before being integrated into the formation of the predictive model. The data set collected on stuck pipes is only supported for stuck pipe incidents occurring in Malaysian fields. The predictive model must be trained using a data set for the same fields so that its predictive ability to predict stuck pipe incidents can be presented. Machine learning algorithms are able to better discover the regional trend if the training data set comes from the same region. A preliminary literature review suggests several statistical analyses and machine learning models that are suitable to be used in developing the stuck pipe prediction model. Among the statistical analyses are logistic regression and discriminant analysis. On the other hand, the artificial neural network (ANN) and support vector machine (SVM) are also machine learning methodologies that are suitable to be used to predict stuck pipes. Since the purpose of this research was to focus on machine learning models, the ANN and SVM machine learning methodologies were to be used to generate the stuck pipe prediction model. The R programming software was used to develop different predictive models from different machine learning algorithms such as the artificial neural network (ANN) and support vector machine (SVM). The R language is widely used among statisticians and data miners for developing statistical software [10] and data analysis [11]. R programming is chosen to create the machine learning model because the software is free and open-source. Moreover, many software developers have been devoted to developing various code package libraries, including machine learning packages that can

be easily loaded into the software to improve its features and functionality. Other than that, the R programming software can read data sets and write output in Excel format, which is easy to work with.

Research on predicting stuck pipes using the SVM has been performed [12]. The researchers used a modified version of the SVM called the least squares-SVM (LS-SVM), which was claimed to be able to reduce run time and show more adaptivity. Besides that, they also used the coupled simulated annealing (CSA) optimization technique to fine tune the SVM's hyper-parameters, which were the regularization factor (C) and kernel bandwidth (σ^2). In the present research, an SVM base case model was built using the "vanilladot" kernel, a linear kernel function. The regularization factor (C), sigma (σ) and degree (D) were varied in the SVM model for sensitivity analysis. In the ANN model case, two different activation functions, which are the logistic activation function and hyperbolic tangent activation function, were used to predict the accuracy.

2. Literature Review

The idea of stuck pipes caused by differential sticking was first proposed by Shell engineers Helmick and Longley in their research in the year 1957 [13]. They managed to demonstrate the effect of differential sticking in a laboratory experiment. Their result revealed that the force required to pull out a pipe proportionally increases with differential pressure. Differential pressure is defined as the pressure difference between the hydrostatic pressure of the drilling fluid and the formation pressure. Differential sticking is caused by wellbore conditions where the differential pressure is present at the wellbore area together with a thick and permeable mud cake across a permeable formation. The ratio of the pipe to bore-hole diameter can have an effect on the initial contact area between the drill string and mud cake [13]. The contact area increases over time if the drill string remains static. This is due to the mud cake formed caused by the continued filtration of mud into the formation [14]. Hence, the contact area is indirectly governed by filtrate loss and solid percentage in the mud [13]. In some cases, during well-cleanup operations with coiled tubing, the sticking of the coiled tubing occurs, especially during sand transport from horizontal wellbores [15,16]. A computational fluid dynamics approach has been used to circulate the injection of low quality foam at a high flow rate to successfully clean the wellbore without the sticking of the coiled tubing [17,18]. Moreover, an optimized design of the coiled tubing nozzle has been presented to increase the outlet pressure for higher sand removal from the wellbore to avoid the sticking of the tubing [19]. Compared to existing stuck pipe prediction approaches, the new proposed approach can distinguish the well-cleanup-related stuck pipe risk and other reasons for stuck pipes [20]. Recently, research on an oil drilling system with a stick-slip phenomenon based on wave Partial Differential Equation (PDE) models was carried out to design an adaptive controller, which is used in torsional vibration suppression for an oil drilling system with uncertain stick-slip instability and disturbances at the drilling bit [21]. A newly developed BN model is capable of probability adapting and probability updating including predictive analysis and diagnostic analysis. By sensitivity analysis, key influencing factor can be determined [22]. Another study describes the blowout scenarios and important risk factors that should be taken into account for blowout risk assessment [23]. Similarly the risk and time assessments of the pipe-stuck incidents are required to overcome non-productive time due to human errors, mechanical failure or with other petro physical properties.

Siruvuri et al. (2006) developed a convolutional neural network to predict stuck pipes in their study [24]. The study used 120 differential stuck cases and 50 non-stuck cases as data. The results suggested that the neural network model was able to predict non-stuck cases with 100% accuracy but had only 10% accuracy for predicting stuck cases. Miri et al. (2007) also worked on the artificial neural network to predict differential stuck pipes in Iranian Offshore Oil Fields [25]. The study tried to compare the predictive ability of two different types of neural network models, namely, the Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF). The results from the study revealed that the MLP model was able to predict stuck cases with 83.33% accuracy and non-stuck cases with 87.5% accuracy. In the past, a study was conducted to evaluate the selection of the parameters in the SVM with the RBF

kernel function [26]. They presented a double linear grid search method that can give high precision with a small amount of training. A study on satellite images with Neural Networks and Support Vector Machines Using the R/rminer Tool was carried out, and competitive results were obtained, in particular, with the SVM model for the classification tasks and the NN for the regression ones [27]. It has been claimed that the SVM performs better than the learning statistical theory [28]. The support vector machine (SVM) has been applied to predict differential stuck pipe events in Iranian Offshore Oil Fields [29]. In fact, the study compared the stuck pipe prediction accuracy between the support vector machine model and artificial neural network model. Both models were used to categorize cases as stuck and non-stuck. A total of 241 data sets from 63 wells were gathered in this study. Differential sticking occurred in 32 wells, and the rest were non-stuck wells. In this study, 70% of the data sets were assigned for model training purposes while the remaining 30% were assigned for testing purposes. The results indicated that the neural network model with the logsig transfer function and tansig transfer function were able to achieve 76.56% and 82.81% accuracies, respectively. On the other hand, the SVM with the polynomial kernel function and Gaussian kernel function were able to achieve 71.88% and 92.19% accuracies, respectively. In another study, it was concluded that the SVM approach was more applicable than the neural network [29]. In the present study, the best machine learning configurations are presented for both the ANN and SVM models. The objectives of this research included 1. investigating the feasibility of applying machine learning to predict stuck pipe events during drilling operations; 2. comparing the differences between several machine learning methodologies when applied for stuck pipe prediction in the petroleum industry; and 3. determining the best adjustable parameters to fine tune the machine learning model to achieve higher stuck pipe prediction accuracy. Overall, the aim of this study was to create a predictive model that could be used to predict the risk of stuck pipes.

3. Methodology

The objective of this research primarily focused on creating a stuck pipe prediction computer model using machine learning to evaluate stuck pipe risks during a drilling operation. This research intended to build prediction models using two machine learning algorithms, namely, the artificial neural network (ANN) and support vector machine (SVM). The final models were expected to predict cases as stuck or non-stuck. The prediction capabilities and accuracies of the generated models were to be compared to determine the best prediction model. The ultimate goal was to use the prediction model to reduce the occurrence of stuck pipe events and hence the non-productive time (NPT). Lowering the NPT would have a positive impact for controlling the drilling operation costs. There are three types of machine learning, namely, supervised learning, unsupervised learning and reinforcement learning. The nature of this research fulfills the criteria of supervised learning. The data sets collected were labelled as either stuck or non-stuck cases. With the labelled data, the machine learning algorithms (ANN and SVM) would try to uncover the underlying pattern of stuck pipes by examining the data sets. During the training process, direct feedback was given to the model to indicate whether the predicted results from the model matched with the actual cases. The model took note of incorrect predictions and reiterated the training process to improve the prediction accuracy. The finalized model was expected to be able to predict stuck pipe events with the inputting of new data sets. Below is the detail of the planned steps in conducting this study. ANN models are effective in handling data with noise, mimic neural networks, perform better than regression based models and accurately predict highly non-linear parameters [30]. In the past, statistical/regression-based models have shown accurate prediction for a lower number of parameters having linearity [31,32]. As the present study involved a large number of non-linear parameters, the advanced statistical approach of machine learning was adopted to accurately predict stuck pipes to reduce the non-productive time (NPT).

3.1. Data Sourcing and Gathering

Machine learning is usually associated with the term “big data”, a term that refers to large volumes of data. Previously, people extracted useful information from these big data through classification and sequence analysis. However, the emergence of machine learning has allowed users to tap big data and identify hidden patterns and unknown correlation unbiasedly. Machine learning can train a predictive model by taking advantage of the sheer volume of data. This translates to a more generalized trained model that is able to predict different scenarios. In this research, the stuck pipe data set was collected using the Data Analyzer software. The software can be used to perform data querying to systematically pull data from the database. The database contains all the drilling operation data, including but not limited to daily drilling report operation details; daily drilling fluid properties; the Bottom Hole Assembly (BHA) specification; bore-hole conditions such as the inclination, azimuth, bore-hole diameter, Measured Depth (MD) and True Vertical Depth (TVD); the casing Outer Diameter (OD) and Inner Diameter (ID); and operating conditions such as the flow rate, rotational speed and rate of penetration (ROP). Instead of going through all the well documents such as the Notice of Operation (NOOP), Daily Drilling Report (DDR), Final Well Report (FWR), well trajectory and Stuck Pipe Review Report manually to extract the necessary information, pulling data using Data Analyzer is deemed more efficient and less time consuming. On top of that, most of the aforementioned well documents are not readily available and required repeat browsing through various sources such as shared common drives or the Well Files Center to collect them. With that in mind, the data extraction in this study was performed mainly by utilizing the Data Analyzer software. The documents from the wells were used as a secondary source for data for quality assurance/quality control (QA/QC) and cross-checking should there have been any discrepancy among the data collected with the Data Analyzer software. Apart from that, missing data could also be filled in using the information from the well documents.

The data gathering process was divided into two parts, which were gathering stuck data and gathering non-stuck data. Stuck data collection began with obtaining the list of the well names that contained the “PFP” code in the activity codes column. An activity code was assigned for every well activity recorded inside a daily drilling report. An activity code was used to categorize the type of well activity. In the case of stuck pipes or stuck tools, the “PFP” activity code, which was categorized under NPT activity codes, was to be assigned to that particular well activity. By searching for all the wells that contained the “PFP” activity code, a list of wells that experienced stuck pipes/tools could be identified. The activity code “PFP” was classified under the “HOLP” impact code, which represented NPT due to a “Bore-hole Problem”. The next step was gathering all the predetermined drilling parameter data that were related to the causes of stuck pipes based on the well names on the list. However, it should be noted that not all the wells that contained the “PFP” code experienced stuck pipes due to drilling operations that employed “unsuitable” drilling parameters that potentially led to stuck pipes. For example, some of the “PFP” activity codes highlighted stuck tools while wire line operations were being run. These kind of data were filtered out, since our research focus was on the drilling-related stuck case. Table 1 presents all the well activities with the “PFP” activity code that were filtered out and would not be collected as part of the data sets.

Table 1. Well activities with the “PFP” activity code that were filtered out.

No. of Activity Code of Stuck Pipe	Well Activity	Justification
1	Stuck pipe due to formation loss (loss circulation, no return on surface)	Unpredictable event (loss formation)
2	Stuck pipe due to well control issue	Pipe stuck due to excessive time of pipe not moving during restoration of well control
3	Stuck pipe due to waiting on weather	Pipe stuck due to excessive time of pipe not moving during waiting on weather
4	Wire line tool stuck	Not drilling-related stuck case
5	Run in bore-hole casing/liner stuck	Not drilling-related stuck case
6	Cannot retrieve casing after casing cut	Not drilling-related stuck case
7	Completion assembly stuck	Not drilling-related stuck case
8	No stuck case upon checking well activity operation summary	Wrongly assigned “PFP” activity code on well activity that is not stuck pipe

After confirming that a well had encountered a drilling-related stuck pipe, the drilling parameters that the well was subjected to were collected. In order to capture the drilling condition that potentially caused the stuck pipe case, the drilling parameters on the stuck day, which is for the first “PFP” activity code, and the drilling parameters 2 days prior to the stuck pipe event were collected. In total, 3 days’ worth of drilling data were collected for any particular stuck pipe case that was drilling related. However, during the process of data collection, it was observed that a lot of the drilling data on the day of the first “PFP” activity codes were not complete or irrelevant. If a stuck pipe occurred relatively late that day, it was possible for a full set of drilling data to be obtained. However, if the stuck pipe occurred relatively early on that day, the drilling data may have been irrelevant because the recorded data were the drilling parameters that the well experienced during the efforts to free the stuck pipe, which are not from drilling operation. Hence, for any stuck pipe case, the number of data sets could range from one to three sets.

