Implementing Artificial Neural Networks and Support Vector Machines in
Stuck Pipe Prediction

by

Islam Albaiyat, B.S.

A Thesis

In

PETROLEUM ENGINEERING

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for
the Degree of

MASTER OF SCIENCE

IN

PETROLEUM ENGINEERING

Approved

Dr. Lloyd Heinze
Chair of Committee

Dr. Shameem Siddiqui

Dr. Eun Youn

Dr. Mario Beruvides

Peggy Gordon Miller
Dean of the Graduate School

August, 2012

Copyright 2012, Islam Albaiyat

ACKNOWLEDGMENTS

I would like to express my acknowledgment and gratefulness to the staff and faculty of Bob. L. Herd Department of Petroleum Engineering at Texas Tech University for giving me the opportunity to be enrolled in the masters' program in petroleum engineering. Being a graduate student at Texas Tech is a beautiful and unforgettable experience that allowed me to learn many things that would maintain me in the right track of success and excellence. Indeed, I feel proud to be a graduate member in such a high tech school with a high reputation, outstanding performance, and a friendly and healthy environment.

My research work would not have been possible without the help from my committee members starting with the committee chair Dr. Lloyd Heinze who has assisted, supervised, and guided me to complete my thesis requirements within the estimated research time. I also cannot forget the technical assistance and support from Dr. Eun Youn with who I had a complete graduate course in neural networks where he taught and helped me in constructing my machine learning models. My thanks also go to Dr. Shameem Siddiqui who assisted me in editing my thesis defense presentation and supported me in the SPE paper contest with encouragements. I would also highly appreciate the constructive and valuable suggestions that came from Dr. Mario Beruvides who has shared his experience in data analysis and manipulation to achieve more convincing research. In addition, I would like to thank Mrs. Jeannie Bennett who shared her experience in assisting me editing my thesis report.

Finally, I would send my warmest appreciations to my father and mother, and all my family members for their dedication, blessing, and support that give me the strength to achieve my first journey in research and higher education. I also owe my deep gratitude to my wife who supported me during my masters' studies with all her patience.

TABLE OF CONTENTS

ACKNOWLEDGMENTS.....	ii
ABSTRACT	vi
LIST OF TABLES.....	viii
LIST OF FIGURES	ix
I. INTRODUCTION	1
1.1 What is Stuck Pipe?.....	2
1.2 Differential Sticking	3
1.3 Mechanical Sticking	5
1.4 Practices Used in Freeing Stuck Pipe	6
1.5 Statement of the Problem	9
II. LITERATURE REVIEW.....	11
2.1 Stuck Pipe Predictive and Deductive Approaches	11
2.2 Multivariate Statistical Analysis	11
2.3 Machine Learning and Soft Computing	14
2.4 Other Predictive and Detective Methods.....	18
III. MACHINE LEARNING AND SOFT COMPUTING	20
3.1 Background	20
3.2 Applications	23
3.3 Experimental Data.....	24
3.4 Artificial Neural Networks	25
3.4.1 Background	25
3.4.2 Mathematics of ANNs	26
3.4.2.1 Activation functions.....	29
3.4.2.2 Gradient descent	30
3.5 Support Vector Machines	34
3.5.1 Background	34
3.5.2 Mathematics of SVMs	35
3.5.2.1 Linearly separable case	35
3.5.2.2 Linearly non-separable case	41
3.5.2.3 Kernel functions	43
IV. METHODOLOGY	45
4.1 Data Selection and Assembly	45
4.2 Programing Software.....	54
4.3 Artificial Neural Network (ANN) Model Construction	55
4.3.1 Developed Models	55
4.3.2 ANN MATLAB Built-in Models	58
4.4 Support Vector Machines (SVM) Model Construction	59
4.4.1 Developed Models	59
4.4.2 SVM MATLAB Built-in Models	61
4.4.3 SVM MATLAB Interface Models	62

4.5 Accuracy and Error Measures.....	64
4.6 Data Analysis	65
V. RESULTS	67
5.1 Accuracy of the Models.....	67
5.2 Sensitive Study.....	74
5.2.1 Sensitivity of the Independent Variables.....	74
5.2.1.1 Utilizing MATLAB_SVM_ANN.....	75
5.2.1.2 Utilizing Constructed SVM Linear Model.....	80
5.2.2 Sensitivity of the Size of Training Samples	85
5.2.2.1 Group_A Data	85
5.2.2.2 Company_X Data	87
VI. DISCUSSION	95
VII. RECOMMENDATIONS.....	98
7.1 Recommendations for Data Collection	98
7.2 Recommendation for the Model Construction.....	100
VIII. CONCLUSIONS	103
BIBLIOGRAPHY	105
BIOGRAPHY	109
APPENDIX	110

ABSTRACT

Stuck pipe has been recognized as one of the most challenging and costly problems in the oil and gas industry. However, this problem can be treated proactively through predicting it before it occurs.

The purpose of this study is to implement the two most powerful machine learning methods, Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs), to predict stuck pipe occurrences. Two developed models for ANNs and SVMs with different scenarios implemented for prediction purposes. The models were designed and constructed by the MATLAB language. The MATLAB built-in functions of ANNs and SVMs, and the MATLAB interface from the library of support vector machines were applied to compare the results. Furthermore, two databases that include mud properties, directional characteristics, and drilling parameters have been assembled for training and testing processes. The study involved classifying stuck pipe incidents into two groups - stuck and non-stuck - and also into three groups: differentially stuck, mechanically stuck, and non-stuck. This research also has gone through an optimization process which is vital in machine learning techniques to construct the most practical models. It showed that both ANNs and SVMs are able to predict stuck pipe occurrences with reasonable accuracy of over 83%.

It has been shown in this study that the competitive SVM technique is able to generate promising and reasonable results of stuck pipe prediction. Besides, it can be found that SVMs are more convenient than ANNs since they need one or

two parameters at most to be optimized. The constructed models generally apply very well in the areas for which they are built but may not work for other areas. However, they are important especially when it comes to probability measures. Thus, they can be utilized with real-time data and would represent the results on a log viewer.

LIST OF TABLES

4.1. Classification of Group_A data	46
4.2. Number of datasets for learning and testing in Group_A data	47
4.3. Classification of the data in Company_X group	48
4.4. Percentage of each stuck category in Company_X datasets.....	49
4.5. Number of datasets for learning and testing in Company_X data.....	52
4.6. Parameter selection for both Group_A and Company_X data	53
4.7. Number of nodes or neurons for each layer of the constructed ANNs' models	55
4.8. Scenarios of the selected activation functions for the constructed ANNs' models	57
4.9. MATLAB codes utilized to build MATLAB built-in model of ANNs	59
4.10. MATLAB codes utilized to build MATLAB built-in model of SVMs	62
4.11. MATLAB codes for the SVM model from library of support vector machines	63
A.1. Outputs of all designed models on all datasets of Group_A.....	111
A.2. Outputs of all designed models on mechanical stuck datasets of Group_A.....	112
A.3. Outputs of all designed models on differential stuck datasets of Group_A.....	113
A.4. Outputs of all designed models on datasets of Company_X Field no. 6.....	114
A.5. Optimization of SVM_Linear model for Group_A datasets	115
A.6. Optimization of SVM_RBF model for Group_A datasets	116
A.7. Optimization of SVM_Linear and SVM_RBF models for Company_X Field no. 6 datasets.....	117
A.8. Summary of runs by ANN_MATLAB and SVM_MATLAB on different fields of Company_X datasets	118

LIST OF FIGURES

1.1. Top view of differentially stuck pipe. <i>Source: 1997 Drillers Stuck pipe Handbook</i>	3
1.2. Cross sectional view of differentially stuck pipe. <i>Source: http://www.drillingformulas.com/</i>	4
3.1. A schematic of how basically machine learning techniques work	23
3.3. Unipolar logistic function.....	29
3.4. Bipolar sigmoid function.....	30
3.4. Tangent hyperbolic function.....	30
3.5. Explanation of how the error is minimized in the error function	31
4.1. Mapping space of support vector machines	34
4.2. Linearly separable case of support vector machines.....	35
4.3. Support vectors in linearly separable case of support vector machines	40
4.4. Linearly non-separable case of support vector machines.....	41
4.1. Percentages of Group_A data distribution.....	47
4.2. Percentages of Company_X data distribution	49
4.3. Causes of stuck in Company_X database.....	50
4.4. Design scenarios of SVMs' models	63
5.1. Accuracy of the models when applied in all datasets of Group_A.....	68
5.2. Accuracy of the models when applied in Group_A differential stuck datasets.....	70
5.3. Accuracy of the models when applied in Group_A mechanical stuck datasets.....	71
5.4. Accuracy of the models when applied in Company_X-Field no. 6.....	73
5.5. Average accuracy of MATLAB_SVM_ANN in an absence of each parameter for all datasets of Group_A when classifying the data into 2 groups.....	76
5.6. Average accuracy of MATLAB_SVM_ANN model in an absence of each parameter for the mechanical stuck datasets of Group_A	78
5.7. Average accuracy of MATLAB_SVM_ANN model in an absence of each parameter for the differential stuck datasets of Group_A.....	79

5.8. Weighted factor of each drilling parameter by SVM_Linear model for all datasets of Group_A	81
5.9. Weighted Factor of each drilling parameter by SVM_Linear model for the mechanical datasets of Group_A	82
5.10. Weighted factor of each drilling parameter by SVM_Linear model for the differential stuck datasets of Group_A	83
F5.11. Weighted factor of each drilling parameter by SVM_Linear model for Company_X field no. 6 datasets	84
5.12. Number of learning datasets vs. average accuracy of MATLAB_SVM_ANN model	86
5.13a. Sensitivity of the size of the training data with respect to accuracy in Field_6 of Company_X	88
5.13b. Sensitivity of the size of the training data with respect to accuracy in Field_6 of Company_X with more run scenarios	89
5.14. Sensitivity of the size of training data in different fields (2, 4, 6, 7, and 8) of Company_X	90
5.15. Field effect of Company_X data in the accuracy of SVM and ANN models	91
5.16. Applying datasets of Field_6 on same field and different fields Datasets of Company_X	92
5.17. Sensitivity of the size of the training data when applied in the same mechanical stuck category (Packoff)	93

CHAPTER I

INTRODUCTION

The Petroleum industry is one of the biggest industries on earth and it has a very significant impact on the world economy. Its business spreads over continents and it controls the most important resource of nonrenewable energy. One of the major aspects of the oil and gas industry is drilling. Drilling operations are considered to be the most expensive operations in the world as they require huge amount of money to be spent daily. Each day hundreds of oil and gas wells, both onshore and offshore, are being drilled by drilling rigs. The oil and gas industry is like any industry in the world, which involves many problems in its operations. However, drilling downhole problems are ranked among the most serious problems as they result in huge expenditures. Although all downhole drilling problems are challenging, unpredictable, and involve spending lots of time and money, stuck pipe is considered one of the most difficult drilling problems. A stuck pipe situation can take place anywhere in the world where drilling operations are conducted and it has a very long history.

Several authors have come up with some statistics that show the severity of substantial losses due to stuck pipe. In 1991, Bradley et al. conducted some research and found that BP (British Petroleum) had spent more than \$30 million per year for stuck pipe issues. Between 1985 and 1988, an average of \$170,000 was spent per well due to stuck pipe. In addition, they stated that stuck pipe charges in the entire oil industry were estimated to exceed \$250 million yearly. On the other hand, a survey

within Sedco Forex in 1992 showed that stuck pipe accounts for 36% of total drilling problems (Jardine et al.).

Pipe sticking is a common problem in drilling operations and no place in the world is exceptional. However, some places are more affected by this problem than the others, and that might be due to the nature and structure of drilled formations. For example, during 1980s, the Gulf of Mexico and North Sea were the places most affected by stuck pipe. Accordingly and due to the severity of this issue, during 1980s many scholars in the oil industry had conducted deep researches to mitigate this problem and many practices had been published to prevent it. Task forces were created for spreading stuck pipe awareness and knowledge among drilling teams especially rig crews. Consequently, the effort was paid back and pipe sticking incidents were reduced in most affected places specifically the Gulf of Mexico.

1.1 What is Stuck Pipe?

Stuck pipe can be described such that during drilling a well, the up or/and down pipe movement or/and pipe rotation is suddenly restricted or frozen. Thus, drilling operations are suspended as no more drilling is achieved. Pipe sticking could occur due to many causes. It can be because of improper drilling fluid properties, well geometry, the nature of the drilled formation, and/or inappropriate drilling parameters. In drilling industry, stuck pipe has been classified into several types according to different basis and categories. Generally, the most common two types of stuck pipe in the literature are differential and mechanical.

1.2 Differential Sticking

From its name, differential stuck pipe has a direct relation to differential pressure between the hydrostatic column created by the drilling fluid in the well and formation pressure. Differential sticking occurs when high overbalanced drilling fluid exerts a huge amount of differential pressure across a permeable zone, usually a depleted reservoir. That in turn would lead to losing drilling filtrate in that permeable formation and leaving solids with a thick filter cake to build up between the Bottom Hole Assembly (BHA) and the depleted zone. Once that part of BHA, which is exposed to the depleted formation, is embedded sufficiently by the thick filter cake, drillpipe movement would be restricted; and hence sticking would occur due to a huge amount of friction. Figure 1.1 shows a top view of typical differential stuck pipe problem that occurs around the drill collar (the stiffest part of drill string and located at the bottom of BHA).

Figure 1.1. Top view of differentially stuck pipe. *Source: 1997 Drillers Stuck pipe Handbook*

From figure 1.1, P_h depicts the hydrostatic pressure exerted on the drill collar, whereas P_f represents the formation pressure which is usually much less than P_h in this problem. In addition, figure 1.2 shows a cross sectional view of differential stuck pipe. Alternatively, the following figure 1.2 shows a cross sectional view of differential sticking case.

Figure 1.2. Cross sectional view of differentially stuck pipe. *Source:*
<http://www.drillingformulas.com/>

It can be seen from figure 1.2 that the pipe is pushed to the left side because of high overbalance or differential pressure exerted on the right side of the pipe.

If the severity of differential sticking is high, rotation cannot even be initiated and circulation might not be possible (Siruvuri et al., 2006). Therefore, Drilling fluid plays a major role in leading to differential sticking, and thus proper design of drilling mud would significantly protect the pipe not to get differentially stuck.

1.3 Mechanical Sticking

This type of sticking can be any stuck pipe other than the differential one.