Using the Data Analyzer software, the drilling parameters data sets were extracted out batch by batch using several different queries. The initial idea was to extract all the drilling parameters in a single query so that all the data could be presented in a single table. This could facilitate the processing of the data sets later on as one could omit the process of cross checking within several tables of data. However, upon closer examination of the data structure of the drilling database, it was suggested that a single query could not be used to extract all the drilling parameters in single table. For example, a single day of daily drilling report (DDR) may contain up to 30 activity codes, but usually, only a single set of drilling fluid data was available for any single day of drilling operation. Additionally, BHA specification records are not tabulated on a daily basis and are only recorded whenever there is change of BHA during the drilling operation. Hence, it was rather difficult to extract all of the data at once in just a query. The way forward in light of this challenge was to extract the data separately through several queries. Table 2 shows the summarized drilling parameters recorded in the master list. This list shows the initial proposed drilling parameters to be collected. Based on the data quality and data completeness, the data quality pass column assigns a “Y” or “N” to label each row of data.

Table 2. Drilling parameters recorded in the master list.

No.	Data Column	No.	Data Column
1	Well Common Name	15	pH
2	Report date	16	MW (Mud Weight) (ppg)
3	YP (Yield Point) (lbf/100 ft ²)	17	OH diameter (Open Bore-hole) (in)
4	PV (Plastic Viscosity) (cp)	18	Drill Collar OD (in)
5	Gels 10 s (lbf/100 ft ²)	19	BHA length (ft)
6	Gels 10 min (lbf/100 ft ²)	20	True Vertical Depth (TVD) (m)
7	Filtrate (API) (cc/30 min)	21	Measured Depth (MD) (m)
8	Filtrate (HTHP) (cc/30 min)	22	Inclination (°)
9	Mud type	23	Current RPM (rpm)
10	Calcium (ppm)	24	Rate of Penetration (ROP) (ft/h)
11	Cl- (ppm)	25	Average RPM (rpm)
12	Total solids (%)	26	Flow rate (gpm)
13	H ₂ O % (%)	27	Stuck? (Y/N)
14	Oil % (%)	28	Data quality pass (Y/N)

For gathering non-stuck data, the “DRL” activity code is used. The “DRL” activity code is assigned to well activities that are related to drilling or sidetrack drilling. Any well that does not have the “PFP” activity code could be a target well for gathering non-stuck data. Since the data scarcity lay in stuck data and not with the non-stuck data, greater flexibility could be allowed in gathering non-stuck data. The stuck data were gathered from any wells in the database of an identified drilling operator that bore the “PFP” activity code regardless of the year of drilling, since the stuck cases were rarer than the non-stuck case. To the contrary, the non-stuck data were to be gathered from the wells with no “PFP” activity code, which were drilled from the year 2016 to the year 2019. This is because the data quality

and completeness are generally improved with the latest data record as observed during the process of gathering non-stuck data. Furthermore, data records that were not complete or doubtful could be simply filtered out, as abundant non-stuck data were collected. In the case of stuck data, 3 days' worth of drilling data prior to and during the pipe stuck event were taken, but one drilling data set was taken for each bore-hole section for non-stuck data. For example, a well that drilled 17.5", 12.25", 8.5" and 6" bore-hole sections would have four data records. The purpose behind this was to have more data variety included in the data sets so that the machine learning model could be trained to be more generalized in terms of its predictive capability. At the end of data gathering, a total of 111 wells are identified as having the "PFP" activity code. There were 50 wells remaining after filtering out the wells that had non-drilling-related "PFP" activity codes. Stuck data were to be collected from these 50 wells. On the other hand, for the non-stuck data, a total of 82 wells that were drilled between the year 2016 and year 2019 were identified as target wells. As mentioned before, a few data rows could be extracted from any particular well, be it a stuck or non-stuck well. The final number of a data row could only be determined after the collected data were subjected to a data cleaning process, where the data integrity, quality and completeness were examined.

3.2. Data Review, Cleaning and Preparation

After extracting stuck and non-stuck data with the Data Analyzer software, the drilling parameter data rows were tabulated in Excel format. Although some basic filtering was performed on the raw data to remove unqualified data, the extracted data values were not thoroughly examined. Hence, the next step was to check for data quality and completeness. The percentage of data completeness was measured using Equation (1).

$$\text{Data Completeness (\%)} = \frac{\text{Number of cells with data value}}{\text{Total number of data row}} \times 100 \quad (1)$$

Upon checking for overall data completeness, it was found that several data columns such as "calcium (ppm)" and "Cl- (ppm)" had relatively low data completeness. Their data completeness was measured to be 38.89% and 61.11%, respectively, for stuck data and 19.78% and 26.92%, respectively, for non-stuck data. Hence, it was decided that these two data columns would not be dropped as inputs for machine learning training because the drilling parameters were no longer representative with such high missing data points. Apart from that, the data completeness of "Filtrate, American Petroleum Institute (API) (cc/30min)" and "Filtrate (API) (cc/30min)" were also not high, comparatively. Their data completeness was measured to be 65.87% and 55.56%, respectively, for stuck data and 34.62% and 75.27% respectively, for non-stuck data. However, upon further investigation, it was discovered that for most of the synthetic based muds (SBM) used in drilling, only high-temperature high-pressure (HPHT) filtrate loss tests were conducted, not API filtrate loss tests. The same happened for WBM drilling mud, where only API filtrate loss tests were generally conducted and not HPHT filtrate loss tests. This explained their low data completeness. In order to rectify this problem, it was decided to combine these two data columns into one. For SBM drilling mud, HPHT filtrate loss was used, while for WBM drilling mud, API filtrate loss was used to populate the combined data column. Although putting API and HPHT filtrate test values into a single data column seems to be counter-intuitive because their lab testing parameters are different, their difference would be captured by the machine learning model training with the introduction of a data column that indicated the type of mud used. Next, a check was performed on the solid %, water % and oil % data column. In an ideal situation, these three columns should add up to 100%. Some of the data rows record solid % separately from water % and oil %, which results in total values exceeding 100%. An example of a defective data record may have 23% solids, 70% water and 30% oil. In this case, Equations (2) and (3) were applied on the water % and oil % to scale them proportionally so that the sum of solid %, water % and oil % was 100%.

$$\text{scaled water\%} = (100 - \text{solid\%}) \times \text{water\%}/100 \quad (2)$$

$$\text{scaled oil\%} = (100 - \text{solid\%}) \times \text{oil\%}/100 \quad (3)$$

Besides that, it is worth mentioning that TVD was excluded as an input for machine learning model training. This is because the database of the operator has data corruption in all the depth-related data such as for the MD and TVD. Any attempt to query depth-related data with the Data Analyzer software caused the software to crash. At the beginning of data collection using the Data Analyzer software, it was difficult to troubleshoot the problem of the software crashing. Eventually, it was possible to pin down the root cause of the software crashing, which was the querying of corrupted depth-related data. Although the depth-related data could not be extracted directly from the Data Analyzer software, we were able to compile the data column for "MD" by manually reading through daily operation summaries extracted with the Data Analyzer software; the mentioned drilled depth could be taken as the "MD" data point. The daily operation summary usually only mentioned the drilled depth (MD) and lacked TVD information. This resulted in rather low data completeness in the TVD data column, with 24.6% for stuck data and 2.75% for non-stuck data. Despite best efforts to extract all the depth-related data, only the "MD" data column had the necessary data completeness to fulfill the machine learning model training requirements. The lack of TVD data might not be as critical where the inclusion of MD and inclination data are able to compensate the loss of TVD information to a certain extent. MD was extracted from the operation summary in the Daily Drilling Report (DDR). The DDR is in PDF format whereas the Excel sheet is the output format for queries from the Data Analyzer software. Furthermore, the differential pressure data column was excluded from the training inputs list because the key components to calculate it were absent. Differential pressure can be calculated by finding the difference between the hydrostatic pressure of the drilling fluid and the formation pressure. Drilling mud hydrostatic pressure can be calculated using Equation (4).

$$\text{Hydrostatic pressure (psi)} = 0.052 \times \text{Mud weight (ppg)} \times \text{TVD (ft)} \quad (4)$$

Since TVD data were not available, the drilling mud hydrostatic pressure could not be calculated. Moreover, the formation pressure measurement was not taken at all depths during drilling operation. The only possible way to obtain the formation pressure at any particular depth is to get it from the Pore Pressure Fracture Gradient (PPFG) graph, which is at best only an estimation. Differential pressure is one of the vital parameters in measuring differential sticking, and it would definitely have been useful to include it in the model training. However, a check on the collected data showed that only 8 out of the 50 wells that experienced stuck pipes had differential sticking. In other words, only 16% of the stuck pipes were due to differential sticking. If the non-stuck data were factored in, the differential sticking data portion was further reduced. The data sets for differential sticking were too small compared to the mechanical sticking and non-stuck data, and hence, the benefit of including differential pressure as a model training parameter was minuscule. This is also the reason why the scope of the machine learning model changed from predicting differential sticking, mechanical sticking and non-stuck cases to predicting stuck and non-stuck cases. Table 3 shows a list of the drilling parameters to be inputted into the training model. Nineteen drilling parameters were finalized after taking the above-mentioned issues into consideration.

Noise and outlier in data sets will introduce errors and interfere with the model training. The training of a model might be skewed, which can result in the inability of the model to predict accurately. Hence, the data values need to be evaluated before they are ready to be used for model training. Data evaluation is performed to check for any erroneous or erratic values that do not make sense from an engineering perspective. For each data column, the inspection starts at the minimum and maximum values because illogical numbers usually appear at the extreme ends of the value ranges. Examples of invalid data values are a 3335 gpm flow rate, 450 rpm rotation speed and 4908 ft BHA length, just to name a few. These unacceptable data values were corrected by cross-checking with other sources of information, primarily, the operation summary. In some cases, invalid data values that were extracted with the Data Analyzer software were removed. The last step of preparing the data sets

was converting qualitative data columns such as mud type and stuck/not stuck condition to binary format so that the data set was readable by the machine learning programming software. For the mud type data column, 1 was assigned to all WBM, while 0 was assigned to all SBM data points. For the stuck/no stuck condition, 1 was assigned to all stuck data, while 0 was assigned to all non-stuck data. At the end of the data review, cleaning and preparation process, 123 rows of stuck data and 145 rows of non-stuck data had been successfully collected. In total, 268 rows of data set were to be used in machine learning model training and testing.

Table 3. Drilling parameters for model training.