According to drillers stuck pipe handbook (1997) by Schlumberger, mechanical sticking can be caused by the following conditions:

1. Inadequate hole cleaning
2. Formation instability (brittle, sloughing, or swelling shales)
3. Key seating
4. Undergauge hole
5. Tectonically stressed formations
6. Plastic or mobile formations
7. Underpressured formations
8. Junk
9. Ledges and doglegs
10. Collapsed casing/tubing
11. Unconsolidated formations
12. Large boulders falling into the hole
13. Running large gauge tools
14. Cement blocks
15. Green cement

This kind of stuck pipe can be avoided by proper well planning, optimal mud design, and right directional plan. However, pipe can become stuck even though protective measures have been already taken, according to DeGeare et al., 2003.

1.4 Practices Used in Freeing Stuck Pipe

Several approaches have been developed to avoid or prevent stuck pipe. Some of them are considered reactive, where the action is made after the pipe got stuck. However, some other approaches are considered proactive that utilize effective techniques to predict or detect stuck pipe before it occurs. The proactive way of predicting or detecting stuck pipe is discussed in the literature review chapter since this research is based on one of its techniques.

When pipe gets stuck, everybody involved in that operation is being in tension as nobody is aware how long the sticking problem is going to last. Many practices have been published and well written to mitigate stuck pipe issue. Usually when the pipe suddenly gets stuck, the first action is how to get it free as quickly as possible. Regardless of the cause of sticking, the most common action in drilling industry to free the pipe is to work it out by firing the jars, which are parts of drilling assembly. Jars exert huge amount of energy on drill string, which can be transferred to an impact force that may be able to free the pipe. Depending on the cause of the sticking, jars may be applied downwards or upwards. However, Jars sometimes are ineffective especially when they are placed below stuck point, which can be due to poor estimation of stuck zone or lacking of proper field experience. However, jarring operations might last days or sometimes weeks, which maximizes the loss time, according to Gonzalez et al., 2007.

Resonant vibration can be alternative to jarring and some scholars claim that it is more effective especially when it is installed and used directly after sticking occurs. Gonzalez et al. stated that this technique has advantages since it quickly frees the pipe

and it is easy to apply. It doesn't need any hole intervention since it is operated from surface. However, its effectiveness is minimized when the time of sticking exceeded 24 hours without applying this technique.

One of the very common practices to free the pipe particularly when it is differentially stuck is pumping chemical pills. These pills are prepared on surface and sometimes during working the pipe out. Spotting fluids can be oil base, water base, or acid base. They are pumped to be placed in the point of sticking. When they reach the stuck place, chemical pills are left to soak with filter cake for a while and sometimes the exposure time might be for hours. Oil base is used as lubricant between the portion of stuck pipe being stuck and filter cake. However, many researchers have proposed water based pills since oil base is not friendly with the environment especially in offshore operations; since disposing the pills would kill some creatures in the sea and damage the environment. Halliday and Clapper in 1989 specified the features of spotting fluid as to be 1) with low toxicity, 2) water miscibility, 3) adequate barite suspension, 4) no adverse effects, 5) moderate cost. On the other hand, Montgomery et al. in 2007 proposed the use of chelation fluid to increase the permeability of filter cake by the means of dissolving the weighting materials formed by barite, a major ingredient of filter cake; this reduces the shear strength and causes the filter cake to crack. Alternatively, it is a common practice for drilling people to prepare Hydrochloric Acid pill to soak across the filter cake. Hence this pill would reduce the contact area for the purpose to free the pipe. This technique is well known, commonly practiced, and already showed its effectiveness.

In 2008, Sharad and Agarwal proposed a new design of drill collar that would reduce the risk of differential sticking. This drill collar was designed in a way that could reduce the contact area with the filter cake, which in turn would reduce the force required to free the pipe. The authors claimed that this drill collar has additional benefits over the conventional drill collars. It could prevent stuck pipe, avoid usage of additional reamers, and prevent mud cake formation.

Efforts have been made throughout task forces to maximize the knowledge of rig crew to be able to identify the type or the cause of sticking by following the chain of events. Comprehensive training courses were also designed for drilling team to be aware of the warning signs and rig site indications that would lead to stuck pipe. Hence, preventive actions would be made to avoid such an incident to occur. In 1986, Courteille et al. proposed a computerized system called Drilling Adviser or SECOFOR that would provide some consultations to drilling team, and identify the drag forces and type of pipe sticking. The system is able to diagnose the problem based on the entered data and recommend some solutions and treatments. Courteille et al. stated that this system was not designed to replace human experts; instead it was built to support the drilling people to come up with reasonable decisions. In contrast, recently there is a usage of decision trees which consist of nodes that involve the causes of stuck pipe and would lead to a final decision identifying the type of sticking.

Sticking situation can be evaluated to measure the risk of freeing the pipe. T. Love in 1983 published a paper regarding an empirical equation that predicts the chances of freeing the stuck pipe and called it Stickiness Factor. The equation was applied in Gulf of Mexico wells. The author showed that the stickiness factor is

increased by increasing the hole angle, mud weight, open hole size, API fluid loss, and the length of bottom hole assembly.

If all attempts, that utilize all possible methods to free stuck pipe, are not feasible, backing off operation is resorted. In order for the pipe to be backed off, the free point of stuck pipe has to be estimated. Two approaches are utilized to locate the free point: pipe stretch calculations and free point logs. If the backing off operation cannot be achieved, pipe can be cut by some cutter tools. When the pipe is backed off and pulled out to surface, the fish or the remaining pipe in hole will seek fishing operations. The fish in hole can sometimes be easily retrieved and sometimes hardly, which makes the operation to last for several days. Alternatively, drilling people might do fishing calculations which estimate the feasibility to recover the fish in hole, otherwise sidetrack would be initiated.

1.5 Statement of the Problem

The purpose of this study is to utilize the two most powerful machine learning techniques i.e., artificial neural networks and support vector machines, to predict stuck pipe occurrences. The research is also intended to show

- a) how effective machine learning techniques with regard to classification and prediction of stuck pipe, and
- b) which method is more powerful than the other in terms of accurate measurement of correct classification, and which one has the least error between the true and predicted labels.

The study would also

- c) measure the significance or relative importance of the specified drilling and mud parameters with respect to the prediction process by the machine learning methods, and
- d) show the sensitivity of the size of the training datasets with respect to the accuracy of the models.

CHAPTER II

LITERATURE REVIEW

2.1 Stuck Pipe Predictive and Deductive Approaches

While in the previous section, a survey of the literature about the practices to free the stuck pipe after its occurrence was discussed, this literature review will specifically focus on the proactive approaches to avoid and predict such an incident to occur. Most of practices in oil and gas industry to deal with or mitigate pipe sticking are reactive which provide protective actions after the pipe gets stuck. Montgomery et al. stated that differential stuck pipe still occurs everywhere and the methods to free the pipe have not been much effective. In contrast, proactive approaches play an important role to detect or predict pipe sticking before it occurs. Various techniques have been utilized to predict the chances or probabilities for the pipe to become stuck. One of the major predictive techniques is Multivariate Statistical Analysis (MSA). However, recently some researchers have tried to touch or test machine learning methods to measure the probability for the pipe to become stuck. This literature review will cover the main researches have been done in this area since the first attempt in 1980s.

2.2 Multivariate Statistical Analysis

One of the major and early stuck pipe prediction techniques is Multivariate Statistical Analysis. Historically this technique was developed in 1930s. The idea of using MSA to predict stuck pipe occurrences was initially brought by Hemphkins et al.

through a research done in 1985. Hemphkins et al. claimed that before 1985 there had not been any statistical analysis that could bring some study to stuck pipe avoidance. Their study was applied on wells in Gulf of Mexico. One major drive that forced the scholars in oil industry to explore several proactive approaches to predict stuck pipe is the high rate of sticking problems occurred in early 1980s especially in Gulf of Mexico and North Sea (Hempkins et al.). The authors stated that, between 1981 and 1984, almost 131 stuck pipe incidents reported in Gulf of Mexico.

The research of Hemphkins et al. included 131 stuck pipe cases and 20 drilling variables. They utilized discriminant analysis to develop discriminant functions which are set of equations derived from correlations based on relationships between dependent and independent drilling parameters that would lead to stuck pipe. These functions are developed to classify the input data in accordance to values of their parameters into three groups; mechanical stuck, differential stuck, and non-stuck pipe. The authors found that discriminant analysis brought a success rate of 81-87%. In other word, the model has an ability to correctly classify the data into their predetermined groups by 81-87%. Moreover, the researchers extended their work by attempting to optimize the drilling parameters throughout monitoring drilling operations in order to shift or move the outcome from stuck region to non-stuck region.

In 1994, Howard and Glover applied the same technique of Multivariate Statistical Analysis to predict stuck pipe. However, they applied their model on databases that involve more than 1000 wells drilled in Gulf of Mexico and North Sea. They considered 40 drilling variables for their research. Like Hemphkins et al., Howard

and Glover classified the data of the study into three groups; mechanical stuck, differential stuck, and non-stuck. They achieved 75% of success rate when classifying the data into these three categories and 80% when classifying them into only stuck and non-stuck. The scholars also showed that MSA can be used to measure the probability of freeing the pipe when gets stuck and they called that function FI (Freeing Index). Furthermore, the discriminant analysis plots and graphs built by Howard and Glover can be used to optimize the drilling parameters during drilling by moving the well from stuck zone to non-stuck zone.

In the same year of 1994, Biegler and Kuhn constructed a model that could predict stuck pipe using the same technique of MSA. Their model included physical parameters of stuck pipe and large drilling datasets. Biegler and Kuhn claimed that their model has a capability to not only predict or detect stuck pipe, but also can identify the driving mechanism of pipe sticking. Thus, the current model can be utilized to optimize the drilling parameters and lead to stuck avoidance. However, their model was limited to water based mud; and commercial software was used to analyze the data. There were 8 independent meaningful physical variables selected for each well. The model is capable to optimize these variables during well planning to minimize the risk of stuck pipe.

A recent work in 2011 done by Shoraka et al. that demonstrated the usage of multivariate statistical regression and discriminated analysis to predict stuck pipe. Their developed model was applied in Iranian fields. They implemented two types of multivariate techniques: regression analysis and discriminant analysis. The former was used for stuck pipe prevention while the latter was applied to develop functions that

can detect differences in drilling variables between wells. The discriminant functions' plots constructed in the study could be used to move well from stuck zone to non-stuck zone. Shoraka et al. found that discriminant method is better and more accurate than the regression method since it further estimates the conditions and includes more variable consistency. They found that discriminant functions have the power to predict stuck pipe before three days whereas regression functions can predict the incident only before one day. They stated that since this study uses statistical analysis, it contains errors. However, their goal was to minimize the errors involved. The authors found that the model created in the study came up with 80 to 86% of a success rate to classify the data into three groups of stuck pipe. Like the previous work done with MSA, Shoraka et al. demonstrated that their model has the ability to optimize the drilling parameters during drilling when a well is moving toward stuck zones.

2.3 Machine Learning and Soft Computing

Siruvuri et al. in 2006 were the first scholars who utilized Artificial Neural Networks (ANNs) to predict stuck pipe occurrences. They stated that artificial intelligence techniques particularly neural networks have been widely applied in different fields of sciences including petroleum industry. However, the technique has limited implementations in drilling industry. The researchers claimed that neural networks are more powerful than the previous methods used to identify the importance of stuck pipe parameters, such as multivariate statistical analysis and simulated sticking tests using different drilling fluids. Based on their research, Siruvuri et al.

have observed that ANNs are a practical technique in terms of reasonable outputs even though the data might be incomplete or have some errors.

The research done by Siruvuri et al. was limited to differential stuck pipe cases. They constructed an ANN model that includes 200 datasets, 120 reported as stuck pipe and 50 as non-stuck pipe. Also, 45 datasets were assigned for cross validation. They specified 10 drilling parameters as input variables for the model. These drilling and mud parameters were selected based on their connection to differential sticking. In this research, commercial NN (Neural Network) software was utilized to analyze the datasets. The model was designed based on three layers: input, hidden, and output. The number of hidden units was assigned automatically in accordance to the strength of the data. Seven error measures were considered to calculate how accurate the model is.

According to their research, Siruvuri et al. observed that the proposed method has an accurate prediction with respect to stuck pipe incidents. The accuracy of this model depends on the proper selection of drilling parameters and the size of the database. Also, they claimed that this technique has the power to represent the output on a log viewer which allows the user to monitor the risk of stuck pipe based on real time data. However, Siruvuri et al. stated that better prediction of stuck pipe can be achieved by further research that would include proper construction of neural network model, which can involve improved algorithms and optimal design.

Miri et al. in 2007 made another attempt of utilizing ANNs in order to predict stuck pipe incidents. They claimed that neural networks have the ability to predict stuck pipe incidents not only during well planning, but also during drilling operations.

They stated that neural networks are capable to identify the cause of the stuck pipe if the model has been constructed with the proper design. Therefore, proper action, treatment, or mitigation method could be applied. The authors constructed a neural network model that contained 109 datasets, where 61 reported as differentially stuck pipe and 48 as non-stuck pipe. The data were collected from 32 wells that experienced stuck pipe incidents and 31 wells that did not. The datasets belonged to different fields which are located in Persian Gulf. The model constructed by Miri et al. had the same number of layers, which is the same as the model utilized by Siruvuri et al. Further, its implementation was limited to differential stuck pipe incidents. They considered 9 drilling and mud characteristics as inputs. They also constructed training algorithms that involved feed forward networks, Multi Layer Perceptron (MLP) and Radial Basis Functions (RBFs) with Backpropagation.

In their research, Miri et al. tried to optimize the model to get the proper and reasonable outputs. They optimized the number of hidden layers and hidden neurons. They also selected the proper activation functions for the output and hidden layers. Besides, cross validation process was utilized to sort out the memorization issue, stop the training, and thus get the best generalization with the least error. Generalization on the other hand is the production of reasonable outputs in case the inputs are not encountered while training. Several tests have been applied in the model to remove the noisy datasets that could lead to poor generalization. Further, five error measures were specified in the model to calculate how accurate the results are.

The authors extended the research into a sensitive analysis between the inputs and outputs utilizing NNs. This analysis shows the effect of input parameters to stuck

pipe. It also could identify the drilling parameters that have significant effect on stuck pipe. Based on their research, R. Miri et al. found that ANNs are a powerful tool to predict stuck pipe incidents. The technique that they utilized can be used to determine the probability of differential sticking during well planning of Persian Gulf wells. They claimed that proper structure, training, and large datasets would lead to an accurate prediction.

In 2009 Murillo et al. conducted a research that included two models of machine learning to predict stuck pipe incidents. In addition to the neural network technique, Murillo et al. utilized another soft computing method which is fuzzy logic. They were the first scholars who utilized such technique to predict stuck pipe. They also utilized discriminant analysis to classify the data of stuck pipe. Fuzzy logic, on the other hand, is considered as a complimentary tool to discriminant functions to predict stuck pipe occurrences. In their project, Murillo et al. considered 185 datasets with 18 drilling parameters and mud properties as inputs. The data were classified into three groups of stuck pipe: 59 differential, 68 mechanical, and 58 non-stuck. The researchers applied different runs of discriminant functions to classify the data. They found that such analysis brings an accuracy of 98.4%.