No.	Features	Abbreviation	Unit	No.	Features	Abbreviation	Unit
1	Yield Point	YP	1 lbf/100 ft ²	11	pH	PH	* dimensionless
2	Plastic Viscosity	PV	cp	12	Mud Weight	MW	ppg
3	10 Second Gel	GEL1	1 lbf/100 ft ²	13	Open Bore-Hole Diameter	BORE-HOLE	in
4	10 Minute Gel	GEL2	1 lbf/100 ft ²	14	Drill Collar Outside Diameter	DC	in
5	Filtrate (API/HPHT)	FILTR	cc/30 min	15	Bottom-hole assembly (BHA) Length	BHA	ft
6	Mud Type	MUD	* dimensionless	16	Measured Depth	MD	m
7	Rate of Penetration	ROP	ft/hour	17	Well Inclination Angle	INC	degree
8	Solid Percent	SOLID	* fraction	18	Rotation Per Minute	RPM	rotations/min
9	Water Percent	WTR	* fraction	19	Flow Rate Output	GPM	gallons/min
10	Oil Percent	OIL	* fraction	20		STUCK	* dimensionless

* Required output to predict occurrence of pipe stuck based on input parameter.

3.3. Prediction Model Building

Two machine learning methodologies were used in this study, namely, the artificial neural network and support vector machine. The R programming software was used to create the training models. Data sets were extracted with the Data Analyzer software and tabulated in Excel format. The data sets were subjected to data quality checking to minimize noise in the data. Each data set was labeled to distinguish between stuck and non-stuck cases. The data sets were saved in Excel format then loaded into the R programming software.

3.3.1. Data Normalization

Data normalization was performed on the data sets as part of the data pre-processing step. The objective of normalizing the data was to change the values of the numeric columns in the data set to use a common scale, without distorting the differences in the ranges of the values or losing information. This avoided creating new values and maintained the general distribution and ratios in the source data, while keeping the values within a scale applied across all the numeric columns used in the model. In the end, the training process would become more efficient. The data normalization process scaled the data values to a range of [0, 1]. The minimum value and maximum value in a data column were converted to 0 and 1, respectively. All the other values in the data column were scaled proportionally and fell somewhere between 0 and 1. However, in the case of the ANN model using the hyperbolic tangent activation function, the data set was normalized to a range of [-1, 1] to compensate the nature of the activation function.

3.3.2. Training/Testing Data Allocation

For both machine learning models, 70% of the data sets were allocated for training and 30% of the data sets, for testing. Thirty percent of the data were taken into account to test the proposed algorithm [29,33,34]. Since there was an imbalance between the stuck and non-stuck data (123 data sets versus 145 data sets), the testing data were formed by combining 30% of the stuck data and 30% of the non-stuck data. The reason behind this was to ensure that a fair share of the different types of data was represented in both the training and testing data. As a result, 37 rows of stuck data and

44 rows of non-stuck data were randomly selected using the function in the R programming software. The selected data were then combined to become testing data, while the remaining data were combined to become training data. In the end, the testing data consisted of 81 rows of data (30.22%) whereas the training data consisted of 187 rows of data (69.78%).

3.3.3. Model-Building Software Packages

Several additional packages were installed into the R programming software library. The add-on functionality package could facilitate the building of the machine learning models. Table 4 summarizes the installed additional packages. These last two packages in Table 4 were prepared with machine learning features and functions that could be called upon during the coding phase of building the ANN and SVM models. Without these packages, the mathematical equations and algorithms behind the machine learning methodologies would have needed to have been developed manually from scratch.

Table 4. Additional packages installed into R programming software.

No.	Additional Packages	Description
1	readxl	Enable user to load data in Excel “xlsx” format instead of Excel “csv” format.
2	RSSQLite, sqldf	Enable user to use SQL statement (a common database query programming language) in R programming. Used in selecting out the remaining data rows that were not selected as testing data.
3	neuralnet, NeuralNetTools	Enable user to access functions used in the building, training and testing of the Artificial Neural Network (ANN) machine learning model.
4	kernlab	Enable user to access functions used in the building, training and testing of the Support Vector Machine (SVM) machine learning model.

3.3.4. ANN Model Building

After categorizing the data sets and installing the necessary functional packages, the next step was to write code to build and configure the training models. Two sets of codes were written for the ANN and SVM models, respectively. For the ANN model, a 3-hidden-layers network topology was chosen, with each hidden layer consisting of a maximum of 10 neurons. All possible combinations of neuron numbers in each layer were to be tested in the ANN model training. The simplest neural network had a [19, 1, 1, 1, 1] topological structure, where layers 1, 2 and 3 had one neuron each. The most complex neural network had a [19, 10, 10, 10, 1] topological structure, where layers 1, 2 and 3 had ten neurons each. A total of 187 rows of training data, each with 20 columns of drilling parameters, were loaded to initiate the model training process. A total of 1000 models ($10 \times 10 \times 10$) were run when the testing of all the possible combinations of neuron numbers in each layer was complete. The purpose of choosing a maximum of [19, 10, 10, 10, 1] network topology was to achieve a balance between the total training time and model complexity. The total training time to finish all the possible combination in the [19, 10, 10, 10, 1] network topology generally hovered around 5 to 6 h. Any attempt to increase the number of neurons in each layer would result in an exponential increase in the training time. The increment in training time can be explained by (1) the increased number of models to be trained. For example, a [19, 11, 11, 11, 1] network topology would have 1331 models to be trained as compared to 1000 models in a [19, 10, 10, 10, 1] network topology. This would result in 33% more models to be trained. (2) As the number of neurons increases in each layer, the model complexity also increases exponentially. More time is needed to be allocated to achieve convergence during the training process. If the model is too complex, the training process may fail to converge in a given number of iteration steps. A base case ANN model was built using the “logistic” activation function. A different activation function was to be used during the model fine tuning process to investigate the

effect of different activation functions on model prediction capability. Figures 1–3 show the topologies of different neural networks having different input parameters, nodes, layers and outputs.

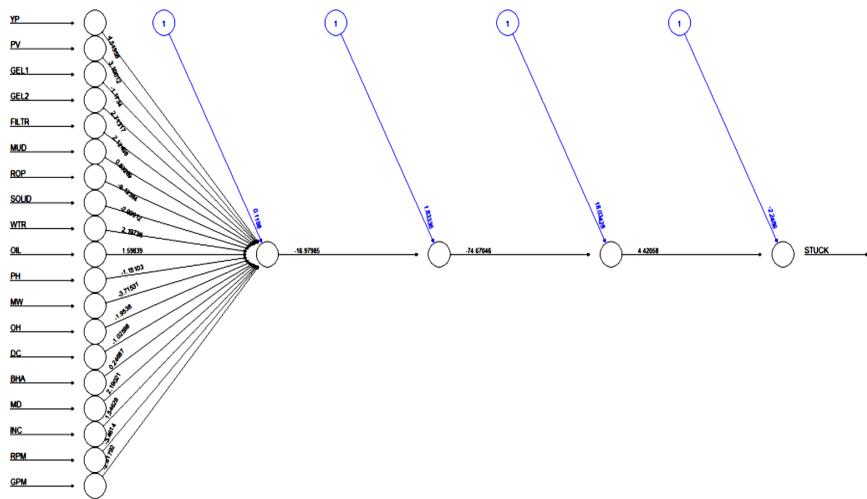


Figure 1. Neural network topology [19, 1, 1, 1, 1].

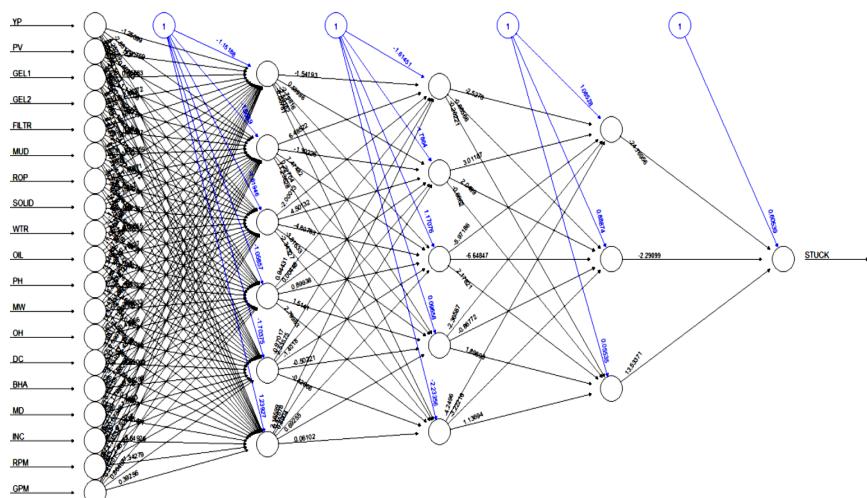


Figure 2. Neural network topology [19, 6, 5, 3, 1].

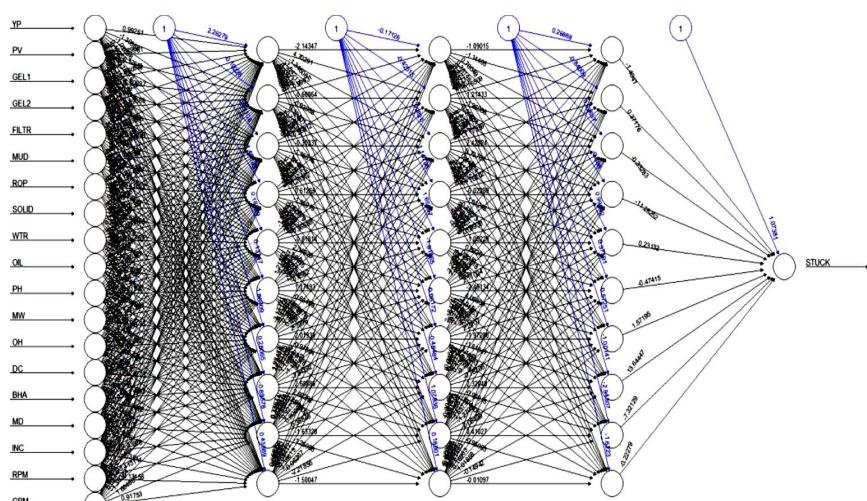


Figure 3. Neural network topology [19, 10, 10, 10, 1].

3.3.5. SVM Model Building

In building the base case for the SVM model, the “kernlab” packages that were installed provided several popular kernel functions that could be used in model training and prediction. Table 5 shows a summary of all the default kernel functions offered by the “kernlab” package. The kernel setting string was the keyword needed to configure the type of kernel used during the training of the SVM model. The “Hyper-parameters” column consisted of all the adjustable parameters for each kernel.

Table 5. Summary of kernel functions offered by the “kernlab” package.

No.	Kernel Name	Kernel Setting Strings	Hyper-Parameters
1	Radial Basis kernel function “Gaussian”	<i>rbfdot</i>	<i>sigma</i>
2	Polynomial kernel function	<i>polydot</i>	<i>degree, scale, offset</i>
3	Linear kernel function	<i>vanilladot</i>	-
4	Hyperbolic tangent kernel function	<i>tanhdot</i>	<i>scale, offset</i>
5	Laplacian kernel function	<i>laplacedot</i>	<i>sigma</i>
6	Bessel kernel function	<i>besseldot</i>	<i>sigma, order, degree</i>
7	ANOVA radial basis function (RBF) kernel function	<i>anovadot</i>	<i>sigma, degree</i>
8	Spline kernel	<i>splinedot</i>	-
9	String kernel	<i>stringdot</i>	<i>length, lambda, type, normalized</i>

An SVM base case model was built using “vanilladot” kernel, a linear kernel function. There is no hyper-parameter to be adjusted in this kernel. However, all SVM models universally allow the adjustment of the regularization factor (C), which governs model optimization in terms of the size of the hyper-plane margin. Together with the hyper-parameters as stipulated in Table 5, a sensitivity analysis could be conducted by using different types of kernel with different combinations of regularization factor (C) and respective hyper-parameters. Once the coding was done, the codes were executed to start the model training process. It took a while for the computer to complete the computational tasks in the training processes. The prediction models were ready to be tested once the training process was complete. Ultimately, two stuck pipe prediction base models were created using the artificial neural network and support vector machine.

3.4. Prediction Model Testing

The testing data sets were inputted into these prediction base models to test their prediction capabilities. Several evaluation metrics were introduced to test the model’s performance in predicting cases as stuck and non-stuck cases. The precedent criteria of the prediction models lay in their ability to differentiate between stuck and non-stuck cases. A confusion matrix of actual events against predicted stuck pipes was developed as shown in Table 6.

Table 6. Confusion matrix of actual events against predicted stuck pipes.

	Actual Events	Model Prediction
True Positive	Stuck Pipe	Stuck Pipe
False Positive	No Stuck Pipe	Stuck Pipe
True Negative	No Stuck Pipe	No Stuck Pipe
False Negative	Stuck Pipe	No Stuck Pipe

Three evaluation metrics—namely, sensitivity, specificity and accuracy—were used to dictate the performance of the prediction base models based on the confusion matrix above. Table 7 summarizes the definition of each evaluation metric. Sensitivity is the ability of the model to say “there is a stuck pipe” when the stuck pipe event did occur. In other words, sensitivity is the accuracy in predicting stuck pipes out of all the stuck pipe events in the testing data. Equation (5) was used for sensitivity.

Specificity is the ability of the model to say “there is no stuck-pipe” when the stuck pipe event did not occur. In other words, specificity is the accuracy in predicting non-stuck cases out of all the non-stuck events in the testing data. Equation (6) was followed for specificity. Accuracy reflects the total proportion of actual stuck pipe events that are correctly predicted or classified, as calculated according to Equation (7).

$$\text{Sensitivity} = \text{True Positive}/(\text{True Positive} + \text{False Negative}) \quad (5)$$

$$\text{Specificity} = \text{True Negative}/(\text{True Negative} + \text{False Positive}) \quad (6)$$

$$\text{Accuracy} = (\text{True Positive} + \text{True Negative})/(\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}) \quad (7)$$

Table 7. Ranges of regularization factor (C), sigma (σ) and degree (D) used in support vector machine (SVM) model sensitivity analysis.

No.	Kernel Function	C	Adjustable Parameters		Number of Cases
			Sigma, σ	Degree, D	
1	Linear kernel function	0.1, 1, 10, 100, 1000, 5000	-	-	6
2	Radial basis kernel function	0.1, 1, 10, 100, 1000, 5000	0.1, 0.25, 0.5, 0.75, 1	-	30
3	Polynomial kernel function	0.1, 1, 10, 100, 1000, 5000	-	1, 2, 3, 4, 5	30

3.5. Prediction Model Fine Tuning

The ANN and SVM prediction base models were subjected to further fine tuning to improve their prediction performance. Sensitivity analysis was conducted on these base models to identify the best configuration of a machine learning architecture. For the ANN model, sensitivity analysis was conducted based on different activation functions. Another ANN model was built using the hyperbolic tangent activation function. The same 3-hidden-layers network topology was used with each hidden layer consisting of a maximum of 10 neurons. All possible combinations of neuron numbers in each layer were tested; hence, another 1000 training models were produced at the end. For the SVM model, sensitivity analysis was conducted by altering the kernel functions from the “vanilladot” Linear kernel function to the Radial Basis kernel function and Polynomial kernel function. Apart from that, the regularization factor (C) and hyper-parameters such as the polynomial degree (D) and sigma (σ) were also adjusted to create a variety of case studies to test for the potential of the SVM model. As for the ANN model, all possible combinations of the regularization factor (C) and hyper-parameters were tested. The ranges of the regularization factor (C), sigma (σ) and degree (D) used in this sensitivity analysis are summarized in Table 7.

In summary, for the linear kernel function SVM model, by varying the regularization factor (C), a total of six models were produced at the end. For the radial basis kernel function SVM model, by varying the regularization factor (C) and sigma (σ) to form different combinations, a total of 30 models were produced. For the polynomial kernel function SVM model, by varying the regularization factor (C) and polynomial degree (D) to form different combinations, a total of 30 models were produced. In total, 66 models were produced for the sake of performing sensitivity analysis on the SVM model. In the ANN model case, each activation function was used to produce 1000 models. Hence, by having two different activation functions, which are the logistic activation function and hyperbolic tangent activation function, a total of 2000 models were produced. At the end of the sensitivity analysis, the prediction results from all these models were collected and tabulated. By examining all the results, the best machine learning configuration could be determined for both the ANN and SVM models. The results also give an insight into the potential for improvement through the fine tuning of the base models.

4. Results and Discussion

4.1. ANN Model Prediction Results

The logistic activation function and hyperbolic tangent activation function were used to produce 1000 ANN models, respectively. A total of 2000 ANN models were intended to be produced upon the finishing of the training process. However, during the training process for the hyperbolic tangent ANN model, 30 models failed to achieve convergence to finish the training process. This translated to a 3% model training failure rate for the 1000 training models. Hence, in the hyperbolic tangent ANN model, only 970 models successfully finished the training process and were ready to be tested for their prediction accuracy. The network topologies of the failed models are recorded in the Table 8.

Table 8. Network topologies of the failed models in the hyperbolic tangent artificial neural network (ANN) model.

No.	Number of Neurons			No.	Number of Neurons			No.	Number of Neurons		
	Layer 1	Layer 2	Layer 3		Layer 1	Layer 2	Layer 3		Layer 1	Layer 2	Layer 3
1	1	8	2	11	1	3	6	21	1	1	9
2	2	6	2	12	1	5	6	22	1	4	9
3	1	7	2	13	1	2	7	23	1	5	9
4	2	1	3	14	1	4	7	24	1	9	9
5	1	8	4	15	1	5	7	25	1	10	9
6	2	8	4	16	1	10	7	26	3	1	10
7	1	4	5	17	1	5	8	27	1	3	10
8	1	5	5	18	1	6	8	28	1	5	10
9	1	7	5	19	1	8	8	29	1	7	10
10	1	9	5	20	1	9	8	30	1	9	10

A quick look at the failed model list suggests that 26 out of the 30 models (86.67%) had only one neuron in Layer 1. The possible reason for this phenomenon is that the condition may have been too stringent for the ANN algorithm to converge because the 20 inputs were forcefully being passed through a single neuron in Layer 1. This reduced the flexibility for assigning weights in the subsequent layers, which increased the difficulty of the model converging in a given number of iteration steps. Efforts such as increasing the maximum limit of iteration steps were made to allow more computation and hence increase the chance of the model training process converging. Increasing the error tolerance for specifying the threshold for the partial derivatives of the error function as the training stopping criteria was also performed to provide a more relaxed condition for the model to converge. Still, these efforts were not very helpful in increasing the number of models that finished the training process successfully. The prediction results were acquired by letting the trained ANN models predict the testing data. A total of 81 data sets were tested, which consist of 37 stuck data and 44 non-stuck data. The model prediction capability was measured in four aspects, which are True Negatives (TNs), False Negatives (FNs), True Positives (TPs) and False Positives (FPs), as stipulated in the confusion matrix previously. Based on these four aspects, benchmarking elements such as the accuracy, sensitivity and specificity can be calculated using Equations (5)–(7), respectively. Due to the vast number of prediction results (~1000 results), it is unfeasible to display all of them in tabular form. Hence, only the top twenty prediction results are displayed in Tables 9–14 for each benchmarking element. Alternatively, a histogram is used to illustrate the distribution of the model's accuracy to give a sense of the overall accuracy among all the trained models.

4.1.1. ANN Model Prediction Results (Logistic Activation Function)

Figure 4 shows the distribution of accuracy among all 1000 ANN models using the logistic activation function. It is observed that around 60% of all the 1000 models had 70–80% prediction accuracy. It is safe to say that around 93% of all the models had at least 70% prediction accuracy.

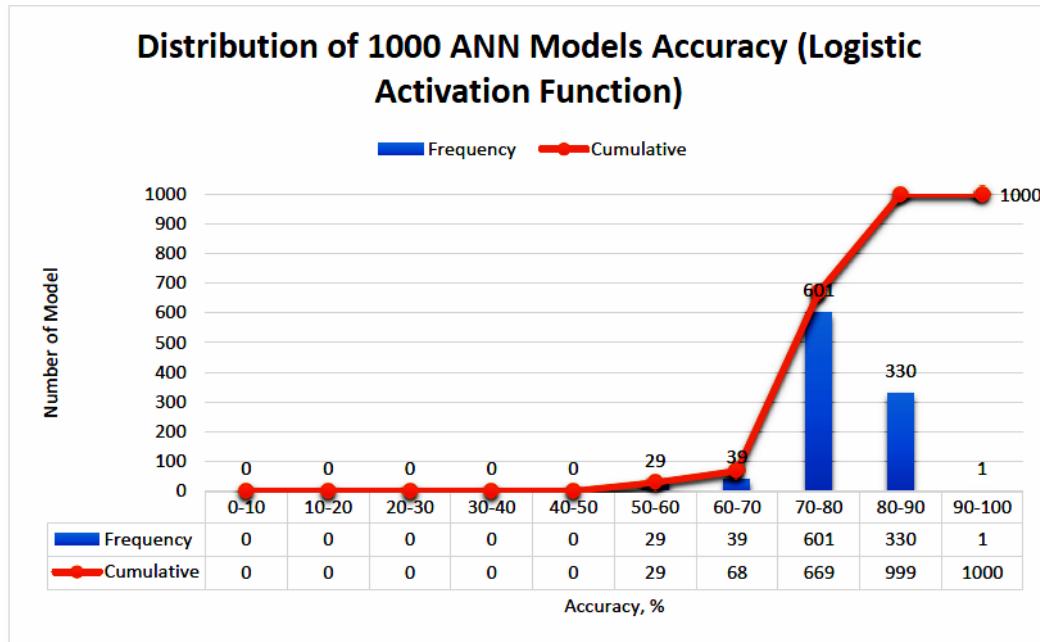


Figure 4. Histogram illustrating the distribution of accuracy among all 1000 ANN models using the logistic activation function.

Tables 9–11 show the top twenty prediction results for the ANN models with the logistic activation function, as sorted according to accuracy, sensitivity and specificity. The criteria of the best model include a good combination of high accuracy, sensitivity and specificity. Priority was given to accuracy because it measures the total prediction accurateness for both stuck and non-stuck cases. The ANN models with high accuracy usually had high sensitivity and specificity, as indicated in Table 9. The highest achievable accuracy among all the ANN models using the logistic activation function was 90.12%. After evaluating the accuracy, sensitivity and specificity were assessed to check for the accurateness in terms of predicting stuck cases and non-stuck cases independently. High sensitivity indicates good results in predicting stuck cases, whereas high specificity indicates good results in predicting non-stuck cases.

Table 9. Top twenty “Accuracy” results for logistic activation function ANN models.

No.	Model Number	Layer 1	Layer 2	Layer 3	Correctly Predicted Case	Incorrectly Predicted Case	TN	FN	TP	FP	Accuracy, %	Sensitivity, %	Specificity, %
1	80	10	8	1	73	8	40	4	33	4	90.12	89.19	90.91
2	580	10	8	6	72	9	38	3	34	6	88.89	91.89	86.36
3	168	8	7	2	71	10	39	5	32	5	87.65	86.49	88.64
4	449	9	5	5	71	10	39	5	32	5	87.65	86.49	88.64
5	598	8	10	6	71	10	38	4	33	6	87.65	89.19	86.36
6	160	10	6	2	70	11	40	7	30	4	86.42	81.08	90.91
7	234	4	4	3	70	11	39	6	31	5	86.42	83.78	88.64
8	300	10	10	3	70	11	38	5	32	6	86.42	86.49	86.36
9	328	8	3	4	70	11	39	6	31	5	86.42	83.78	88.64
10	369	9	7	4	70	11	40	7	30	4	86.42	81.08	90.91
11	490	10	9	5	70	11	40	7	30	4	86.42	81.08	90.91
12	515	5	2	6	70	11	40	7	30	4	86.42	81.08	90.91
13	532	2	4	6	70	11	40	7	30	4	86.42	81.08	90.91
14	585	5	9	6	70	11	41	8	29	3	86.42	78.38	93.18
15	647	7	5	7	70	11	38	5	32	6	86.42	86.49	86.36
16	707	7	1	8	70	11	41	8	29	3	86.42	78.38	93.18
17	730	10	3	8	70	11	38	5	32	6	86.42	86.49	86.36
18	820	10	2	9	70	11	36	3	34	8	86.42	91.89	81.82
19	959	9	6	10	70	11	39	6	31	5	86.42	83.78	88.64
20	47	7	5	1	69	12	40	8	29	4	85.19	78.38	90.91

Table 10. Top twenty “Sensitivity” results for logistic activation function ANN models.