In addition to stuck pipe prediction, Murillo et al. utilized neural networks and fuzzy logic to optimize the input parameters in order to get stuck pipe case shifted or moved from stuck region to non-stuck region. However, the authors reported some misclassified data in both artificial intelligence models.

Murillo et al. found that fuzzy logic, neural networks, and discriminant functions are powerful tools to predict stuck pipe incidents into differential,

mechanical, and non-stuck. They also found that neural network results had less misclassified data than fuzzy logic. The authors claimed that the utilized models in their research could be applied not only during well planning, but also during drilling operations through real-time data. That approach would add a significant value when it comes to making instant decisions to avoid stuck pipe occurrence.

2.4 Other Predictive and Detective Methods

Another approach to detect stuck pipe is by utilizing friction monitoring system. In 1992, a research, done by Jardine et al., revealed a system that used real-time surface data to predict some changes in hole conditions that would lead to stuck pipe. The key point in this research was automatic processing of represented data of hook load and torque to detect any anomalies; which would be in turn an indication of stuck pipe. The authors showed that this system had an advantage of displaying the results in the surface on a log scale. Therefore, it could allow the drilling personal to detect and identify any abnormal or irregular change in an early stage. Furthermore, it has an ability to warn the driller with an alarm that indicates there is downhole problem or risk of sticking while tripping.

Meschi et al. conducted a research in 2010 that contained analytical approach to detect if suck pipe would occur. They came up with a parameter called Reducing Stuck Index (RSI). This index has proportional relations to some mud parameters and inverse relations to others. The study was applied on Ahwaz Iranian oil field in member no. 7 of Gachsaran formation. The data collected were based on mud logging and daily drilling reports of 75 wells, and compared for stuck and non-stuck situations

at different depths. A representative plot that showed RSI vs. Depth was developed with a safe window. This window represents that the risk of stuck pipe is very low, whereas zones above or below it indicate a high risk of stuck pip.

CHAPTER III

MACHINE LEARNING AND SOFT COMPUTING

3.1 Background

Soft computing, machine learning, artificial intelligence, data modeling, and smart machines are names given to those computational means, methods, techniques, or tools that can learn from previous experience through sets of learning data.

Machine learning techniques are constructed based on mathematical models. These models are able to learn the trend or behavior of experimental or real data; and thus create a pattern accordingly. These methods are called machine learning or soft computing due to their capabilities to learn from certain datasets of previous experience. Based on their learning capabilities, the models of these methods are able to generate generalization, approximation, or interpolation in a specified problem; that includes certain number of datasets with outputs that indicate the occurrences of the problem. Consequently, the model is able to initiate a prediction process from new datasets and also make decisions.

Learning from experience data goes back to late 1960s when Frank Rosenblatt built the first learning machine called perceptron which can learn from experimental data. That kind of achievement is considered as the cornerstone of the world of machine learning techniques. That machine was designed for pattern recognition tasks. However, in 1968 V. Vapnik and A. Chervonenkis brought the theory of utilizing the statistical learning techniques. After that, the learning techniques had gone through

tremendous improvements and development; and they were driven from the theoretical arena to the practical applications.

Many scientific and engineering fields have recently applied artificial intelligence to predict common and serious problems. They seek soft computing or machine learning methods due to the complications of today's problems which are hard to be solved through the traditional methods or what is called hard computing, according to Vojislav Kecman. Many contemporary problems are considered complicated enough not to have an exact or precise solution such as weather forecasting, stock market prediction, or pattern recognition of handwriting and speech. These problems due to their uncertainties and imprecision would require a certain method that can fulfill their complexities such as soft computing techniques; since the traditional hard computing or analytical methods are not capable enough to handle them. Kecman stated that sometimes the solution of a certain problem could be exact but its cost is quit high, while this problem can be solved with alternative method that would give less precision but with acceptable accuracy. Machine learning methods can be an option of these techniques that can meet these requirements of solutions with reasonable accuracy. In addition, artificial intelligence techniques have been considered effective when they have been applied to many contemporary complicated problems. Kecman stated that artificial intelligence methods may not need to be pursued if the analytical approach is able to solve the problem with reliable and approved functions in short time and cost effective way. However, he explained that due to the complexity of today's problems, which involve multi-dimensional parameters, would lead to the necessity of utilizing these systems of machine learning.

The machine learning techniques are sometimes called artificial intelligence methods since they fairly follow and mimic some of human properties and abilities such as learning, generalization, memorization, and prediction. However, scholars found that smart machines need to be improved and developed in such a way they can detect noisy data and faults, and also can plan under large amount of uncertainties, (Kecman).

The main objective of seeking smart machine methods is to predict the occurrence of some problems based on previous experience with reasonable cost and time. The reliability of the method depends on the accuracy of prediction and the error between the actual and the predicted class labels of the problem.

The most common machine learning techniques are Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs). Both of these tools are designed mainly for classification or pattern recognition. According to Kecman, these two computing methods are considered complimentary to each other rather than competitive. The difference between these two methods is in their mathematical approach and construction. Thus, each technique would initiate a pattern that could produce outputs of a certain problem. Figure 3.1 illustrates how the processes of learning and testing are in both ANNs and SVMs.

Figure 3.1. A schematic of how basically machine learning techniques work

The variations between the outputs of these tools can be obtained by the means of accuracy measures and error calculations. SVM technique was not very much appreciated when it was brought to the industry at the beginning; as there was a belief that this tool is not practically suitable for many contemporary problems. However, nowadays, the two techniques –ANNs and SVMs- are utilized in many fields such as biology, sociology, medicine, finance, engineering, and many other scientific fields.

3.2 Applications

Many applications in the most important and critical fields have been implementing artificial intelligence techniques to solve complex nonlinear problems that involve parameters in high dimensional space. Due to the complexity of these applications, it can be difficult for classical statistical tools to come up with solutions for such problems with reasonable accuracy. Machine learning techniques have been

applied in many of these high dimensional space problems and showed promising results. Smart machines are effectively used in pattern recognition of visual, face, and sound. They are also used in forecasting and time series problems such as weather or stock prices in stock markets, and many other forecasting areas in finance or other fields. In addition, they are widely implemented in robotics for control, navigation, coordination, and object recognition, according to Kecman. They are also being applied in medicine for diagnosing purposes and predictions of the common diseases such as heart attack, high blood pressure, diabetes, and other serious medical disorders. They are developed and operated in different practical applications of engineering for classification e.g. satellite images, prediction, optimization, and process control such as control of chemical plants, power stations, missiles, (Kecman).

3.3 Experimental Data

In both ANNs and SVMs, the learning process depends on sets of data that belong to a certain problem. Basically, the problem can be represented with the following function:

$$Y = f(x)$$

Where Y is dependent variable or output and x is set of independent variables or inputs. The learning datasets have to be with sufficient size or capacity that can allow the supervised tools or machine learning techniques to build a pattern which is usually nonlinear and with multidimensional parameters. As per Kecman, the training datasets can be represented with the following formula:

$$T = \{(x_i, y_i) \in X \times Y\}, i = 1, S$$

Where:

S: the number of the datasets

Kecman emphasized that the size of the training datasets for all data modeling techniques has to be large enough in order for the created pattern to be able to predict the outputs reasonably. This, in turn, will increase the learning capacity of the built model and thus would reduce the error accordingly. Otherwise, the learning tools would produce improper results since the error is inversely proportional to the size of learning datasets, according to Kecman. In addition, learning process can be off-line or sometimes called explicit and on-line or implicit. It is considered off-line when the learning date sets are run for one time without iteration or updates. Conversely, the learning can be with implicit method when the weighted factors are updated through sweeps of iterations (Kecman).

Testing samples or training datasets are utilized in the machine learning methods to test the reliability or accuracy of the built model. Based on the power or capacity of the learning process, the data modeling tools could predict the outputs of the training datasets reasonably. The error thus could be measured with the error functions which calculate the difference between the true or actual indices and the predicted or estimated values of a certain problem.

3.4 Artificial Neural Networks

3.4.1 Background

Neural networks are inspired by human brain function. Whereas human brain consists of hundreds of billions of neurons and each one is considered as processing

unit, computers usually consist of one processor. However, these neurons are simpler than computer processors, according to Ethem Alpaydin. Artificial neural networks, on the other hand, encompass many neurons connected together through weights which are similar to the Synapses that connect biological neurons in the human nerves system. These weights represent the parameters of the problem.

The earliest work of neural networks goes back to 1943 when the physiologist Warren McCulloch and the scientist Walter Pitts introduced the first neural model to the industry. The earliest artificial neural networks were the Perceptron and proposed in 1958 by the psychologist Frank Rosenblatt, as per Daniel Graupe. In 1980s, there was huge development of artificial neural networks in the same time that the computers with sophisticated processors were handy and available for the individuals in the market. During that period, many scholars such as Le Cun in 1985, Parker in 1985, and Rumelhart et al. in 1986 revealed researches on backpropagation network which provides solution to the learning or training process of the hidden layer weights.

3.4.2 Mathematics of ANNs

The most common type of artificial neural networks is the Multilayer Perceptrons. These models consist of three layers: Input, Hidden, and Output. Each of these layers consists of neurons (sometimes called nodes). The input layer is the first layer which has neurons that represent the inputs or parameters of a certain problem. The second layer is the hidden layer where the computational process is initiated in. Depending on the complexity and optimization of each problem, ANN model can have single or multiple hidden layers. Like other layers in ANNs, each hidden layer consists

of certain number of neurons that are connected by weights. The output layer represents the labels or outputs of the problem, and consists of certain number of neurons based on the nature of the problem. Figure 3.2 shows the typical architecture of the Multilayer Perceptrons model which is the most common type of ANNs.

Figure 3.2. Architecture of Artificial Neural Networks

From Figure 3.2:

x_i = input unit i

z_i = hidden unit j

y_k = output unit k

v_{0j} = bias on hidden unit j

w_{0k} = bias on output unit k

Model inputs are represented by X where:

X= matrix of all inputs

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & \dots & \dots & x_{1j} \\ x_{21} & x_{22} & x_{23} & \dots & \dots & \dots & x_{2j} \\ x_{31} & x_{32} & x_{33} & \dots & \dots & \dots & x_{3j} \\ \vdots & \dots & \dots & & \ddots & & \vdots \\ x_{i1} & x_{i2} & x_{i3} & \dots & \dots & \dots & x_{ij} \end{bmatrix}$$

Where:

i = number of training datasets

j = number of inputs or parameters of the problem

Also:

Net input to Z_j : z_j

$$z_j = \sum_{i=0}^n x_i v_{ij}$$

Output signal (activation) of z_j : z_j

$$z_j = f(z_j)$$

Net input to Y_k : y_k

$$y_k = \sum_{j=0}^p z_j w_{jk}$$

Output signal (activation) of y_j : y_j

$$y_k = f(y_k)$$

3.4.2.1 Activation functions

Models of ANNs contain an activation function in each hidden layer and output layer. Thus, each neuron at certain layer is processed with the same activation function that the other neurons in the same layer are. In the hidden layer, the role of the activation functions is approximating the solution of the problem. Each activation function should be continuous, differentiable, and monotonically non-decreasing. Also, it is desirable that its derivative is easy to compute. Depending on the values of the output labels, each problem has to be optimized with the proper activation function that produces the most accurate results with the least error. Examples of the most common activation functions utilized in the multilayer perceptron model are squashing sigmoidal functions: unipolar logistic function and bipolar sigmoidal functions such as tangent hyperbolic. Whereas unipolar logistic function maps the results in values from 0 to 1, bipolar sigmoid or tangent hyperbolic function maps the result from -1 to 1 as shown in the following figures 3.3 - 3.5.

$$f(x) = \frac{1}{1+e^{-x}} \text{ (Unipolar logistic function)}$$

Figure 3.3. Unipolar logistic function

$$f(x) = \frac{1}{1+e^{-x}} - 1 \text{ (Bipolar sigmoid function)}$$

Figure 3.4. Bipolar sigmoid function

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \text{ (Tangent hyperbolic)}$$

Figure 3.5. Tangent hyperbolic function

The basic derivation of unipolar logistic and bipolar sigmoid activation functions is:

$$f'(x) = f(x)(1 - f(x))$$

There is no certain criterion that specifies the proper activation function for a certain problem. However, the right selection of the activation function can be through optimization process by trial and error.

3.4.2.2 Gradient descent

The following figure 3.6 shows the error function followed by the procedure of how the error can be minimized toward the least error.

Figure 3.6. Explanation of how the error is minimized in the error function

$$x^{i+1} = x^i - \alpha \nabla g(x^i)$$

Thus, the error is $E = \frac{1}{2} \sum_{k=1}^m (t_k - y_k)^2$

$t_k = \text{true label}$

$y_k = \text{predicted label}$

$$y_k = f(y_{-k})$$

$$y_{-k} = \sum_{j=0}^p z_j w_{jk}$$

The error is minimized as follows:

$$\frac{\partial E}{\partial w_{jk}} = \left(\frac{\partial E}{\partial y_k} \right) \frac{\partial y_k}{\partial w_{jk}}$$

$$\frac{\partial E}{\partial w_{jk}} = -[t_k - y_k] f'(y_k) \frac{\partial y_k}{\partial w_{jk}}$$

$$\frac{\partial E}{\partial w_{jk}} = -[t_k - y_k] f'(y_k) Z_j$$

$$\frac{\partial E}{\partial w_{jk}} = -\delta_k Z_j$$

$$\frac{\partial E}{\partial V_{IJ}} = \sum_k \left(\frac{\partial E}{\partial y_k} \right) \frac{\partial y_k}{\partial V_{IJ}}$$

$$\frac{\partial E}{\partial V_{IJ}} = - \sum_k [t_k - y_k] f'(y_k) \frac{\partial y_k}{\partial V_{IJ}}$$

$$\frac{\partial y_k}{\partial V_{IJ}} = \left(\frac{\partial y_k}{\partial Z_J} \right) \frac{\partial Z_J}{\partial V_{IJ}}$$

$$\frac{\partial y_k}{\partial Z_J} = w_{JK} \frac{\partial Z_J}{\partial V_{IJ}}$$

$$w_{JK} \frac{\partial Z_J}{\partial V_{IJ}} = f'(Z_J) \frac{\partial Z_J}{\partial V_{IJ}}$$

$$\frac{\partial Z_J}{\partial V_{IJ}} = X_I$$

$$\frac{\partial E}{\partial V_{IJ}} = - \sum_k [t_k - y_k] f'(y_k) w_{JK} f'(Z_J) \cdot X_I$$

$$\frac{\partial E}{\partial V_{IJ}} = -\alpha_j X_I$$

The updates for the weights on the output units are as follows:

$$\Delta W_{JK} = -\alpha \frac{\partial E}{\partial W_{JK}}$$

$$\Delta W_{JK} = -\alpha (t_k - y_k) f'(y_k) Z_j$$

$$\Delta W_{JK} = -\alpha \sigma Z_j$$

The updates for the weights on the hidden units are as follows:

$$\Delta v_{ij} = -\alpha \frac{\partial E}{\partial v_{ij}}$$

$$\Delta v_{ij} = -\alpha f'(Z_j) X_i \sum_k \sigma_k w_{jk}$$

$$\Delta v_{ij} = -\alpha \sigma x_i$$

Therefore, the multilayer perceptrons model can be constructed through the following phases:

- Initialization of V and W to small random values for example [-1,1]
- Feed forward phase
- Backpropagation phase
- Computing the error at the hidden unit
- Updating the weights
- Testing phase

3.5 Support Vector Machines

3.5.1 Background

Support Vector Machines (SVMs) are considered as one of the major supervised machine learning techniques for nonlinear classification and regression. This method is based on statistical learning theory developed by Vapnik and Chervonenkis; and it was introduced in 1995 by Vladimir Vapnik at AT&T Bell Laboratory. Basically, SVMs map and transfer nonlinearly inputs into very high dimensional feature space where the inputs are separated linearly by a hyperplane as shown in figure 3.7. This transformation can be achieved by applying several nonlinear mappings such as polynomial or Radial Bases Functions. The major concept behind SVMs is to classify the data into two groups with the maximum margin between the boundaries that separate the data. The points where the decision lines are constructed on are called support vectors.