No	Model Number	Layer 1	Layer 2	Layer 3	Correctly Predicted Case	Incorrectly Predicted Case	TN	FN	TP	FP	Accuracy, %	Sensitivity, %	Specificity, %
1	580	10	8	6	72	9	38	3	34	6	88.89	91.89	86.36
2	820	10	2	9	70	11	36	3	34	8	86.42	91.89	81.82
3	80	10	8	1	73	8	40	4	33	4	90.12	89.19	90.91
4	598	8	10	6	71	10	38	4	33	6	87.65	89.19	86.36
5	980	10	8	10	69	12	36	4	33	8	85.19	89.19	81.82
6	960	10	6	10	64	17	31	4	33	13	79.01	89.19	70.45
7	90	10	9	1	62	19	29	4	33	15	76.54	89.19	65.91
8	168	8	7	2	71	10	39	5	32	5	87.65	86.49	88.64
9	449	9	5	5	71	10	39	5	32	5	87.65	86.49	88.64
10	300	10	10	3	70	11	38	5	32	6	86.42	86.49	86.36
11	647	7	5	7	70	11	38	5	32	6	86.42	86.49	86.36
12	730	10	3	8	70	11	38	5	32	6	86.42	86.49	86.36
13	320	10	2	4	67	14	35	5	32	9	82.72	86.49	79.55
14	550	10	5	6	66	15	34	5	32	10	81.48	86.49	77.27
15	940	10	4	10	65	16	33	5	32	11	80.25	86.49	75
16	5	5	1	1	63	18	31	5	32	13	77.78	86.49	70.45
17	398	8	10	4	63	18	31	5	32	13	77.78	86.49	70.45
18	67	7	7	1	62	19	30	5	32	14	76.54	86.49	68.18
19	511	1	2	6	61	20	29	5	32	15	75.31	86.49	65.91
20	234	4	4	3	70	11	39	6	31	5	86.42	83.78	88.64

Table 11. Top twenty “Specificity” results for logistic activation function ANN models.

No.	Model Number	Layer 1	Layer 2	Layer 3	Correctly Predicted Case	Incorrectly Predicted Case	TN	FN	TP	FP	Accuracy	Sensitivity	Specificity
1	2	2	1	1	44	37	44	37	0	0	54.32	0	100
2	3	3	1	1	44	37	44	37	0	0	54.32	0	100
3	10	10	1	1	44	37	44	37	0	0	54.32	0	100
4	19	9	2	1	44	37	44	37	0	0	54.32	0	100
5	21	1	3	1	44	37	44	37	0	0	54.32	0	100
6	27	7	3	1	44	37	44	37	0	0	54.32	0	100
7	81	1	9	1	44	37	44	37	0	0	54.32	0	100
8	84	4	9	1	44	37	44	37	0	0	54.32	0	100
9	94	4	10	1	44	37	44	37	0	0	54.32	0	100
10	101	1	1	2	44	37	44	37	0	0	54.32	0	100
11	110	10	1	2	44	37	44	37	0	0	54.32	0	100
12	115	5	2	2	44	37	44	37	0	0	54.32	0	100
13	121	1	3	2	44	37	44	37	0	0	54.32	0	100
14	132	2	4	2	44	37	44	37	0	0	54.32	0	100
15	161	1	7	2	44	37	44	37	0	0	54.32	0	100
16	182	2	9	2	44	37	44	37	0	0	54.32	0	100
17	231	1	4	3	44	37	44	37	0	0	54.32	0	100
18	241	1	5	3	44	37	44	37	0	0	54.32	0	100
19	253	3	6	3	44	37	44	37	0	0	54.32	0	100
20	311	1	2	4	44	37	44	37	0	0	54.32	0	100

Table 10 illustrates that high sensitivity also corresponded to fairly high accuracy, generally ranging between 75% to 90%. However, the same did not apply to high specificity, as shown in Table 11. All the top twenty highest sensitivity models predicted non-stuck cases with 100% accurateness but wrongly predicted all the stuck cases. In other words, these models predicted all the testing data as non-stuck cases, which is far from the truth. These models are highly skewed and totally unusable. A similar trend can be observed where most of the problematic models contained only one neuron in at least one of the layers. Apart from causing failure to converge in the model training process, models with this kind of network structure also generally have low prediction accuracy. In conclusion, for twenty input parameters, using one neuron in any layer was proven to be too extreme for the ANN algorithm to actually produce a viable trained model with high prediction capability.

Judging from the accuracy, sensitivity and specificity, model no. 80 and model no. 580 seem to be good candidates for being the best model. Model no. 80 has a network topology of [19, 10, 8, 1, 1]. The accuracy, sensitivity and specificity were 90.12%, 89.19% and 90.91%, respectively. The model had the highest accuracy among all the models. Out of the 81 testing data, model no. 80 managed to predict 73 data correctly. On the other hand, model no. 580 has a network topology of [19, 10, 8, 6, 1]. The accuracy, sensitivity and specificity were 88.89%, 91.89% and 86.36%, respectively. The model had the highest sensitivity among all the models. Out of the 81 testing data, model no. 580 managed to predict 72 data correctly, only short by one compared to model no. 80. Model no. 580 is the best model

of all the ANN models using the logistic activation function, despite its overall accuracy being 1.23% (one case) lower than that of model no. 80. Model no. 580 may have had one more misclassified datum, but it managed to predict stuck pipe cases with a sensitivity 2.7% (one case) higher than that of model no. 80. In other words, model no. 580 could capture stuck pipe events better, although false alarms, where non-stuck cases were predicted as stuck cases, were also higher. Figure 5 shows the network topology of model no. 580.

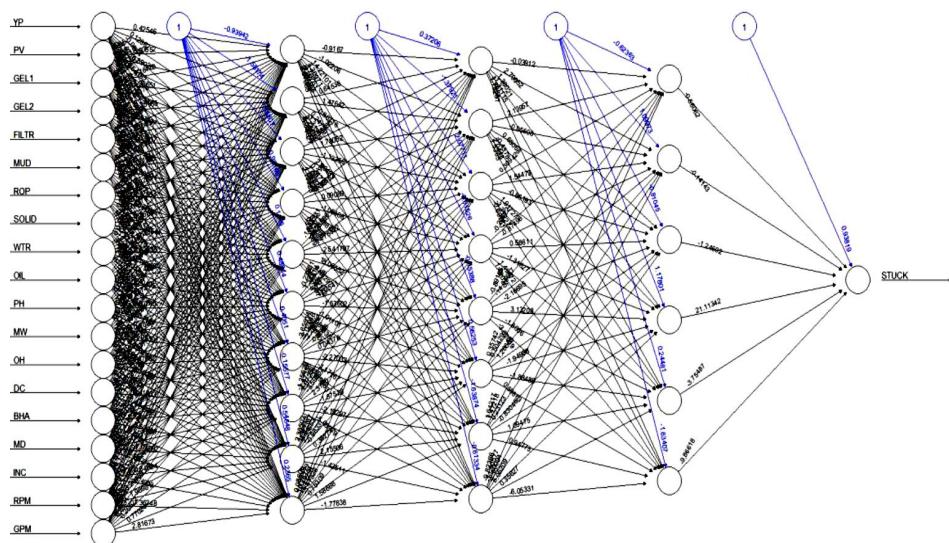


Figure 5. Topology [19, 10, 8, 6, 1] of the best ANN model using the logistic activation function (model no. 580).

4.1.2. ANN Model Prediction Results (Hyperbolic Tangent Activation Function)

Figure 6 shows the distribution of accuracy among all the ANN models using the hyperbolic tangent activation function. It is observed that around 55% of all the 1000 models had 70–80% prediction accuracy. It is safe to say that around 73% of all the models had at least 70% prediction accuracy.

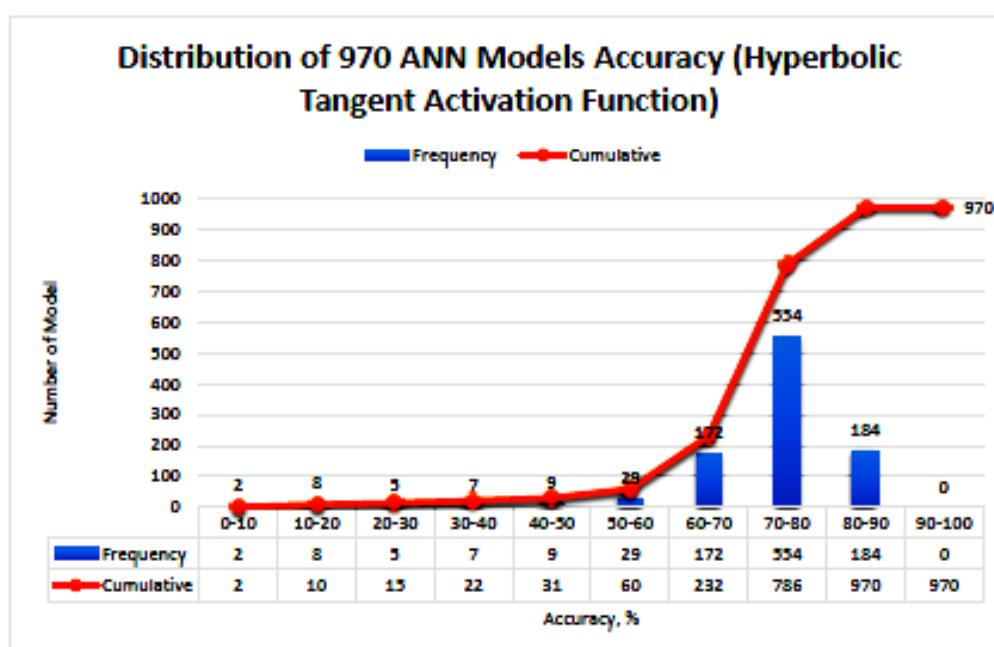


Figure 6. Histogram illustrating the distribution of accuracy among all 1000 ANN models using the hyperbolic tangent activation function.

Tables 12–14 presents the first 20 prediction results for the ANN models with the hyperbolic tangent activation function, sorted by accuracy, sensitivity and specificity. Judging by the accuracy, sensitivity and specificity as shown in Tables 12–14, it can be concluded that model no. 227 and model no. 296, which had the highest accuracy, had a fair balance between sensitivity and specificity. Model no. 227 has a topology of [19, 10, 3, 3, 1] while model no. 296 has a topology of [19, 9, 10, 3, 1]. Both models had an accuracy of 88.89%, sensitivity of 88.89% and specificity of 90.91%. They both predicted an equal number of stuck and non-stuck data. Both models successfully predicted 72 of the 81 test data. Since they had the same accuracy, the best model was evaluated according to the complexity of the model. The simpler network topology is preferred because the required computational power is less. Therefore, model no. 227, which has the network structure [19, 10, 3, 3, 1], is better than the network structure [19, 9, 10, 3, 1] of model no. 296 because the previous one has fewer neurons overall.

Table 12. Top twenty “Accuracy” results for the hyperbolic tangent activation function ANN models.

No.	Model Number	Layer 1	Layer 2	Layer 3	Correctly Predicted Case	Incorrectly Predicted Case	TN	FN	TP	FP	Accuracy, %	Sensitivity, %	Specificity, %
1	227	10	3	3	72	9	40	4	32	4	88.89	88.89	90.91
2	296	9	10	3	72	9	40	4	32	4	88.89	88.89	90.91
3	203	6	1	3	71	10	39	4	32	4	87.65	88.89	90.70
4	284	7	9	3	71	10	37	3	34	7	87.65	91.89	84.09
5	761	9	8	8	71	10	38	3	33	6	87.65	91.67	86.36
6	17	7	2	1	70	11	40	7	30	4	86.42	81.08	90.91
7	143	3	5	2	70	11	39	6	31	5	86.42	83.78	88.64
8	183	5	9	2	70	11	37	4	33	7	86.42	89.19	84.09
9	243	6	5	3	70	11	39	3	31	5	86.42	91.18	88.64
10	336	9	4	4	70	11	40	6	30	4	86.42	83.33	90.91
11	635	9	5	7	70	11	40	5	30	4	86.42	85.71	90.91
12	918	5	5	10	70	11	38	5	32	5	86.42	86.49	88.37
13	8	8	1	1	69	12	38	6	31	6	85.19	83.78	86.36
14	36	6	4	1	69	12	38	6	31	5	85.19	83.78	88.37
15	69	9	7	1	69	12	39	6	30	5	85.19	83.33	88.64
16	117	7	2	2	69	12	37	5	32	7	85.19	86.49	84.09
17	168	10	7	2	69	12	37	5	32	6	85.19	86.49	86.05
18	323	6	3	4	69	12	37	5	32	6	85.19	86.49	86.05
19	325	8	3	4	69	12	38	5	31	4	85.19	86.11	90.48
20	342	5	5	4	69	12	38	5	31	5	85.19	86.11	88.37

Table 13. Top twenty “Sensitivity” results for the hyperbolic tangent activation function ANN models.

No.	Model Number	Layer 1	Layer 2	Layer 3	Correctly Predicted Case	Incorrectly Predicted Case	TN	FN	TP	FP	Accuracy, %	Sensitivity, %	Specificity, %
1	696	1	2	8	14	67	2	0	12	2	17.28	100.00	50.00
2	387	2	10	4	15	66	5	0	10	6	18.52	100.00	45.45
3	500	9	1	6	37	44	0	0	37	44	45.68	100.00	0.00
4	667	1	9	7	37	44	0	0	37	44	45.68	100.00	0.00
5	35	5	4	1	21	60	0	0	21	4	25.93	100.00	0.00
6	1	1	1	1	18	63	0	0	18	6	22.22	100.00	0.00
7	877	1	1	10	15	66	0	0	15	1	18.52	100.00	0.00
8	81	1	9	1	14	67	0	0	14	5	17.28	100.00	0.00
9	82	2	9	1	12	69	0	0	12	2	14.81	100.00	0.00
10	686	1	1	8	12	69	0	0	12	6	14.81	100.00	0.00
11	63	3	7	1	11	70	0	0	11	1	13.58	100.00	0.00
12	21	1	3	1	6	75	0	0	6	1	7.41	100.00	0.00
13	101	1	1	2	5	76	0	0	5	1	6.17	100.00	0.00
14	376	1	9	4	16	65	0	1	16	4	19.75	94.12	0.00
15	284	7	9	3	71	10	37	3	34	7	87.65	91.89	84.09
16	453	10	6	5	66	15	32	3	34	11	81.48	91.89	74.42
17	761	9	8	8	71	10	38	3	33	6	87.65	91.67	86.36
18	524	4	4	6	64	17	31	3	33	10	79.01	91.67	75.61
19	503	2	2	6	49	32	27	2	22	8	60.49	91.67	77.14
20	243	6	5	3	70	11	39	3	31	5	86.42	91.18	88.64

Figure 7 shows the network topology of model no. 227, the best model among all the hyperbolic tangent ANN models. Table 15 summarizes the topology, accuracy, sensitivity and specificity of the two best ANN models using the logistic activation function and the hyperbolic tangent activation function. In comparison, the two best models of two different activation functions had the same accuracy, which was 88.89%. The best model of the logistic activation function had a higher sensitivity

than the best model of the hyperbolic tangent activation function (91.89% versus 88.89%). However, the best hyperbolic tangent activation function model had higher specificity than the best logistic activation function model (90.91% versus 86.36%).

Table 14. Top twenty “Specificity” results for the hyperbolic tangent activation function ANN models.

No.	Model Number	Layer 1	Layer 2	Layer 3	Correctly Predicted Case	Incorrectly Predicted Case	TN	FN	TP	FP	Accuracy, %	Sensitivity, %	Specificity, %
1	829	1	6	9	44	37	44	37	0	0	54.32	0.00	100.00
2	199	1	1	3	30	51	30	5	0	0	37.04	0.00	100.00
3	41	1	5	1	26	55	26	8	0	0	32.10	0.00	100.00
4	217	10	2	3	66	15	42	10	24	1	81.48	70.59	97.67
5	902	8	3	10	62	19	39	11	23	1	76.54	67.65	97.50
6	228	1	4	3	43	38	31	11	12	1	53.09	52.17	96.88
7	225	8	3	3	50	31	42	27	8	2	61.73	22.86	95.45
8	885	10	1	10	69	12	41	7	28	2	85.19	80.00	95.35
9	324	7	3	4	64	17	41	14	23	2	79.01	62.16	95.35
10	597	8	1	7	60	21	39	13	21	2	74.07	61.76	95.12
11	112	2	2	2	42	39	30	7	12	2	51.85	63.16	93.75
12	814	5	4	9	66	15	41	12	25	3	81.48	67.57	93.18
13	472	10	8	5	69	12	40	7	29	3	85.19	80.56	93.02
14	353	6	6	4	66	15	40	10	26	3	81.48	72.22	93.02
15	399	4	1	5	64	17	40	13	24	3	79.01	64.86	93.02
16	913	9	4	10	69	12	39	7	30	3	85.19	81.08	92.86
17	485	4	10	5	68	13	39	8	29	3	83.95	78.38	92.86
18	912	8	4	10	67	14	39	8	28	3	82.72	77.78	92.86
19	401	6	1	5	64	17	39	12	25	3	79.01	67.57	92.86
20	676	10	9	7	66	15	38	7	28	3	81.48	80.00	92.68

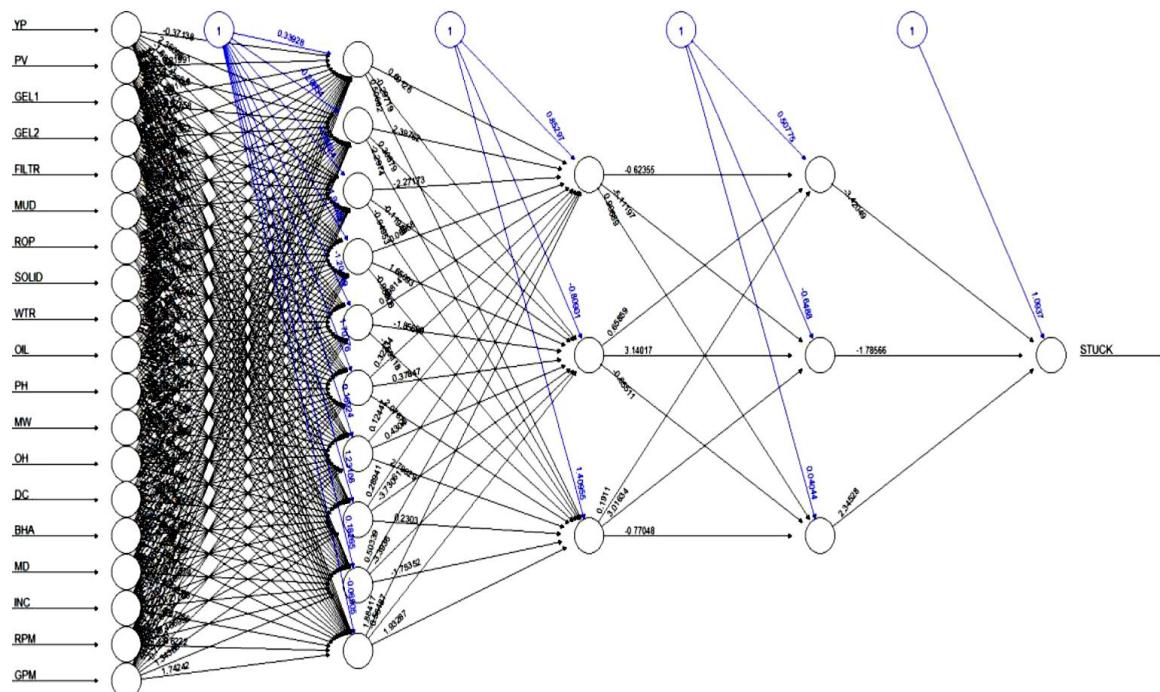


Figure 7. Topology [19, 10, 3, 3, 1] of the best ANN model using the hyperbolic tangent function (model no. 227).

Table 15. Summary of best ANN models of logistic and hyperbolic tangent activation functions.

Activation Function	Model Number	Layer 1	Layer 2	Layer 3	Accuracy, %	Sensitivity, %	Specificity, %	Best ANN Model
Logistic	580	10	8	6	88.89	91.89	86.36	✓
Hyperbolic Tangent	227	10	3	3	88.89	88.89	90.91	✗

ANN models are effective in handling data with noise, mimic biological neural networks, perform better than statistical/regression-based models and accurately predict and model highly non-linear and complex biological processes [30]. The present study shows that the ANN model can give better accuracy with the logistic activation function; therefore, it should be preferred over the hyperbolic tangent activation function because it has higher sensitivity. This will allow the better detection of stuck pipe events, although some non-stuck cases may be misclassified as stuck pipe cases. It is always best to be overly sensitive in detecting stuck pipes to ensure a minimum number of uncaught stuck pipe events.

4.2. SVM Model Prediction Results

Three different kernel functions were used to perform sensitivity analysis on the SVM model. By modifying the regularization factor (C) and hyper-parameters such as the polynomial degree (D) and the sigma (σ), a total of 66 SVM models were constructed. These SVM models were tested against the test data to verify their prediction accuracy.

4.2.1. SVM Model Prediction Results (Linear Kernel Function)

The same methodology as for the ANN model was applied to produce the prediction results for the SVM model. The model's prediction capacity was measured in four aspects, which were True Negatives (TNs), False Negatives (FNs), True Positives (TPs) and False Positives (FPs). Based on these four aspects, the accuracy, sensitivity and specificity could be calculated. Table 16 summarizes the prediction results for the SVM model using the linear kernel function. A total of six cases of sensitivity were developed by changing the regularization factor (C) within a range of 0.1 to 5000. Based on the results, in the case of the linear kernel function, the SVM models with regularization factors (C) of 10, 100 and 1000 achieved the highest prediction accuracy of 75.31%. These three SVM models also had the same sensitivity and specificity, which were 24.69% and 67.57%. All the models successfully predicted 25 of the 37 stuck data and 36 of the 44 non-stuck data. Since the three models had the same accuracy, sensitivity and specificity, the model which uses the lower regularization factor (C) is chosen as the best model because it requires less computing power and time. Therefore, the best SVM model using the linear kernel function is model no. 3, with a regularization factor (C) of 10.

Table 16. Prediction results for the SVM model using the linear kernel function.