Figure 3.7. Mapping space of support vector machines

SVM technique has been applied in many critical fields and showed promising results and outperformance over ANNs in some applications. The modeling of SVMs is based on optimization process and utilization of kernel functions such as linear, RBF, or polynomial. That is why SVMs sometimes are called kernel machines.

3.5.2 Mathematics of SVMs

3.5.2.1 Linearly separable case

In the case of training data are linearly separable:

$\exists w_1 w_0$ such that

$$g(x_i) = w^T x_i + w_0 > 0 \quad \text{for } x_i \in C_1 (y_1 = +1)$$

$$g(x_i) = w^T x_i + w_0 < 0 \quad \text{for } x_i \in C_2 (y_2 = -1)$$

Figure 3.8. Linearly separable case of support vector machines

SVMs choose the maximum margin decision boundary as shown in figure 3.8. *Margin* is the smallest distance between the decision line and any of the samples. The distance of a point from a hyperplane is given by $\frac{|g(x)|}{\|w\|}$, in SVMs it is required to have:

$\max_{w, w_0} \frac{1}{\|w\|}$ such that:

$$w^T x_i + w_0 \geq 1 \quad \forall x_i \in C_1 (y_i = +1)$$

$$w^T x_i + w_0 \leq -1 \quad \forall x_i \in C_2 (y_i = -1)$$

$$y_i(w^T x_i + w_0) \geq 1 \quad \forall x_i$$

$$\Leftrightarrow \min_{w, w_0} \frac{\|w\|^2}{2}$$

Lagrange Multipliers has an objective to find the stationary point of a function of several variables subjected to one or more constraints.

Considering: $\max f(x) \quad s.t. \quad g(x) = 0$ (equality constraint)

$\nabla g(x)$ is orthogonal to the constraint surface $g(x)$

It is required to find x^* on the surface $g(x)$ s.t. $f(x)$ is maximized. Such a point must have the property that the vector $\nabla f(x)$ is orthogonal to the constraint surface.

$\nabla f(x)$ & $\nabla g(x)$ are parallel or anti-parallel

$$\exists \lambda \neq 0 \quad s.t. \quad \nabla f(x) = -\lambda \nabla g(x)$$

$$\nabla f(x) + \lambda \nabla g(x) = 0$$

λ : *Lagrange Multiplier*

Thus, *Lagrangian Function* $L(x, \lambda) := f(x) + \lambda g(x)$

$$\frac{\partial L(x, \lambda)}{\partial x} = \nabla f(x) + \lambda \nabla g(x) = 0$$

$$\frac{\partial L}{\partial \lambda} = g(x) = 0 \text{ (constant)}$$

$$\max f(x) \text{ s.t. } g(x) = 0$$

$$L(x, \lambda) = f(x) + \lambda g(x)$$

It is required to find x s.t. $\frac{\partial L}{\partial x} = 0$ & $\frac{\partial L}{\partial \lambda} = 0$

$$\frac{\partial L}{\partial X} = \nabla f(x) + \lambda \nabla g(x) = 0$$

$$\frac{\partial L}{\partial \lambda} = g(x) = 0$$

$$\max f(x) \quad \text{s.t.} \quad g_1(x) = 0, \quad g_2(x) = 0$$

$\nabla f(x)$ = a linear combination of $\nabla g_1(x)$ and $\nabla g_2(x)$

$$= -\lambda_1 \nabla g_1(x) - \lambda_2 \nabla g_2(x)$$

Therefore, $L(x_1, \lambda_1, \lambda_2) = f(x) + \lambda_1 \nabla g_1(x) + \lambda_2 \nabla g_2(x)$

$$\frac{\partial L}{\partial X} = \nabla f(x) + \lambda_1 \nabla g_1(x) + \lambda_2 \nabla g_2(x) = 0$$

$$\frac{\partial L}{\partial \lambda_1} = g_1(x) = 0$$

$$\frac{\partial L}{\partial \lambda_2} = g_2(x) = 0$$

In general:

$$\min f(x) \text{ s.t. } g_k(x) \geq 0, k = 1, \dots, K$$

$$L(x, \{\mu_k\}) := f(x) - \sum_{k=1}^K \mu_k g_k(x)$$

Finding the stationary point with respect to x such that:

$$\text{For } k = 1, \dots, K \quad \begin{cases} g_k(x) \geq 0 \\ \mu_k \geq 0 \\ \mu_k \cdot g_k(x) = 0 \end{cases}$$

$$\min_{w, w_0} \frac{\|w\|^2}{2} \quad \text{such that} \quad y_i(w^\tau x_i + w_0) \geq 1, \quad \forall i = 1, \dots, N$$

Therefore,

$$L(w, w_0, \lambda) := \frac{\|w\|^2}{2} - \sum_{i=1}^N \lambda_i (y_i (w^\tau x_i + w_0) - 1)$$

$$\text{Where: } \lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{pmatrix}$$

Applying Saddle Point Theorem such that:

$$(x^*, \lambda^*) = \text{optimum solution}$$

$$\Leftrightarrow L(X^*, \mu) \leq L(X^*, \mu^*) \leq L(X, \mu^*)$$

minmizing L w.r.t X

maximizing L w.r.t μ

Elminating w & w_0

Thus:

$$L(w, w_0, \lambda) = \frac{w^\tau w}{2} - \frac{1}{2} \sum_{i=1}^N \lambda_i y_i x_i^\tau \sum_{j=1}^N \lambda_j y_j x_j - \sum_{i=1}^N \lambda_i [y_i \left(\left(\sum_{j=1}^N \lambda_j y_j x_j \right)^\tau x_i + w_0 \right) - 1]$$

Therefore, the solution of L can be represented as:

$$L(w, w_0, \lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j x_i^\tau x_j$$

Where:

$x_i = \text{training data}$

$y_i = \text{lable data}$

$\lambda_i = \text{variable, Quadaric function}$

$$\max_{\lambda} L(\lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j x_i^T x_j \quad s. t. \quad \sum_{i=1}^N \lambda_i y_i = 0$$

$$\lambda_i \geq 0, i = 1, \dots, N$$

For a new test sample x ,

$$g(x) = w^T x + w_0$$

$$g(x) = \sum_{i=1}^N \lambda_i y_i x_i^T + w_0$$

$$g(x) \geq 0 \rightarrow +1$$

$$g(x) < 0 \rightarrow -1$$

Thus, the optimization satisfies the KKT (Karush–Kuhn–Tucker) conditions:

1. $\lambda_i \geq 0, i = 1, \dots, N$
2. $y_i(w^T x_i + w_0) - 1 \geq 0$
3. $\lambda_i(y_i g(x_i) - 1) = 0$

For each x_i , either $\lambda_i = 0$ or $y_i g(x_i) = 1$

For x_i associated with $\lambda_i = 0$, x_i does not play any role in prediction

For x_i associated with $\lambda_i > 0$, $y_i g(x_i) = 1$

The non-support vectors could be disregarded to find the decision boundary. The support vectors can be found from the solution of:

$$\lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{pmatrix}$$

Figure 3.9. Support vectors in linearly separable case of support vector machines

How to find w_0 ?

$$x_i \text{ is a SV} \Rightarrow y_i g(x_i) = 1 \quad \forall i \in S$$

$$\Rightarrow y_i \left(\sum_{j=1}^N \lambda_j y_j x_i^T x_j + w_0 \right) = 1$$

By multiplying both sides by $y_i = \begin{bmatrix} +1 \\ -1 \end{bmatrix}$

$$\Rightarrow \sum_{i \in S} \sum_{j=1}^N \lambda_j y_j x_i^T x_j + w_0 = \sum_{i \in S} y_i \quad \forall i \in S$$

Therefore,

$$w_0 = \frac{(\sum_{i \in M} y_i - \sum_{i \in S} \sum_{j=1}^N \lambda_j y_j x_j^T x_{i-})}{N_S}$$

$N_S = |S|$ (number of support vectors)

3.5.2.2 Linearly non-separable case

Unlike the previous case, the following figure 3.10 shows the points that belong to each group when they are linearly non-separable.

Figure 3.10. Linearly non-separable case of support vector machines

$\xi_n = 0$ is for data points on or inside the correct margin boundary

$$\xi_n = |y_n - g(x)|$$

Constraints:

$$y_n(w^T x_n + w_0) \geq 1 - \xi_n \quad n = 1, \dots, N$$

$$\xi_n \geq 0 \quad (\xi_n = \text{slack variables})$$

Objective,

$$\min \frac{\|w\|^2}{2} + C \sum_{n=1}^N \xi_n$$

C: is trade-off between the slack variable penalty and margin

Note:

x_n with $\xi_n 1$ is missclassified,

and $\sum_{n=1}^N \xi_n$ is an upper bound on the number of missclassified points.

When $C \rightarrow \infty$, hard margin case

Constraints: for $n = 1, \dots, N$ $\begin{cases} y_n g(x_n) \geq 1 - \xi_n \\ \xi_n \geq 0 \end{cases}$

Lagrangian Function:

$$L(w, w_0, \xi, \lambda, \mu) := \frac{\|w\|^2}{2} + C \sum_{n=1}^N \xi_n - \sum_{n=1}^N \lambda_n (y_n g(x_n) - 1 + \xi_n) - \sum_{n=1}^N \mu_n \xi_n$$

$$\xi = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_N \end{pmatrix} \quad \lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{pmatrix} \quad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_N \end{pmatrix}$$

Optimality condition-the optimal solution must satisfy the following conditions:

1. $\frac{\partial L}{\partial w} = \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}$
2. $\frac{\partial L}{\partial w_0} = 0$
3. $\frac{\partial L}{\partial \xi} = \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}$
4. $\lambda_n (y_n g(x_n) - 1 + \xi_n) = 0$
5. $y_n g(x_n) - 1 + \xi_n \geq 0$
6. $\mu_n \xi_n = 0$
7. $\lambda_n \geq 0, \mu_n \geq 0, \xi_n \geq 0$

Therefore, the solution will be:

$$\max_{\lambda} L(\lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j x_i^T x_j$$

$$s. t. \begin{cases} \sum_{i=1}^N \lambda_i y_i = 0 \\ C \geq \lambda_i \geq 0, \quad i = 1, \dots, N \end{cases}$$

How to find w_0

Unbounded SVs, $0 < \lambda_i < C \Rightarrow \xi_i = 0$

$$\lambda_i (y_i g(x_i) - 1 + \xi_i) = 0, i \in M$$

$$y_i g(x_i) = 1$$

$$y_i (w^T x_i + w_0) = 1$$

$$y_i \left(\sum_{k \in SVs} \lambda_k y_k x_k^T x_i + w_0 \right) = 1, \forall i \in M$$

Therefore, the solution will be:

$$w_0 = \frac{1}{N_M} \left(\sum_{i \in M} y_i - \sum_{i \in M} \sum_{k \in SVs} \lambda_k y_k x_k^T x_i \right)$$

$$N_M = |M| \text{ (number of support vectors)}$$

3.5.2.3 Kernel functions

A kernel is a function K such that:

for all $x, y \in I$

$$K(x, y) := \phi(x)^T \cdot \phi(y)$$

Where ϕ is a mapping from an input space (I) to a feature space (F). The main advantage of kernel functions when are utilized is avoiding mapping in a higher

dimensional space. It has been found that the dot product in (F) space can be computed using the dot product in (I) space. The most common kernel functions are:

- Linear Kernel: $k(x, y) := x^T y$
- Polynomial Kernel: $k(x, y) := (x^T y + d)^M, d > 0, M = 2, 3, \dots$
- Radial Basis Function (RBF): $k(x, y) := e^{-\frac{\|x-y\|^2}{2\sigma^2}}$

Kernel can be of different methods such as kernel PCA (Principle Component Analysis), and kernel FLD (Fisher Linear Discriminant). Since the dot product gives a measure of similarity between two vectors, and kernel is a method for computing similarity in the transformed space using the original input data; thus kernel function can be considered as similarity function. Mercer's Theorem ensures that the kernel function can always be expressed as the dot product between the input vectors in some higher dimensional space. The kernel matrix can be represented as follows:

$$K_{gram\ matrix} = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_N) \\ \vdots & \ddots & \vdots \\ k(x_N, x_1) & \cdots & k(x_N, x_N) \end{bmatrix}$$

Kernel matrix is a positive semi definite matrix which has nonnegative Eigen values.

Thus for any nonzero x , $x^T A x \geq 0$

Where A is said to be positive semi definite.

CHAPTER IV

METHODOLOGY

This research is intended to predict stuck pipe before its occurrence, so proper actions might be taken proactively to avoid such an incident. Machine learning techniques of artificial neural networks and support vector machines are selected in this study since they are powerful techniques in terms of prediction and they don't need strong statistical background to be utilized.

The approach used is to design models of artificial neural networks and support vector machines to predict stuck pipe probability or class label using some data from the field for training and testing. While artificial neural networks are very common in many fields including drilling engineering, there is no published work that utilizes support vector machines to predict stuck pipe. Thus a comparison approach between these two methods in terms of accuracy and error is revealed in the result section. The comparison is planned to show which method is more feasible and reliable than the other in terms of accuracy. Stuck pipe in this study is considered as the dependent variable while the drilling parameters are considered the independent variables.

4.1 Data Selection and Assembly

The quality and authenticity of the data play a major role in the reliability of the machine learning models. Since the prediction process or pattern recognition of machine learning techniques are highly dependent on the learning datasets from

previous experience with certain experiment or problem, the data have to be large enough and also high quality enough to get the model to learn properly and initiate better efficiency. Otherwise, lots of noise might be created which affect the model negatively. Sometimes the model will detect and delete the noisy dataset, if it has been sufficiently trained with high quality data.

In this project, two groups of data have been utilized; denoted as Group_A and Company_X. Data sets of Group_A were retrieved from the technical paper SPE 120128 by Murillo et al, 2009. The data selected from the paper are well designed to be implemented in stuck pipe prediction by the ANN and SVM techniques. The total number of datasets for learning in Group_A is 49 and for testing is 18. The datasets from Group_A were sorted into testing or learning randomly, but evenly distributed among the class labels. For Group_A datasets, each set of data has a target that is denoted by stuck index as DS: Differential Stuck, MS: Mechanical Stuck, or NS: None-Stuck. The number of datasets and percentage of each stuck category of Group_A datasets are shown in table 4.1.

Table 4.1. Classification of Group_A data as entered into the models

Type	# Datasets	Percentage
NS	22	33%
DS	19	29%
MS	25	38%

Alternatively, the following figure 4.1 shows the pie chart of data distribution.

Figure 4.1. Percentages of Group_A data distribution

It can be seen from table 4.1 and the pie chart of figure 4.1 that mechanical stuck (MS) datasets are the most frequent with 38%, however, the distribution is still very similar among the class labels. The datasets were further split into learning datasets and testing datasets for learning and testing processes. Table 4.2 shows the number of datasets for each class label in learning and testing of Group_A data.

Table 4.2. Number of datasets for learning and testing in Group_A data

Group_A				
Stuck Index	DS	MS	NS	Total
Learning	12	20	16	48
Testing	7	5	6	18

Company_X data were provided by certain company denoted by X, which will remain anonymous for privacy purposes. The data sets in this group are classified into S: Stuck and NS: None-Stuck. The data have been collected from different 8 fields. The distance between the fields, as well as the type, and characteristics of these 8

fields have not been provided. Each dataset in Company_X is remarked with the condition under which the sticking occurred. For example, some sticking cases occurred during connections, while others during backreaming. The following table 4.3 shows the most common conditions under which the sticking occurred for each field of Company_X data.

Table 4.3. Classification of the data in Company_X group

Field	Stuck	non-stuck	Total	%	Diff.	BR	Conn.	Tort.	Pack-off
1	2	2	4	2%	x	x	x	x	x
2	17	8	25	13%	1	1	x	x	x
3	10	4	14	7%	1	4	x	x	x
4	17	9	26	13%	1	10	1	x	x
5	3	2	5	3%	1	1	x	x	x
6	38	27	65	34%	2	2	3	x	2
7	8	16	24	12%	x	x	x	x	1
8	22	9	31	16%	1	1	1	x	2
Total	8	117	77	194					

Where:

Diff.: Differential Stuck

BR: Backreaming

Conn.: Connection

Tort.: Tortuosity

The following figure 4.2 shows the distribution of the number of datasets among the fields in Company_X.

Figure 4.2. Percentages of Company_X data distribution

It can be seen from figure 4.2 that field no. 6 has the highest share of datasets. In addition, the following table 4.4 shows the total number datasets for each stuck condition of Company_X data.

Table 4.4. Percentage of each stuck category in Company_X datasets

Stuck Case	# Datasets	Percentage
Diff.	7	6%
Increase MW	12	10%
BR	30	26%
Tight hole/Spot	19	16%
Tight while BR	10	9%
Connection	5	4%
Pack-off	5	4%
INFLUX/SI	6	5%
Misc.	23	20%
Total	117	100%

Where:

SI: Shut-in

And this same data is represented in figure 4.3 as a pie chart.

Figure 4.3. Causes of stuck in company_X Database

It can be noticed from table 4.4 and the pie chart, figure 4.3, that most of the stuck cases in Company_X data occurred during backreaming operations. Backreaming is an operation planned to condition the hole and make sure the hole is not undergaged. This operation is performed if a tight hole or tight spots are noticed. During this operation, sometimes, the filter cake that is created across the permeable zone or drainage reservoir might be removed, which in turn would cause filtrate to travel from the mud to the permeable zone and cause filter cake to accumulate across the bottom hole assembly. Under that condition, differential sticking might occur if the mud hydrostatic pressure is high enough for the pipe to become stuck across the pay or

permeable zone. It is noticed from table 4.3 that tight hole or tight spot plays a major role in causing the pipe to be stuck, based on Company_X provided datasets.

Field no. 6 has been selected to represent Compnay_X data due to the following reasons:

1. The majority of the Company_X datasets are from field no. 6 as shown in figure 4.2.
2. Since data on the physical location or characteristics of each field have not been provided; Field no. 6 has been selected to assume consistency.
3. Fewer datasets are available in the other 7 fields relative to field no. 6 as shown in figure 4.2.
4. Some fields have a large number of stuck datasets but few datasets for non-stuck cases and vice versa, which creates noise in the models and affects the results, as previously discussed.

Therefore, field no. 6 datasets were utilized in the major model runs. However, the other fields are going to be tested and utilized in the sensitive study of this research. Such sensitive study includes observing the effect of the size of the training datasets on the accuracy of the model. Also it includes studying the importance of the parameters that play roles in causing a pipe to become stuck either mechanically or differentially. The number of datasets for learning and testing of field no. 6 is shown in the following table 4.5.

Table 4.5. Number of datasets for learning and testing in Company_X data

Company_X-Field no. 6			
Stuck Index	S	NS	Total
Learning	26	9	35
Testing	12	12	24

The input parameters for each group are listed based on well directional data, mud properties, and drilling parameters. Group_A has 18 parameters. Company_X data originally contained 26 parameters; however, due to the missing values and lack of information in some parameters of Company_X data, 16 parameters representing that group were selected. The parameters that are involved in this study for both Group_A and Company_X datasets are shown in table 4.6.

Table 4.6. Parameter selection for both Group_A and Company_X data

Data Type	Input Variable		Unit	Group_A	Company_X
Well Directional Data	MD	Measured Depth	ft	X	X
	TVD	True Vertical Depth	ft	X	
	INC	Inclination	deg		
	AZI	Azimuth	deg		
	DLS	Dogleg Severity	deg/ 100 ft	X	X
Mud Properties	GEL 10M	Gel Strength	10 min	X	X
	GEL 10S	Gel Strength	10 sec	X	X
	MW	Mud Weight	lb/ ft ³	X	X
	PV	Plastic Viscosity	cp	X	X
	FV	Funnel Viscosity	sec		X
	YP	Yield Point	lb-100 ft ²	X	X
	CA	Calcium Filtrate	ppm	X	
	CL	Chloride Filtrate	ppm	X	
	WATER	Water Content	%		X
	SOLIDS	Solid Content	%		X
Drilling Parameters	CP	Circulation Pressure	psi	X	
	TQ	Torque	lb-ft	X	
	ROP	Drilling Rate	ft/hr	X	
	WOB	Weight on Bit	lb	X	
	RPM	Rotation per Minute	rpm	X	X
	GPM	Gallon per Minute	gpm	X	X
	DP-AV	Drill pipe Annular Velocity	ft/ min		X
	DC-AV	Drill Collar Annular Velocity	ft/ min		X
	CIRC	Circulation	%		X
	DG	Drag	lb		
	BS	Bit Size	in	X	X
Other	BHA	Bottomhole Assembly	ft	X	

It can be seen from table 4.6 that some parameters are available in Group_A and not in Company_X and vice versa. Some variables have been considered with certain level

of importance based on some studies conducted in the literature. However, due to the lack of the availability of the data, some important parameters were not provided with the datasets being utilized in this study.

4.2 Programming Software

This research requires programming, computational, and numerical processes. Thus, it needs certain software that can handle such research requirements. The program that has been utilized is MATLAB, an important software in engineering field. This software is selected for its convenience when it is used to build or construct ANN and SVM models.

MATLAB stands for Matrix Laboratory. This tool is designed for algorithm development, data analysis, visualization, and numerical computation. MATLAB deals mainly with matrices and is able to initiate graphics in different directions. This program is used in basic and advanced mathematics, as well as in most engineering and science fields that involve complex technical problems. It is recognized that MATLAB is powerful in terms of capabilities, power, and performance. The program is developed by MathWorks which is a corporation that specializes in mathematical computation software and was founded in 1984. The MATLAB version being utilized in this research is R2010a.

4.3 Artificial Neural Network (ANN) Model Construction

4.3.1 Developed Models

The constructed ANN model for this research is based on the Multilayer Perceptrons, which is discussed in the previous chapter. Generally this model consists of three layers: Input, Hidden, and Output. The input layer represents the inputs of the problem. Therefore, the number of neurons in this layer is the same as the number of parameters of the problem. The hidden layer is the one where the computational process occurs. Depending on the complexity of the problem, the model might have more than one hidden layer. The output layer represents the output or the class label of the problem. Table 4.7 shows the number of nodes or neurons for each layer of the constructed ANN model.

Table 4.7. Number of nodes or neurons for each layer of the constructed ANN models

# of neurons for each layer		
	Group_A	Company_X
Input	18	16
Hidden	19	17
Output	1	1

It can be seen from table 4.7 that ANN model for Group_A contains 18 neurons for the input layer since the number of parameters provided for that group is 18. However, based on the optimization process, it has been found that the number of neurons in the hidden layer in Group_A model is the number of inputs plus one which is 19 in this case. Since the model will produce one class label for each dataset (either stuck or non-stuck), the output layer is designed to have a single neuron. After a selection process, Company_X datasets are provided with 16 parameters, therefore, the number of neurons in the input layer for this database is 16. Like Group_A datasets, the hidden

layer of Company_X is designed with 17 neurons; one plus the number of inputs.

Similar to Group_A, the output layer of Company_X is developed to contain one neuron which produces one class label, either stuck or non-stuck.

For both groups of data, the most common two activation functions in artificial neural networks have been implemented; Sigmoid and Tanh. Sigmoid function is represented by:

$$F(x) = \frac{2}{1 + e^{-x}}$$

Where it can be derived as follows:

$$\frac{dF}{dx} = \frac{1}{2} (1 - F)(1 + F)$$

The sigmoid function has been implemented for transformation in hidden and output layers. It will in turn map each neuron in the layer between -1 and 1. The Tanh function does the same in terms of mapping the neurons between -1 and 1; and is represented by:

$$F(x) = \tanh(X)$$

Where it can be derived as follows:

$$\frac{dF}{dx} = \text{sech}^2(X)$$

Therefore, four scenarios in the constructed ANN models are going to be implemented in this project. Each scenario is denoted by F followed by a number that represents its sequence as shown in table 4.8.

Table 4.8. Scenarios of the selected activation functions for the constructed ANNs' models

Scenario	Hidden Layer	Output Layer
F1	Sigmoid	Sigmoid
F2	Tanh	Sigmoid
F3	Tanh	Tanh
F4	Sigmoid	Tanh

The constructed models are able to measure the probability of getting stuck, which is vital especially when it comes to real-time situations. The algorithm designed to construct the model is as follows:

1. Initialize V and W
2. Training stage
 - 2.1 Feed forward phase
 - 2.2 Backpropagation
 - 2.2.1 Compute the error at the output unit
 - 2.2.2 Compute the error at the hidden unit
 - 2.2.3 Update weights

$$W = W + \Delta W$$

$$V = V + \Delta V$$
3. Testing stage

The developed models have been designed with two major phases; feed forward phase and backpropagation phase. Initially both weight matrices V and W are loaded up with random values between -1 and 1; this is fixed for all constructed ANN models to make them all start from the same point. Otherwise, when the matrices V and W are updated, the output values might alter and thus the error would change. Next, the training process is initiated, starting with the feed forward phase. In this phase, the

predicted Y value, which is the predicted stuck index, is calculated. Then, the program proceeds with the backpropagation phase. In that phase, the error at the output unit is computed and ΔW can be calculated accordingly. Next, the error at the hidden unit is computed, and thus ΔV can be calculated as well. Thereafter, both W and V can be updated. In addition, the program has been designed with certain number of epochs to achieve the least error; therefore it would produce better estimation.

4.3.2 ANN MATLAB Built-in Models

The MATLAB software is equipped with a built-in neural network model that can classify the data into two or more classes. This model is optimized in such a way that can produce as accurate results as possible. This program is not designed to produce or measure probabilities which are considered key in stuck pipe prediction, especially in real-time cases. This proved to be a limitation in this research. The main reason for applying MATLAB neural networks in this research is to compare the results produced by the models developed in this study, and also to find how feasible these models are.

Basically, the built-in neural network model in MATLAB is developed by utilizing the following codes:

- i. $T = \text{ind2vec}(T_c)$
- ii. $\text{net} = \text{newpnn}(P, T)$
- iii. $Y = \text{sim}(\text{net}, X)$
- iv. $Y_c = \text{vec2ind}(Y)$
- v. $Y_2 = \text{sim}(\text{net}, X_2)$

vi. `output= vec2ind(Y2)`

Where:

Tc: True or actual class labels for training datasets

X: Matrix of training datasets

X2: Matrix of testing datasets

The codes are described in table 4.9.

Table 4.9. MATLAB codes utilized to build MATLAB built-in model of ANNs

MATLAB Code	Description
ind2vec	Convert the indices of class labels into vectors
vec2ind	Convert vectors into indices of class labels
newpnn	Design probabilistic neural network
sim	Simulate neural networks

4.4 Support Vector Machines (SVM) Model Construction

4.4.1 Developed Models

Two machine learning models are developed in this research based on the most common two kernel functions. The first one is based on the Linear Kernel function and the second one is based on the Radial Basis Function (RBF). The models are able to classify the data into two classes, either stuck or non-stuck. The algorithms of these models were written utilizing MATLAB language.

- i. Linear and RBS Kernel Functions

$$k(x_i, x_j) := x_i^T x_j$$

$$k(x_i, x_j) := e^{(-\frac{\|x_i - x_j\|^2}{2\sigma^2})}$$

- ii. Quadprog formulation
- iii. Calculating w

- iv. Calculating w_0
- v. Testing stage

Quadprog is a quadratic programming function in MATLAB. This function is designed to optimize the values of certain parameters to solve a specific problem. In this project this function is utilized to optimize the values of alpha that can be used to calculate w and w_0 . The developed models are designed to utilize both linear kernel and RBF. Thus there is one array of w which represents the parameters and one single value of w_0 for each developed model at each run.