Model Number	C	Correctly Predicted Case	Incorrectly Predicted Case	TN	FN	TP	FP	Accuracy, %	Sensitivity, %	Specificity, %
1	0.1	57	24	33	13	24	11	70.37	64.86	75.00
2	1	60	21	36	13	24	8	74.07	64.86	81.82
3	10	61	20	36	12	25	8	75.31	67.57	81.82
4	100	61	20	36	12	25	8	75.31	67.57	81.82
5	1000	61	20	36	12	25	8	75.31	67.57	81.82
6	5000	60	21	35	12	25	9	74.07	67.57	79.55

4.2.2. SVM Model Prediction Results (RBF Kernel Function)

Table 17 summarizes the prediction results for the SVM model using the RBF kernel function. A total of 30 cases of sensitivity were developed by changing the regularization factor (C) within a range of 0.1 to 5000 and sigma (σ) within a range of 0.1 to 1. Based on the tabulated results, in the case of the RBF kernel function, the highest achievable accuracy was 83.95%, which was attained collectively by nine models (model nos. 7, 11, 12, 16, 17, 21, 22, 26 and 27). Within these nine models, the models highlighted in blue (model nos. 11, 16, 21 and 26) had a sensitivity of 86.49% and specificity of 81.82%. They managed to correctly predict 32 of the 37 stuck data and 36 of the 44 non-stuck data. The models highlighted in green (model nos. 7, 12, 17, 22 and 27) had a sensitivity of 83.78% and specificity of 84.09%. They managed to correctly predict 31 of the 37 stuck data and 37 of the 44 non-stuck data. Judging by the overall results, the SVM model with sigma (σ) 0.1 and 0.25 generally had good prediction accuracy, around 82% to 83%. The only exception was when sigma (σ) 0.1 and

0.25 were associated with a regularization factor (C) of 0.1. This combination gave comparatively low prediction accuracies of 56% and 69%, respectively. In choosing the best model, preference is given to higher sensitivity instead of higher specificity in cases where the accuracy of the models is the same. Therefore, models highlighted in blue with higher sensitivity are desired. A lower regularization factor (C) is also preferred to reduce the computational time. Taking all these considerations into account, the best SVM model using the RBF kernel function is model no. 11, with a regularization factor (C) of 10 and sigma (σ) of 0.1.

Table 17. Prediction results for the SVM model using the RBF kernel function.

Model Number	C	σ	Correctly Predicted Case	Incorrectly Predicted Case	TN	FN	TP	FP	Accuracy, %	Sensitivity, %	Specificity, %
1	0.1	0.1	56	25	24	5	32	20	69.14	86.49	54.55
2	0.1	0.25	46	35	44	35	2	0	56.79	5.41	100.00
3	0.1	0.5	44	37	44	37	0	0	54.32	0.00	100.00
4	0.1	0.75	44	37	44	37	0	0	54.32	0.00	100.00
5	0.1	1	44	37	44	37	0	0	54.32	0.00	100.00
6	1	0.1	67	14	37	7	30	7	82.72	81.08	84.09
7	1	0.25	68	13	37	6	31	7	83.95	83.78	84.09
8	1	0.5	65	16	43	15	22	1	80.25	59.46	97.73
9	1	0.75	64	17	44	17	20	0	79.01	54.05	100.00
10	1	1	63	18	44	18	19	0	77.78	51.35	100.00
11	10	0.1	68	13	36	5	32	8	83.95	86.49	81.82
12	10	0.25	68	13	37	6	31	7	83.95	83.78	84.09
13	10	0.5	64	17	42	15	22	2	79.01	59.46	95.45
14	10	0.75	65	16	44	16	21	0	80.25	56.76	100.00
15	10	1	63	18	44	18	19	0	77.78	51.35	100.00
16	100	0.1	68	13	36	5	32	8	83.95	86.49	81.82
17	100	0.25	68	13	37	6	31	7	83.95	83.78	84.09
18	100	0.5	64	17	42	15	22	2	79.01	59.46	95.45
19	100	0.75	65	16	44	16	21	0	80.25	56.76	100.00
20	100	1	63	18	44	18	19	0	77.78	51.35	100.00
21	1000	0.1	68	13	36	5	32	8	83.95	86.49	81.82
22	1000	0.25	68	13	37	6	31	7	83.95	83.78	84.09
23	1000	0.5	64	17	42	15	22	2	79.01	59.46	95.45
24	1000	0.75	65	16	44	16	21	0	80.25	56.76	100.00
25	1000	1	63	18	44	18	19	0	77.78	51.35	100.00
26	5000	0.1	68	13	36	5	32	8	83.95	86.49	81.82
27	5000	0.25	68	13	37	6	31	7	83.95	83.78	84.09
28	5000	0.5	64	17	42	15	22	2	79.01	59.46	95.45
29	5000	0.75	65	16	44	16	21	0	80.25	56.76	100.00
30	5000	1	63	18	44	18	19	0	77.78	51.35	100.00

4.2.3. SVM Model Prediction Results (Polynomial Kernel Function)

Table 18 summarizes the prediction results for the SVM model using the polynomial kernel function. A total of 30 cases of sensitivity were developed by changing the regularization factor (C) within a range of 0.1 to 5000 and degree (D) within a range of 1 to 5. Based on the tabulated results, in the case of the polynomial kernel function, the highest achievable accuracy was 81.48%, which was attained collectively by six models (model nos. 5, 10, 15, 20, 25 and 30). These six models had a sensitivity of 67.57% and specificity of 93.18%. They managed to correctly predict 25 of the 37 stuck data and 41 of the 44 non-stuck data. Judging by the overall results, all SVM models with a degree (D) of 5 produced the best prediction accuracy, at 81.48%. However, the prediction capacity for stuck data was rather weak, at only 67.57%. In choosing the best model, preference is given to a lower regularization factor (C) to reduce the computational time. Therefore, the best SVM model using the polynomial kernel function is model no. 5, with a regularization factor (C) of 0.1 and degree (D) of 5. Table 18 shows the best SVM models using the functions of the linear kernel, the RBF kernel and the polynomial kernel.

Upon comparison, it is an opinion that the best SVM model is that with the RBF kernel function because it had the highest accuracy as well as sufficiently high and fairly balanced sensitivity and specificity. In addition, the above results show that the SVM polynomial model was very good at predicting non-stuck data but it worked poorly in predicting stuck data.

Table 18. Summary of the best SVM models of the linear kernel function, RBF kernel function and polynomial kernel function.

Kernel Function	Model Number	Accuracy, %	Sensitivity, %	Specificity, %	Best SVM Model
Linear	3	75.31	67.57	81.82	✗
RBF	11	83.95	86.49	81.82	✓
Polynomial	5	81.48	67.57	93.18	✗

4.3. Comparison of Prediction Results for ANN and SVM Best Models

The best ANN and SVM models were selected to be compared to determine the best machine learning model for predicting stuck pipes. Table 19 summarizes the details of both the best ANN and SVM models and their respective prediction results. Upon comparison, it is concluded that the ANN model can perform better in predicting stuck pipes. The accuracy, sensitivity and specificity of the best ANN model are consistently higher than those of the best SVM model. The best ANN model has a higher number of correctly predicted cases and a lower number of incorrectly predicted cases than the SVM model. In conclusion, with regard to this research, the best machine learning model to apply in building a stuck pipe prediction model is the Artificial Neural Network (ANN). The best achievable stuck pipe prediction in terms of accuracy for this model in this study was as high as 88.89%.

Table 19. Comparison between the best models from the ANN and SVM.

Machine Learning Model	Description of Best Model	Correctly Predicted Case	Incorrectly Predicted Case	Accuracy, %	Sensitivity, %	Specificity, %	Best Model
Artificial Neural Network (ANN)	Network topology [19, 10, 8, 6, 1]	72	9	88.89	91.89	86.36	✓
	Activation function Logistic						
Support Vector Machine (SVM)	Kernel Function RBF C σ 10 0.1	68	13	83.95	86.49	81.82	✗

Based on the prediction results for stuck pipes from both the ANN and SVM models, it can be determined which machine learning methodology is able to achieve better prediction accuracy with respect to this study. In addition, the results from this study can give an overall perception of the feasibility of applying machine learning to predict stuck pipe events in drilling operations. The study also serves to spark the interest of researchers in the subject of machine learning and also arouses the thought of incorporating machine learning into other petroleum service operations.

5. Conclusions

In this study, two machine learning methodologies—namely, the artificial neural network (ANN) and the support vector machine (SVM)—were used to construct a prediction model for stuck pipes. The data used for training and testing the machine learning models were extracted from a database of well drilling operations. The data extraction process was performed through the use of the Data Analyzer software. Then, the raw data collected were subjected to cleaning and quality checking. Nineteen drilling parameters related to stuck pipes were identified as the inputs for model training. A total of 268 data sets were collected, of which 123 data sets were stuck pipe data and 145 data sets were non-stuck data. Of the data collected, 70% were designated as training data while the remaining 30% were classified as test data. The R programming software was used to build, train and test the machine learning models. In the ANN model, cases of sensitivity were created by varying the network structure and the activation function. Two activation functions—namely, the logistic activation function and the hyperbolic tangent activation function—were tested in this study. In addition, all the possible combinations of network structure, from [19, 1, 1, 1, 1] to [19, 10, 10, 10, 1], were tested for each activation function. A total of 1970 cases of sensitivity were created for the ANN model. In the SVM model, cases of sensitivity were created by changing the kernel function and the hyper-parameters.

Three kernel functions—namely, linear, RBF and polynomial—were tested in this study. In addition, the regularization factor (C) and hyper-parameters such as the polynomial degree (D) and sigma (σ) were also used to create cases of sensitivity for the SVM model.

Based on the prediction results for the stuck pipes for all the ANN models, the best model is identified as being the ANN model that uses a logistic activation function with the network structure [19, 10, 8, 6, 1]. The best ANN model achieved an accuracy of 88.89%, a sensitivity of 91.89% and a specificity of 86.36%. For the SVM model, the best model is identified as the SVM model that uses the RBF kernel function with a value of 10 for the regularization factor (C) and a value of 0.1 for sigma (σ). The best SVM model achieved an accuracy of 83.95%, a sensitivity of 86.49% and a specificity of 81.82%. By comparing both of the best models from the ANN and SVM, our conclusion is that the best model of the ANN has better stuck pipe prediction capability than the best model of the SVM, as indicated by its accuracy, sensitivity and specificity. Through the process of analyzing the results of this study, an overall perception of the feasibility of applying machine learning to the prediction of stuck pipes can be created. The percentage accuracy of the prediction from the machine learning model is encouraging, suggesting that the prediction of stuck pipes using machine learning is indeed practical. The best adjustable parameters were determined to tune the machine learning model to achieve a higher stuck pipe prediction accuracy. The potential of the machine learning model was also studied in depth by performing a sensitivity analysis on the aforementioned machine learning model. Several adjustable parameters were identified to be used to build sensitivity cases. In the end, the model could be fine-tuned based on the results of the sensitivity analysis, in order to obtain better results in predicting stuck pipes during drilling operation.

6. Recommendations

Based on real-time operational drilling data as inputs to a presented machine learning approach, the risks of stuck pipes can be predicted by well service engineers, and warnings can also be provided when a high risk of stuck pipe scenarios is encountered. It is recommended that more data sets be included so that the model training can be exposed to different patterns of stuck pipe events. This could help to generalize the model for predicting stuck pipe events and prevent the over-fitting of data during the training process. Ultimately, this could improve the predictive power of the machine learning model. In addition, with sufficient quality data, the machine learning model could be trained to predict differentially stuck, mechanically stuck and non-stuck cases. Unlike the current machine learning model in this study, which can only predict cases as stuck or non-stuck, the ability to predict the type of stuck pipe will certainly help in decision making for carrying out counter measures to avoid stuck pipes. Apart from that, it is also recommended to include more drilling parameters as inputs for model training. Due to the limitation of the completeness of the data, certain drilling parameters such as the TVD and differential pressure were dropped from being used as training inputs. The inclusion of additional drilling parameters related to stuck pipes will certainly help to build a better machine learning model.