Support vector machines are based on this equation:

$$g(x) = x_i^T w + w_0$$

Where x represents the matrix of inputs. Once the model is designed to find w and w_0 with the optimization process, the testing samples can be fed to the equation and the model can predict the outputs accordingly. w and w_0 can be found based on the following equations:

$$w = \sum_{i=1}^N \lambda_i y_i x_i^T$$

$$w_0 = \frac{(\sum_{i \in M} y_i - \sum_{i \in S} \sum_{j=1}^N \lambda_j y_j x_j^T x_i)}{N_S}$$

These two equations can be solved in MATLAB utilizing the quadprog function based on the optimization process.

Each model of support vector machines has one or two parameters that should be optimized to produce the best results possible. With regard to the model with linear kernel function, the parameter that can be changed or altered to optimize the model is

C value. It is called the penalty factor that trades off the complexity and it is derived from:

$$\min \text{norm} \frac{w^2}{2} + C \sum \xi$$

If the problem for a study has two dimensions, C value effect can be observed in terms of boundary, margins and support vectors. Depending on its value, C can cause the margins to become wider or narrower and also might cover or include more vectors. In the case of the Radial Basis Function model, two parameters can be altered to optimize the model. In the case of a 2 dimensional problem, these parameters are C and sigma (or sometimes called σ^2) that have a certain effect on the margins, decision boundaries, and support vectors. In the case of multidimensional problems that include many parameters, the effect of these parameters can be noticed or observed in the results or output values. Therefore, the developed models have to be optimized with trial and error to produce the proper C values, in case of linear kernel, C and sigma, in case of RBF, which in turn could create the proper pattern that is strong enough to predict outputs with high accuracy.

4.4.2 SVM MATLAB Built-in Models

For comparison and qualification purposes, the SVM built-in function in MATLAB was utilized in this research. The SVM MATLAB built-in function is equipped and designed to classify the data into 2-classes only. The model is developed based on the following algorithm:

- i. `svmStruct = svmtrain(X,T)`
- ii. `classify=svmclassify(svmStruct,X2)`

Where:

T: True or actual class labels for training datasets

X: Matrix of training datasets

X2: Matrix of testing datasets

The programming codes are described in table 4.10.

Table 4.10. MATLAB codes utilized to build MATLAB built-in model of SVMs

MATLAB Code	Description
svmtrain	Train MATLAB SVM model
svmclassify	Classify new data based on SVM model simulated by svmtrain

4.4.3 SVM MATLAB Interface Models

The SVM MATLAB built-in function is not able to classify the data into three classes, which is needed because the datasets of Group_A are provided with three class labels. Therefore, MATLAB interface codes from the Library of Support Vector Machine (LIBSVM) were loaded into MATLAB, and thus used to classify the data in Group_A into three classes. The Library of Support Vector Machines or LIBSVM was created by Chih-Chung Chang and Chih-Jen Lin from Taiwan University. The program was designed to make it easier for non-specialists in SVMs to handle or predict class labels for complicated or high dimensional problems in different fields.

MATLAB interface from LIBSVM is equipped with two run scenarios; one for linear kernel model and the other one for pre-computed kernel model. Therefore, the program is going to predict based on these two models. The algorithm for LIBSVM was created as follows:

- i. `model_linear = svmtrain(T, X)`
- ii. `[predict_label_L] = svmpredict(T2, X2, model_linear)`

- iii. `model_precomputed = svmtrain(T, X);`
- iv. `[predict_label_P] = svmpredict(T2, X2, model_precomputed)`

Where:

T: True or actual class labels for training datasets

T2: Actual class labels for testing samples

X: Matrix of training datasets

X2: Matrix of testing datasets

The model codes are described in table 4.11.

Table 4.11. MATLAB codes for the SVM model from Library of Support Vector Machines

MATLAB Code	Description
<code>svmtrain</code>	Train MATLAB interface of LIBSVM Model
<code>svmpredict</code>	Predict new data based on LIBSVM model simulated by <code>svmtrain</code>

Therefore, the scenarios being run utilizing SVM technique are shown in the following figure 4.4.

Figure 4.4. Design scenarios of SVM models

4.5 Accuracy and Error Measures

The input data for stuck pipe are within different ranges; for example, depths are described with thousands of feet while doglegs are described in single digit values or decimal. Therefore, the inputs have to be normalized such a way that the system can read consistent ranges and compute properly. Thus, a normalization function has been applied for all models of this research to normalize the input values between 0 and 1.

The normalization function utilized is shown as follows:

$$X_i = \frac{(X_i - X_{min})}{(X_{max} - X_{min})}$$

Where X_i is the normalized X value at point i. There is tremendous improvement in the prediction process when normalizing the data than when normalization has not been initiated. Therefore, this step is vital and important in designing machine learning models for both ANNs and SVMs.

The accuracy for each scenario or each model is based on how many outputs are correctly classified out of all testing samples. Therefore, the accuracy would be calculated as follows:

$$Accuracy = \frac{\text{Correctly Classified Testing Samples}}{\text{Total Number of Testing Samples}} \times 100$$

The error function that has been utilized to measure the error of the results for all models of this project is MSE: Mean Square Error, which has the following formula:

$$MSE = \sum_{j=1} \frac{(Y - T)^2}{N}$$

Where:

Y: Predicted stuck index using machine learning methods

T: Actual stuck index

N: Number of outputs

4.6 Data Analysis

This study is based on utilizing two machine learning techniques to predict one of the most critical and severe problems in drilling industry, stuck pipe. Each of the two methods has a certain level of accuracy depending on the way the models were designed and the type of data that were collected. Therefore, the results are compared based on the accuracy and the mean square error. The comparison is based on a specified number of testing samples for Group_A and Company_X. As specified earlier, Group_A has 18 datasets for the testing process while Company_X Field no. 6 has 24 testing samples. The testing datasets or samples are the same for all designed models. The results are presented in such a way that compares the models based on their categories which are constructed models, MATLAB built-in functions, and MATLAB interface. The criteria for comparison and the accuracy measure for each model is how many correctly classified testing samples each model predicts out of the total testing samples.

A sensitive study is also intended in this research. This study includes measuring the level of importance of drilling parameters with respect to differential and mechanical sticking. In addition, the sensitive study also includes finding or observing the relationship between the size of learning datasets and accuracy of prediction. Furthermore, it measures the reliability of the models that utilized learning data from one field and tests the model with data from different fields.

There are two scenarios with regard to the sensitive study that takes into account measuring the relative importance of drilling parameters with respect to differential and mechanical sticking. The first approach is through MATLAB built-in functions of ANNs and SVMs, and the second approach is through the constructed SVM model. The idea behind utilizing MATLAB built-in functions of ANNs and SVMs is running the system or program in an absence of each independent variable at a time. The model run with relatively lower accuracy indicates the importance of that absent parameter. The second study measuring the importance of the drilling parameters with respect to the prediction process is through the constructed SVM models. In this case, the developed models of SVMs are able to produce the weighted factors which represent the independent variables. The weighted factors of these models are taken into account: in the case of linear kernel, optimum C values were used, in the case of RBF model, both optimum C and sigma values were used. This sensitive study has been applied in both groups of data.

In addition, another sensitive study is designed in such a way as to find the effect of the size of learning datasets with respect to the accuracy of prediction. This has been done by utilizing both groups of datasets specified for learning, and reducing the number of training datasets in sequence. Both MATLAB built-in functions of ANNs and SVMs were applied in this stage and the average accuracy was considered.

This study considers the case that takes into account Field no. 6 data of Company_X as training sets and from other fields of Company_X as testing samples. The reason for such a study is to find how reliable the model is when it is applied in a different field.

CHAPTER V

RESULTS

5.1 Accuracy of the Models

In this study, each group of data has been specified with a number of testing samples. In the case of Group_A, the number of testing samples is 18 while it is 24 for Company_X data. The testing datasets for Group_A include all cases of stuck: mechanical, differential and non-stuck cases. The 24 testing datasets which represent Company_X data are from Field no. 6. However, the other fields' datasets are included in the sensitive study of this research. The testing samples are utilized in all models of this study to find out how feasible each model is in terms of accuracy. The designed models can be classified as constructed, MATLAB built-in, and MATLAB interface. The constructed models are those developed with different scenarios by programming the algorithms of both ANNs and SVMs in MATLAB programming language. While Developed ANN models include four scenarios of selected activation functions, SVM developed models include two kernel functions. In contrast, MATLAB built-in functions are those models of ANNs and SVMs that are already built into the MATLAB program. The MATLAB interface is the model from the Library of Support Vector Machine which has the ability to classify the data into 3 classes by utilizing the SVM technique. This MATLAB interface is already designed to predict with two models; linear and pre-computed.

In the following figures 5.1 – 5.4 are four bar charts that show the accuracy represented as a percentage for each model utilized in this study. The bars indicated

with green color are for those constructed models of both ANNs and SVMs. The bars with blue color are for those MATLAB built-in functions of ANNs and SVMs, while the violet colored bars represent the MATLAB interface of SVMs for both linear and pre-computed models. In case of the 3-group classification, the bars filled with red color indicate the models utilized to classify the testing samples of Group_A into 3 classes.

Figure 5.1. Accuracy of the models when applied in all datasets of Group_A

Figure 5.1 shows the results for all testing samples of Group_A, which consists of both mechanically and differentially classified datasets. It can be seen that among all the constructed models, both developed SVM models generate the best predictions with no misclassified testing dataset (100% accuracy). The Tanh-Sigmoid scenario of ANNs has the highest accuracy at 72%, compared with the other developed ANN models. That might be due to the type of data used for training and testing. With regards to MATLAB built-in functions, it is obvious that both models of ANNs and SVMs produce 100% of correctly classified outputs. The Sigmoid-Tanh scenario, at 72% accuracy, differs from the ANN model built into MATLAB, at 100%. That might be due to the way the two programs were designed since the one built into MATLAB is developed in such a way that optimizes the number of hidden units, number of hidden layers, and activation functions. Although the MATLAB built-in models are producing the most promising results in this case, the model is limited to be adjusted whereas the constructed model can be modified to improve the prediction process. Regarding MATLAB interface models of SVMs, it is obvious that the linear model is better than the pre-computed one in terms of prediction accuracy. It can be seen that when classifying the Group_A testing data into 3 classes, the MATLAB built-in model of ANNs is able to classify the data into 3 groups with 94% of accuracy, which indicates better prediction of stuck pipe than SVM linear model by LIBSVM.

The results of the models that utilize only the differential stuck datasets of Group_A for training and testing are shown in figure 5.2.

Figure 5.2. Accuracy of the models when applied in Group_A differential stuck datasets

In figure 5.2, which compares the constructed models, it can be seen that the developed linear SVM model is the most accurate one among the other developed models, 85% of accuracy for SVM_Linear model. It can also be noticed that the constructed ANN models are relatively better when utilizing only differential stuck

datasets than when utilizing all datasets of Group_A. In the case of the built-in models, both ANNs and SVMs are predicting with the same accuracy as in the previous case. In terms of MATLAB interface of LIBSVM, the linear model is more accurate than the pre-computed one when classifying differential stuck data of Group_A, 85% of accuracy for LIBSVM_Linear and 46% for LIBSVM_Pre-computed.

Figure 5.3. Accuracy of the models when applied in Group_A mechanical stuck datasets

Figure 5.3 shows the results of all the models' predictions when the stuck data from Group_A were utilized. It can be noticed that both developed models of SVMs are generating the most accurate outputs among the other constructed models, 91% for both SVM_Linear and SVM_RBF models. Unlike the previous two cases, all scenarios of developed ANN models in this case are not producing promising results in terms of accuracy. That might be due to the nature of the data. However, as in the previous cases when the linear model of LIBSVM and the pre-computed model of MATLAB interface of SVMs utilize only the mechanical stuck data of Group_A, the linear model of LIBSVM is more accurate.

The outputs and the accuracies of all designed models for both groups of data, Group_A and Company_X, and the optimization process for the developed SVM linear and RBF models are shown in Tables A.1 – A.7 in the appendix.

Figure 5.4. Accuracy of the models when applied in Company_X-Field no. 6

Figure 5.4 illustrates the accuracy of each model in this study, when stuck pipe of Company_X Field no. 6 was predicted. It can be seen that the developed linear model of SVMs predicts stuck pipe with reasonable accuracy of 83%. However, RBF model in this case is not as accurate as the previous cases of Group_A data, 50% of accuracy for SVM_RBF when applied in Company_X Field no. 6. It can be shown that

Sigmoid-Sigmoid scenario of ANNs is the most accurate one with a reasonable 71% of accuracy out of the other 3 scenarios of the constructed ANN models. However, MATLAB built-in functions of SVMs and ANNs are quite similar in terms of correctly classified testing samples, as in the previous cases. In case of SVM MATLAB interface models, it is shown that the linear model is more accurate than the pre-computed model, again, as in the previous results of Group_A. It can be noticed that the developed SVM model in this case is more accurate than both MATLAB built-in SVMs and linear model of LIBSVM. That indicates that the user is limited with the built-in models since the program is designed in such a way the optimization is limited, therefore the predictive capabilities are limited. As a result, these programs might be limited with the type of data that can be used to develop an accurate model. The constructed models are more flexible and can be modified and adjusted to predict more efficiently.

5.2 Sensitive Study

This research includes a study that is intended to observe the sensitivity of drilling parameters toward the tendency of stuck pipe. In addition, another study is also designed to attempt to determine the sensitivity of the size of training datasets with respect to the accuracy of the models.

5.2.1 Sensitivity of the Independent Variables

Stuck pipe is a problem that is affected by many independent parameters. Some of these variables are more significant than others. Depending on the type of

sticking, each parameter has a certain effect on the tendency of the pipe to be stuck. For example, in the case of differential sticking, the differential pressure plays a major role in causing the pipe to be differentially stuck across a permeable zone. However, some of the other parameters may have direct or sometimes indirect effects in causing that kind of stuck pipe. Since mechanical sticking occurs for many reasons, some independent parameters may be significant in some cases and not in other cases. Also, the characteristics of the field may have some factors that would alter the significance of some variables.

In this study, two approaches have been designed to fairly analyze the effect of some drilling parameters on causing the pipe to be stuck. This analysis is based on the type of the data provided for both group of databases. The first approach is based on deleting each parameter one at a time and running the MATLAB built-in functions of ANNs and SVMs. The second approach takes into account that the constructed SVM models have the ability to calculate the weighted factor of each parameter in terms of relative values.

5.2.1.1 Utilizing MATLAB_SVM_ANN

The following four charts, figures 5.5 – 5.8, show the results of this study when both MATLAB built-in models of SVMs and ANNs are run by deleting one parameter at time. This approach is to measure the sensitivity or the relative importance of each independent variable of stuck pipe for both groups of data. In case of Group_A, 3 scenarios were performed: all stuck cases, mechanically stuck cases, and differentially stuck cases. In the case of Company_X, this sensitive approach was

applied on Field no. 6 datasets. The bar charts represent the average accuracy of both SVM and ANN MATLAB built-in models in percentage.

This kind of study might have the highest level of accuracy in terms of classifying which independent variable is more significant to cause stuck pipe than the other. However, it gives an idea of how the models are trained and also how they predict in the absence of each parameter at a time.

Figure 5.5. Average accuracy of MATLAB_SVM_ANN in an absence of each parameter for all datasets of Group_A when classifying the data into 2 groups

From figure 5.5, it can be seen that the average accuracy of both ANNs and SVMs is relatively lower when these models are run in the case of an absence of the following parameters: circulating pressure, yield point, gel 10s, both gel 10s/gel 10m, CL concentration, dogleg severity, rate of penetration, length of bottom hole assembly, or weight on bit. This observation might be considered as an indication of the relative influence of each of the absent parameters on stuck pipe in Group_A datasets. However, the accuracy of measuring the significance of the parameters in stuck pipe occurrence might be more reasonable when the training datasets are of sufficient big size.

Fig. 5.6. Average accuracy of MATLAB_SVM_ANN model in an absence of each parameter for the mechanical stuck datasets of Group_A

Figure 5.6 shows the average accuracy of MATLAB built-in models of both ANNs and SVMs in case of deleting one independent variable at time for mechanically stuck datasets of Group_A. As it can be seen, the average accuracy of the models is relatively lower when omitting the column of data for: circulating pressure, torque, dogleg severity, rotation per minute, length of bottom hole assembly, or weight on bit. This might be an indication of how each of these parameters is important in terms of

the prediction process and also in terms of the increasing tendency of the pipe to be mechanically stuck.

Figure 5.7. Average accuracy of MATLAB_SVM_ANN model in an absence of each parameter for the differential stuck datasets of Group_A

In the case of differentially stuck cases of Group_A datasets, it can be noticed from the bar chart of figure 5.7 that in the absence of depth, plastic viscosity, yield point, gel 10s, gel 10m, CL concentration, or length of bottom hole assembly, the average accuracy of SVM and ANN MATLAB built-in functions is relatively reduced. This reduction might be due to the significance of these independent variables in terms of

leading the pipe to be differentially stuck. It is interesting to note that most of these parameters are related to drilling mud rheological properties which play a major role in increasing the tendency of the pipes to be differentially stuck.

5.2.1.2 Utilizing Constructed SVM Linear Model

The developed model of SVMs is able to measure the weighted factors of the parameters provided for the problem of study. In this research, stuck pipe data are provided with independent variables which can be represented by the weighted factors measured by the constructed models of SVMs. Both linear and RBF models of SVMs are capable of calculating the weighted factors for the developed SVMs.

Therefore, the following bar charts show the absolute value of the weighted factors for both Group_A and Company_X calculated by the constructed SVM linear model. The absolute values are considered in this approach instead of the negative values, which might indicate a negative correlation with stuck pipe in this content. In all the following cases the linear kernel scenario is considered since it predicted stuck pipe testing samples with the highest accuracy over the RBF model for all the testing data of both groups. For all cases the weighted factors were measured at the optimum value of C_constant which is the only parameter that affects the output of the SVM linear model.

Figure 5.8. Weighted factor of each drilling parameter by SVM_Linear model for all datasets of Group_A

Figure 5.8 shows the calculated weighted factors by the developed SVM linear model for all datasets in Group_A. The values of the weighted factors for each parameter are arranged in descending order. It can be seen that circulating pressure, weight on bit, plastic viscosity, and also yield point are considered as the highest weighted factors. In

contrast, calcium concentration, dogleg severity, and true vertical depth are considered as the lowest weights respectively. This observation might be considered as reflection of the relative importance of some parameters provided for Group_A datasets. This kind of measurement is based on the way that the constructed SVM linear model calculated the weighted factors.

Figure 5.9. Weighted Factor of each drilling parameter by SVM_Linear model for the mechanical datasets of Group_A

In case of mechanical stuck datasets of Group_A, it can be seen from figure 5.9 that circulating pressure, rotation per minute, weight on bit and plastic viscosity appear to be the highest weighted factors among the other provided parameters. On the other hand, pumping rate, dogleg severity, and measured depth are considered the lowest weighted factors.

Figure 5.10. Weighted factor of each drilling parameter by SVM_Linear model for the differential stuck datasets of Group_A

In the case of differentially stuck datasets of Group_A, it can be seen from figure 5.10 that chloride concentration, rotation per minute, weight on bit, and circulation pressure

represent the highest weighed factors calculated by the developed linear SVM model.

In contrast, gel 10m, bit size, and mud weight are considered to be the lowest weighted factors.

Figure 5.11. Weighted factor of each drilling parameter by SVM_Linear model for Company_X field no. 6 datasets

Figure 5.11 represents the weighted factors of the parameters provided for Company_X Field no. 6 datasets. As it can be seen from this chart, dogleg severity,

mud weight, plastic viscosity, drillpipe annular velocity, and pumping rate appear to be the highest weighted factors measured by the SVM_linear constructed model. In contrast, gel 10m, water fraction, and gel 10s are considered with the least weighted factors in Company_X database.

5.2.2 Sensitivity of the Size of Training Samples

The following plots, figure 5.12 - figure 5.17, show the results of a sensitive study when it is applied on both Group_A and Company_X datasets. The purpose of this study is to observe the relationship between the size of the training datasets and the accuracy of the model. Also, this study includes observing the effect of distribution of class labels in training with respect to the accuracy of prediction. In addition, it shows how efficient the model is when it is trained by datasets from different fields and applied in one field and vice versa.

5.2.2.1 Group_A Data

This group of data was selected for this sensitive study which is based on observing how the size of the training datasets affects the accuracy of the model since they follow higher consistency than Company_X datasets. This observation is made based on the relatively high accuracies measured for Group_A data by the models of this research over Company_X data. The following figure 5.12 shows the effect of reducing the number of training datasets and the accuracy of prediction for fixed number of 18 testing samples in Group_A database.

Figure 5.12. Number of learning datasets vs. average accuracy of MATLAB_SVM_ANN model

It can be observed from figure 5.12 that the accuracy is constant when stuck cases are almost double the non-stuck cases, to the point where both stuck and non-stuck fractions are equal. This high and constant accuracy might be due to the fact that the number of stuck cases in testing samples is more than the number of non-stuck cases. It can also be seen that when the total number of training datasets are reduced from 30 to 18, the accuracy drops to almost less than 90%. However, there is an increase in the average accuracy when the learning datasets were reduced from 18 to 10. This

anomalous behavior might be attributed to the strong relationship or similarities between some of the testing samples and the training datasets in those cases.

5.2.2.2 Company_X Data

The second group of data, Company_X datasets, in this study is selected to observe the effect of the training datasets on the accuracy of the MATLAB_ANN_SVM. In addition, since this database consists of data from different fields, it is vital to notice the accuracy of the models when they are trained by data from one field and applied or tested in a different field. The accuracy of the models in this study may be affected or dropped since each field has certain characteristics that may differ from the other field. The differences between the fields might be formed in the geology of each field or the proper type and properties of drilling fluid that each field is drilled with.

Each of the following graphs, figure 5.13a - 5.17, consists of a primary and a secondary y-axis. The primary y-axis reflects the number of datasets while the secondary y-axis represents the accuracy of the models which are ANN and SVM MATLAB built-in functions. In contrast, the x-axis represents the run number of each scenario. Each bar or column is divided into four fractions. Each portion is filled with certain color. The blue and red filled parts represent the number of training stuck and non-stuck datasets respectively; the green and violet filled portions represent the number of testing stuck and non-stuck samples respectively. The accuracies of the models are represented in figure 5.13a – 5.17.

Figure 5.13a. Sensitivity of the size of the training data with respect to accuracy in Field_6 of Company_X

Figure 5.13a shows the results when only Field no. 6 of Company_X datasets were applied. It can be seen that the highest accuracy occurred when the fractions of stuck and non-stuck datasets for training were equal. This relatively higher accuracy might be due to the fact that the system is already trained evenly with both cases, thus it may predict more efficiently.

Figure 5.13b. Sensitivity of the size of the training data with respect to accuracy in Field_6 of Company_X with more run scenarios

Figure 5.13b shows all cases when Field no. 6 datasets were applied in both training and testing phases. It can be noticed once again that run no. 5 produced the relatively highest accuracy among the other scenarios. That might be due to the equal distribution of stuck and non-stuck cases in the training stage. However, run no. 3 has equal number of datasets for both stuck and non-stuck, but the accuracy is slightly lower than run no. 5. This reduction in accuracy might be because of the size of

training datasets (34) for run no. 3 which is less than the size of the training samples (24) for run no. 5.

Figure 5.14. Sensitivity of the size of training data in different fields (2, 4, 6, 7, and 8) of Company_X

Figure 5.14 shows the cases when the Fields 2, 4, 6, 7, and 8 of Company_X data were applied for training and testing phases. It can be noticed that accuracy of run no. 8 is higher than that of run no. 7. This increase might be explained by the distribution of training datasets for both stuck and non-stuck cases, which is equal in run no. 8, while the number of stuck training datasets in run no. 7 is higher by 20 points than non-stuck

cases in the same run. The accuracy of run no. 8 is higher even though the number of testing samples in this run is more than that in run no. 7. These two observations may support the indication that the models utilizing machine learning techniques need to be equally trained with all classes, such as stuck and non-stuck in case of stuck pipe.

Figure 5.15. Field effect of Company_X data in the accuracy of SVM and ANN models

Figure 5.15 shows the effect when more data from Field no. 8 are combined in sequence with field no. 6 data for training and testing stages. It can be seen that when more training data from Field no. 8 are added into the model and the testing samples

are only for those of Field no. 6, the accuracy drops slightly. That can be an indication that the prediction could be more robust when the training and testing data are from the same field. Otherwise, more data included in training from different fields might cause noise to the system.

Figure 5.16. Applying datasets of Field_6 on same field and different fields Datasets of Company_X

Figure 5.16 shows the scenarios when the data from Field no. 6 are applied for training stage and data from different fields are applied for testing phase, except for the last run no. 12, which includes more data from field no. 8 for training phase. It can be seen that

the scenario with the highest accuracy is for run no. 17. Also, it can be noticed that the accuracy in this run is more than 90% which is quite relatively high. That reasonably good prediction might be due to the strong relationship between the following two cases: case 1, which is the stuck data of testing in Fields no. 2, 3, and 4 and non-stuck data for testing in Fields no. 7 and 8, and case 2, which is training data of Field no. 6.

Figure 5.17. Sensitivity of the size of the training data when applied in the same mechanical stuck category (Packoff)

Figure 5.17 shows the cases of hole pack-off when applied in training and testing phases for most of the fields in Company_X. These scenarios are run in this study to observe the effect of the size of training datasets with respect to the accuracy in case of the same sticking category for different fields. It can be noticed that accuracy is higher in the cases that have more training datasets than those that have relatively fewer datasets. However, the number of testing samples for those with less accuracy is more than those with higher accuracy. That increase in the number of testing samples might be a factor which reduced the accuracy.

The summary of all the runs done on the different fields of Company_X datasets by MATLAB_ANN and MATLAB_SVM models to observe the effect of the size of the data on the accuracy of the models is shown in Table A.8 in the appendix.

CHAPTER VI

DISCUSSION

From the results of this study, it can be seen that constructed SVM linear model is the most accurate one among the other constructed models for both groups of datasets. That is why the constructed linear kernel model of SVMs was utilized to measure the relative importance of the parameters with respect to prediction process rather than RBF model. In addition, the results of this study have shown that both developed models of SVMs, linear and RBF, achieved higher accuracy than the constructed ANN scenarios in terms of stuck pipe prediction. This precedence of SVMs over ANNs in stuck pipe prediction might be due the nature of the data that were utilized, and also the way the constructed models were designed. Otherwise, in some cases or some other fields of study, ANNs may produce better accuracy than SVMs. It can also be noticed that both SVM developed models have achieved relatively higher accuracy than the other scenarios of ANN models. However, both MATLAB built-in functions of ANNs and SVMs have shown quite a good match in all cases. This variation between the constructed ANN models and ANN MATLAB built-in functions might be due to the way the ANN MATLAB built-in function was designed since it might be designed with better optimization than the constructed ANN models.

One of the important observations in this study is that the results of SVM linear constructed model were almost the same as those for SVM linear model from the Library of Support Vector Machine; which is an indication that the constructed

models of SVMs were properly designed. However, there is slight deviation between the results of the constructed SVM linear model and the linear one from LIBSVM in case of Company_X Field no. 6 datasets. This variation might be due to the C_value that was specified for the constructed model and the C_value that was specified in the linear model from LIBSVM.

It can also be noticed that the pre-computed model of SVMs from LIBSVM always produces lower accuracy than the linear model of LIBSVM. That might be because of the way the program was designed since it even shows the same class labels for the testing samples for all the cases of both databases utilized in this study. These unreasonable results might be due the way the program was developed as the stuck pipe data utilized may not be compatible with the program.

It can be seen from the results of this research that the prediction accuracies of the models in case of Group_A are higher than those for Company_X datasets. That can be an indication that Company_X datasets have fewer consistencies than Group_A datasets.

The constructed ANN models have to be developed and optimized with several scenarios of activation functions, since in each case during this study different scenarios achieved the highest accuracy of prediction. Therefore, the program has to be designed and optimized in such a way that it can produce the best prediction by combining the most optimum activation functions.

In case of the sensitive study, which reflects the relative importance of the drilling parameters with respect to the stuck pipe occurrence, it can be seen that the study utilizing the weighted factor is more representative than that one utilizing the

MATLAB built-in functions of SVMs and ANNs. In the latter approach, the program was run by deleting one parameter at a time. This test might not be that accurate since the program is based on a learning process and it might give higher accuracy even in the case of disregarding one parameter. The reason for that higher accuracy might be that since the program found good match or similarities between the training and testing samples, it would predict correctly even though one parameter was omitted during the run. Also it is important to bring to attention that the weighted factors measured by the second approach of SVM linear model are reflective of the relative importance of the independent variables utilized in the study with respect to the prediction process. For example, if the magnitude of the weighted factor of one parameter is relatively higher than the others, this would mean that if that parameter is not given the correct value in a certain testing sample, it would switch the correctly classified label into an incorrectly classified one. That is why if the parameter has higher magnitude it would be very sensitive to the model.