In future research, we would recommend work on a feature selection-based algorithm to justify and present which of the nineteen drilling parameters decisively influence the stuck pipes. Moreover, another recommendation involves more machine learning methodology to provide more in-depth comparisons, such as discriminant analysis and fuzzy logic, which are not covered in this research. This could provide a broader view of the feasibility and capabilities of machine learning to predict stuck pipes. In addition, in the case of the ANN and SVM, more activation functions, network structures, kernel functions and hyper-parameters could be used to perform the sensitivity analysis. A deeper understanding could be gained of the effect of these parameters on the prediction accuracy of the model. Last but not least, machine learning may seem too complex and far from being accessible to people who have no background of data science. Therefore, an easy-to-use graphical user interface (GUI) could be placed on top of the complex machine learning coding to promote the use of machine learning applications. With a GUI, the majority of users would be able to use the machine learning

application with a reduced learning curve because they would not be required to deal directly with the backend coding. For example, loading input data into the machine learning model might be done easily through lines of programming code for a data scientist but not for any ordinary users such as engineers. An interactive GUI could be used instead to allow ordinary users to load the input data on the software interface. This would significantly reduce the entry barrier for ordinary users to apply machine learning.

Author Contributions: J.A.K. performed the project administration, funding acquisition, supervision, literature review, results analysis, paper drafting and conceptualization. M.I. performed the literature review, software development, funding acquisition, article review and drafting. S.I. contributed to the article writing, structuring and review of results, resources and visualization. F.K.Y. performed visualization and analysis. M.S.A.R. performed software analysis. A.R.S. performed the data management. A.G. performed a review of the article and resource management. N.Z. contributed to the tuning of the algorithms. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Education, KSA and Deanship of Scientific Research, Najran University, Saudi Arabia, with the award of the research fund NU/ESCI/17/057.

Acknowledgments: The authors would like to express their gratitude to the Ministry of Education, KSA and the Deanship of Scientific Research, Najran University, Kingdom of Saudi Arabia for their financial and technical support under code number (NU/ESCI/17/057). We also acknowledge Universiti Teknologi PETRONAS, Malaysia, Nazarbayev University, Nur-Sultan City, Kazakhstan and University of Management & Technology, Lahore, Pakistan for facilitating the algorithm and publication.

Conflicts of Interest: The authors have no conflict of interest.

References

1. Aljubran, M.; Al-Yami, A.; Madan, M. Best Practices to Prevent Stuck Pipe Incidents in Offshore Shaly Formations. In Proceedings of the SPE Middle East Oil & Gas Show and Conference, Manama, Bahrain, 6–9 March 2017. SPE-183717-MS. [[CrossRef](#)]
2. Howard, J.; Glover, S. Tracking stuck pipe probability while drilling. In Proceedings of the SPE/IADC Drilling Conference, Dallas, TX, USA, 15–18 February 1994. SPE-27528-MS. [[CrossRef](#)]
3. Hayward, J. Cause and Cure of Frozen Drill Pipe and Casing. In *Drilling and Production Practice*; American Petroleum Institute: Washington, DC, USA, 1937.
4. Javeri, S.M.; Haindade, Z.M.W.; Jere, C.B. Mitigating loss circulation and differential sticking problems using silicon nanoparticles. In Proceedings of the SPE/IADC Middle East Drilling Technology Conference and Exhibition, Muscat, Oman, 24–26 October 2011. SPE-171029-MS. [[CrossRef](#)]
5. Spoerker, H.F.; Maidla, E.E.; Thonhauser, G. Rigorous Identification of Unplanned and Invisible Lost Time for Value Added Propositions Aimed at Performance Enhancement. In Proceedings of the SPE/IADC Drilling Conference and Exhibition, Amsterdam, the Netherlands, 1–3 March 2011. SPE-138922-MS. [[CrossRef](#)]
6. Muqeem, M.A.; Weekse, A.E.; Al-Hajji, A.A. Stuck Pipe Best Practices-A Challenging Approach to Reducing Stuck Pipe Costs. In Proceedings of the SPE Saudi Arabia Section Technical Symposium and Exhibition, Al-Khobar, Saudi Arabia, 8–11 April 2012. SPE-97694-MS. [[CrossRef](#)]
7. Yarim, G.; Uchytil, R.; May, R.B.; Trejo, A.; Church, P. Stuck Pipe Prevention—A Proactive Solution to an Old Problem. In Proceedings of the SPE Annual Technical Conference and Exhibition, Anaheim, CA, USA, 11–14 November 2007. SPE-109914-MS. [[CrossRef](#)]
8. Amanullah, M.; Al-Arfaj, M.K. Method and Apparatus to Reduce the Probability of Differential Sticking. In Proceedings of the IADC/SPE Asia Pacific Drilling Technology Conference, Singapore, 22–24 August 2016. SPE-195095-MS. [[CrossRef](#)]
9. Magana-Mora, A.; Gharbi, S.; Alshaikh, A.; Al-Yami, A. AccuPipePred: A framework for the accurate and early detection of stuck pipe for real-time drilling operations. In Proceedings of the SPE Middle East Oil and Gas Show and Conference, Manama, Bahrain, 18–21 March 2019. SPE-194980-MS. [[CrossRef](#)]
10. Fox, J.; Andersen, R. *Using the R statistical Computing Environment to Teach Social Statistics Courses*; Department of Sociology, McMaster University: Hamilton, ON, Canada, 2005; pp. 2–4.
11. Vance, A. Data Analysts Captivated by R's Power. New York Times, 2009. 6(5.4). Available online: <https://kenbenoit.net/data-analysts-captivated-by-r%E2%80%99s-power/> (accessed on 6 January 2009).

12. Chamkalani, A.; Pordel Shahri, M.; Poordad, S. Support vector machine model: A new methodology for stuck pipe prediction. In Proceedings of the SPE Unconventional Gas Conference and Exhibition, Muscat, Oman, 28–30 January 2013. SPE-164003-MS. [[CrossRef](#)]
13. Helmic, W.; Longley, A. Pressure-differential Sticking of Drill Pipe and How it can be Avoided or Relieved. In *Drilling and Production Practice*; American Petroleum Institute: Washington, DC, USA, 1957.
14. Courteille, J.; Zurdo, C. A new approach to differential sticking. In Proceedings of the SPE Annual Technical Conference and Exhibition, Las Vegas, NV, USA, 22–26 September 1985. SPE-14244-MS. [[CrossRef](#)]
15. Khan, J.A.; Pao, W.K. Fill Removal with foam in horizontal well cleaning in coiled tubing. *Res. J. Appl. Sci. Eng. Technol.* **2013**, *6*, 2655–2661. [[CrossRef](#)]
16. Khan, J.A.; Pao, W.K. Horizontal Well Cleanup Operation using Foam in Different Coiled Tubing/Annulus Diameter Ratios. *J. Appl. Sci.* **2013**, *14*, 3235–3241.
17. Khan, J.A.; Pao, W.K. Effect of different qualities of foam on fill particle transport in horizontal well cleanup operation using coiled tubing. *Adv. Mater. Res.* **2014**, *903*, 39–44. [[CrossRef](#)]
18. Pao, W.; Khan, J.A.; Ofei, T.N.; Irawan, S. Fill removal from horizontal wellbore using foam in different coiled tubing/annulus diameter ratios. *Int. J. Oil Gas Coal Technol.* **2015**, *9*, 129–147. [[CrossRef](#)]
19. Khan, J.A.; Irawan, S.; Padmanabhan, E.; Al-Kayiem, H.H.; Rai, S. Optimization of coiled tubing nozzle for sand removal from wellbore. *J. Pet. Explor. Prod. Technol.* **2020**, *10*, 53–66. [[CrossRef](#)]
20. Zhang, F.; Islam, A.; Zeng, H.; Chen, Z.; Zeng, Y.; Wang, X.; Li, S. Real Time Stuck Pipe Prediction by Using a Combination of Physics-Based Model and Data Analytics Approach. In Proceedings of the Abu Dhabi International Petroleum Exhibition & Conference, Abu Dhabi, UAE, 11–14 November 2019. SPE-197167-MS. [[CrossRef](#)]
21. Wang, J.; Tang, S.X.; Krstic, M. Adaptive output-feedback control of torsional vibration in off-shore rotary oil drilling systems. *Automatica* **2020**, *111*, 108640. [[CrossRef](#)]
22. Cai, B.; Liu, Y.; Liu, Z.; Chang, Y.; Jiang, L. Risk analysis of subsea blowout preventer by mapping GO models into Bayesian networks. In *Bayesian Networks for Reliability Engineering*; Springer: Berlin, Germany, 2020; pp. 165–187.
23. Khan, F.; Abimbola, M.; Hashemi, S.J. Blowout Risk in Drilling and Production. In *Encyclopedia of Maritime and Offshore Engineering*; Wiley: Hoboken, NJ, USA, 2017; pp. 1–19.
24. Siruvuri, C.; Nagarakanti, S.; Samuel, R. Stuck pipe prediction and avoidance: A convolutional neural network approach. In Proceedings of the IADC/SPE Drilling Conference, Miami, FL, USA, 21–23 February 2006. SPE-98378-MS. [[CrossRef](#)]
25. Miri, R.; Afshar, S.; Lourenco, A. Development of artificial neural networks to predict differential pipe sticking in iranian offshore oil fields. In Proceedings of the International Oil Conference and Exhibition in Mexico, Veracruz, Mexico, 27–30 June 2007. SPE-108500-MS. [[CrossRef](#)]
26. Han, S.; Qubo, C.; Meng, H. Parameter selection in SVM with RBF kernel function. In Proceedings of the World Automation Congress 2012, Puerto Vallarta, Mexico, 24–28 June 2012.
27. Cortez, P. Data mining with neural networks and support vector machines using the R/rminer tool. In *Industrial Conference on Data Mining*; Springer: Berlin, Germany, 2010.
28. Vapnik, V. *The Nature of Statistical Learning Theory Autor*; Springer: New York, NY, USA, 2000.
29. Jahanbakhshi, R.; Keshavarzi, R.; Aliyari Shoorehdeli, M.; 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 Dril. Complet.* **2012**, *27*, 586–595. [[CrossRef](#)]
30. Khan, S.; Bhakuni, V.; Praveen, V.; Tewari, R.; Tripathi, C.K.M.; Gupta, V.D. Maximizing the native concentration and shelf life of protein: A multiobjective optimization to reduce aggregation. *Appl. Microbiol. Biotechnol.* **2011**, *89*, 99–108. [[CrossRef](#)] [[PubMed](#)]
31. Al-Kayiem, H.H.; Khan, J.A. Evaluation of Alkali/Surfactant/Polymer Flooding on Separation and Stabilization of Water/Oil Emulsion by Statistical Modelling. *Energy Fuels* **2017**, *31*, 9290–9301. [[CrossRef](#)]
32. Khan, J.A.; Al-Kayiem, H.H.; Aleem, W.; Saad, A.B. Influence of alkali-surfactant-polymer flooding on the coalescence and sedimentation of oil/water emulsion in gravity separation. *J. Pet. Sci. Eng.* **2019**, *173*, 640–649. [[CrossRef](#)]

33. Lee, L.C.; Liong, C.Y.; Jemain, A.A. Validity of the best practice in splitting data for hold-out validation strategy as performed on the ink strokes in the context of forensic science. *Microchem. J.* **2018**, *139*, 125–133. [[CrossRef](#)]
34. Jahanbakhshi, R.; Keshavarzi, R. Intelligent classifier approach for prediction and sensitivity analysis of differential pipe sticking: A comparative study. *J. Energy Resour. Technol.* **2016**, *138*, 052904. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).