The sensitivity study implemented in this study in order to reflect the importance of the size of the training samples might not be highly representative. This may be attributed to the data from Company_X that might not possess a high level of consistency and are not sufficiently large. Thus, the sensitivity study using Company_X data might not be typical since it appeared to have some anomalous trends.

CHAPTER VII

RECOMMENDATIONS

Machine learning methods involve two significant components that would affect the prediction process. The first one is the data and the second one is the developed models.

7.1 Recommendations for Data Collection

This research is based on the machine learning techniques, Artificial Neural Networks and Support Vector Machines. The prediction process of these methods is highly affected by the type of data utilized. Thus, if the data does not possess high level of authenticity, the accuracy of the prediction would not be reasonable. In the case of stuck pipe prediction study, there are a number of areas that have to be taken carefully in consideration in order to predict properly; specifically, the class labels or type of sticking, data distribution among each type, size of the datasets, and the number of fields to be applied in the study.

In terms of the type of sticking that would be utilized in this study, it is recommended to limit the data to differential sticking, and exclude the mechanical sticking. That can be justified since mechanical sticking during drilling could occur due to different causes that can be affected by different drilling parameters. Since different parameters might be involved in each case of mechanical sticking, the learning process would be complicated and lack consistency. Differential sticking would occur due to some common causes with limited parameters that might be due to

high mud weight or as a result of improper rheological properties of drilling fluid, especially fluid loss control. Specifying the common causes of differential sticking would be definitely easier and clearer than mechanical sticking. Therefore, machine learning process by ANNs and SVMs from differential stuck cases would be more consistent than mechanical sticking cases. However, that limitation of mechanical sticking is not an indication that machine learning techniques are not able to predict mechanical sticking reasonably. Instead, if the causes of mechanical sticking are limited or restricted to a number of common causes among the field for study, prediction would be more reasonable than the case when the causes are considered many.

Another recommendation in this study is the distribution of the stuck and non-stuck data in the learning datasets. This distribution needs to be even among each class label to enhance the efficiency of the prediction process. If there are more stuck cases than non-stuck cases represented in the learning data sets, the training process might not be as sufficient for non-stuck cases. This is because the testing process relies on the learning process. Consequently, the prediction efficiency may not be reasonable. This improper prediction might be due to the reason that the system or the model is trained well with many stuck pipe cases but not with non-stuck cases. Therefore, both class labels have to be trained evenly in the system with the same or close number of datasets.

Since stuck pipe problem involves many parameters that belong to different categories. The size of training datasets has to be large enough; thus the model would

be familiar with most of the cases that might cause the pipe to be differentially or mechanically stuck.

As each oil or gas field has its own characteristics in terms of geological properties or drilling parameters, especially the drilling fluid properties, it is not recommended based on this study to have more than one field with different characteristics to be applied in machine learning process. For example, if Field_A is drilled with oil based mud and Field_B is drilled with water based mud, each field has totally different figures of drilling fluid properties. This variation in the mud properties would affect the training process in such a way that the prediction would not be reasonable. Therefore, it is recommended to apply ANNs and SVMs on one field if the fields' characteristics obviously vary. Otherwise, artificial intelligence might be applied in different fields if the characteristics are close or fairly similar.

7.2 Recommendation for the Model Construction

In this study different areas have to be improved to optimize both the constructed and MATLAB models, for them to be more powerful and thus able to predict more properly. However, these modifications would overcome the limitations of this research. One of the most significant areas to be improved is the parameters or independent variables which have to be carefully selected. Since drilling operations involve many parameters and some of them have no direct nor indirect relation with stuck pipe occurrences. Thus such parameters may have to be eliminated and should not be considered in the training and testing stages. Additionally, some parameters may create some noise to the system as their changes have no relation to tendency

toward stuck pipe. Therefore, prior to selecting the parameters that are going to be used in the training process, a study of the parameters is recommended in order to measure the relation between the dependent and the independent variable.

Another aspect of this research to be developed further is with respect to the quality of the utilized data. The data have to possess a high level of quality and authenticity, such that they are carefully collected and recorded. Since drilling operation is a 24 hour operation, figures of the daily drilling report might not be representative of the instantaneous situation of stuck pipe. For example, mud properties might be altered during drilling operation and the recorded figures are based on the time the mud engineer recorded the data. These data might not be descriptive to the stuck pipe situation. Thus, having these data in the training loop of ANNs and SVMs would significantly affect the training and also the prediction processes negatively. Therefore, the data that are going to be collected and utilized, especially in the training stage of machine learning techniques, have to be representative of stuck pipe situations. Otherwise, noise might be created and the accuracy of prediction might be minimized.

In the case of ANNs, cross validation is vital and important to achieve the most powerful developed model. The computational process of the artificial neural network models is based on updates in the hidden layers and the error is decreased during the early updates. However, the updates or the number of epochs will reach to a point at which the magnitude of the error would start to increase. When this shift occurs, it is recommended to stop the model and cross validate. Otherwise, the magnitude of the error would keep increasing and the model would have not been optimized properly.

In addition, the number of hidden layers in the artificial neural network model has to be optimized since additional layers might cause the model to go through a memorization process. Consequently, the constructed model has to be designed in such a way that can optimize the number of hidden layer toward the highest accuracy of prediction. Furthermore, the activation functions to be applied in the constructed models of ANNs have to be properly selected. Although there are different activation functions in the industry, there are some that might be the most proper ones for the data or problem in which ANNs are going to be applied.

In contrast, support vector machines have fewer parameters that can be optimized to generate the most powerful developed model. For example, in linear kernel model, C_value is the only constant that can be optimized to construct the best model in terms of prediction. On the other hand, the SVMs designed with the Radial Bases Function have two parameters that can be altered to create the most optimized model.

CHAPTER VIII

CONCLUSIONS

Although this study involves several findings, the followings are the most important points and conclusions of this research:

1. Stuck pipe is a significant problem in drilling engineering and it requires huge expenditures to get it fixed.
2. By being proactive, the industry can avoid stuck pipe incidents and can save both time and money.
3. ANNs and SVMs are two very powerful tools for predicting stuck pipe incidents which is otherwise very complicated because of the number of variables.
4. This research has shown that machine learning techniques are able to predict the stuck pipe with reasonable accuracy which is more than 83% based on the data that have been utilized.
5. The results showed that SVMs are more accurate in stuck pipe prediction than ANNs based on the data utilized. However, it may not be the case if the different data would have been utilized instead.
6. Based on this study, SVMs are easier to be handled than ANNs since they need one single or two parameters at most to be optimized.
7. Optimization process has to be utilized to construct the most powerful ANN and SVM models.

8. The constructed models generally apply very well in the areas for which they are built but may not work for other areas.
9. The developed models are important especially when it comes to probability measures. Thus, they can be utilized with real-time data and would represent the results on a log viewer.

BIBLIOGRAPHY

1. Agarwal, S. 2008. Auto Release Drill Collars. Paper SPE 112348 presented at the Indian Oil and Gas Technical Conference and Exhibition, Mumbai, India, 4-6 March 2008.
2. Alpaydin, E. 2010. *Introduction to Machine Learning*, second edition. Cambridge, Cambridge, Massachusetts, The MIT Press.
3. Bradley, W. et al. 1991. A Task Force Approach to Reducing Stuck Pipe Costs. Paper SPE/IADC 21999 presented at the SPE/IADC Drilling Conference, Amsterdam, The Netherlands, March 1991.
4. Biegler, M and Kuhn, G. 1994. Advances in Prediction of Stuck Pipe Using Multivariate Statistical Analysis. Paper IADC/SPE 27529 presented at the IADC/SPE Drilling Conference, Dallas, Texas, U.S.A., 15-18 February 1994.
5. Colin Bowes and Ray Procter. 1997. 1997 Drillers Stuck pipe Handbook. Ballater, Scotland: Procter & Collins Ltd.
6. Courteille, J., Fabre, M. and Hollander, C. 1986. An Advanced Solution: The Drilling Advisor. J. Pet Tech, August 1986: 899-904; SPE-12072.
7. DeGeare, J., Haughton, D., and McGurk, M. 2003. *The Guide to Oilwell Fishing Operations*. Burlington, Massachusetts, Gulf Professional Publishing.
8. Dupriest, F., Elks Jr., W., and Ottesen S. 2010. Design Methodology and Operational Practices Eliminate Differential Sticking. Paper IADC/SPE 128129 presented at the Drilling Conference and Exhibition, New Orleans, Louisiana, U.S.A., 2-4 February 2010.
9. Fisk, J. Wood, R. and Kirsner, J. 1998. Development and Field Trial of Two Environmentally Safe Water-Based Fluids Used Sequentially to Free Stuck Pipe. SPE Drilling & Completion, March 1998: 52-55; SPE-35060.
10. Graupe, D. 1997. *Principles of Artificial Neural Networks*. Singapore, World Scientific Publishing Co. Pte. Ltd.

11. Gonzalez, O., Bernat, H., and Moore, P. 2007. The Extraction of Mud-Stuck Tubulars Using Vibratory Resonant Techniques. Paper SPE 109530 presented at the SPE Annual Technical Conference and Exhibition, Anaheim, California, U.S.A., 11-14 November 2007.
12. Gurney, K. 1997. *An Introduction to Neural Networks*. London, UK, UCL Press Limited.
13. Halliday, W. and Clapper, D. 1989. Toxicity and Performance Testing of Non-Oil Spotting Fluid for Differentially Stuck Pipe. Paper SPE/IADC 18684 presented at the SPE/IADC Drilling Conference, New Orleans, Louisiana, U.S.A., 28 February-3 March 1989.
14. Hemphkins, W. et al. 1987. Multivariate Statistical Analysis of Stuck Drillpipe Situations. SPE Drilling Engineering, September 1987: 237-244; SPE 14181.
15. Howard, J. and Glover, S. 1994. Tracking Stuck Pipe Probability While Drilling. Paper IADC/SPE 27528 presented at the IADC/SPE Drilling Conference, Dallas, Texas, U.S.A., 15-18 February 1994.
16. Jardine, S., McCann, D., and Barber, S. 1992. An Advanced System for the Early Detection of Sticking Pipe. Paper IADC/SPE 23915 presented at the IADC/SPE Drilling Conference, New Orleans, Louisiana, U.S.A., 18-21 February 1992.
17. Kecman, V. 2001. *Learning and Soft Computing*. Cambridge, Massachusetts, The MIT Press.
18. Love, T. 1983. Stickiness Factor - A New Way of Looking at Stuck Pipe. IADC/SPE 11383 presented at the IADC/SPE Drilling Conference, New Orleans, Louisiana, U.S.A., 20-23 February 1983.
19. MATLAB Software. R2010a. MathWorks.
<http://www.mathworks.com/products/MATLAB/>.
20. Meschi, M., Shahbazi, K., and Shahri, P. 2010. A New Criteria to Predict Stuck Pipe Occurrence. Paper SPE 128376 presented at the SPE North Africa Technical Conference and Exhibition, Cairo, Egypt, 14-17 February 2010.

21. Mingxin, Z. et al. 2009. Comprehensive Geomechanics Study Helps Mitigate Sever Stuck Pipe Problems in Development Drilling in Bohai Bay, China. Paper IPTC 14091 presented at the International Petroleum Conference, Doha, Qatar, 7-9 December 2009.
22. Miri, R., Sampaio, J., and Afshar, M. 2007. Development of Artificial Neural Networks to Predict Differential Pipe Sticking in Iranian Offshore Oil Fields. Paper SPE 108500 presented at the International Oil Conference and Exhibition, Veracruz, Mexico, 27-30 June 2007.
23. Montgomery, J. et al. 2007. Improved Method for Use of Chelation to Free Stuck Pipe and Enhance Treatment of Lost Returns. Paper SPE/IADC 105567 presented at the SPE/IADC Drilling Conference, Amsterdam, The Netherlands, 20-22 February 2007.
24. Murillo, A. and Neuman, J. 2009. Pipe Sticking Prediction and Avoidance Using Adaptive Fuzzy Logic and Neural Network Modeling. Paper SPE 120128 presented at the SPE Production and Operations Symposium, Oklahoma City, Oklahoma, U.S.A., 4-8 April 2009.
25. Scholkopf, B. and Smola, A. 2002. *Learning With Kernels*. Cambridge, Massachusetts, The MIT Press.
26. Shoraka, S., Shadizadeh, S., and Shahri, P. 2011. Prediction of Stuck Pipe in Iranian South Oil Fields Using Multivariate Statistical Analysis. Paper SPE 151076 presented at the Nigeria Annual International Conference and Exhibition, Abuja, Nigeria, 30 July-3 August 2011.
27. Siruvuri, C., Nagarakanti, S., and Samuel, R. 2006. Stuck Pipe Prediction and Avoidance: A Convolutional Neural Network Approach. Paper IADC/SPE 98378 presented at the IADC/SPE Drilling Conference, Miami, Florida, USA, 21-23 February 2006.
28. Torne, J. et al. 2011. Middle East Case-Study Review of a New Free-Pipe Log for Stuck Pipe Determination and Pipe-Recovery Techniques. Paper SPE 147859 presented at the SPE Asia Pacific Oil and Gas Conference and Exhibition, Jakarta, Indonesia, 20-22 September 2011.

29. Yarim, G. 2007. Stuck Pipe Prevention - A Proactive Solution to and Old Problem. Paper SPE 109914 presented at the SPE Annual Technical Conference and Exhibition, Anaheim, California, U.S.A., 11-14 November 2007.

BIOGRAPHY

Islam Albaiyat was born in June 27th, 1983 in Qatif, Saudi Arabia. He is a member of SPE (Society of Petroleum Engineering). His area of interest is drilling engineering. He received his BS degree from King Fahd University of Petroleum and Minerals in Dhahran, Saudi Arabia in 2006 as a petroleum engineer with 2nd honor. At the same year, he joined Saudi Aramco and worked as drilling engineer in the Gas Drilling Department. In 2007, Mr. Albaiyat was involved in drilling engineering school run by a British company inside Saudi Aramco for 15 months, and ranked top in the class with the highest score. He handled individually gas and exploration rigs for almost 2 years, designed and engineered several challenging gas drilling programs, such as high slanted, horizontal, and delineation wells. During his career, Mr. Albaiyat supervised and witnessed all the routine rig jobs such as running different casing sizes, cementing in deviated holes, completions, well flow back, Drill Stem Tests. He gained experience in running several advanced tools such as Rotary Steerable Systems and MDT logs.

APPENDIX

Table A.1. Outputs of all designed models on all datasets of Group_A

Group_A_All Datasets															
2-class												3-class			
Testing Data	F1	F2	F3	F4	SVM_linear	SVM_rbf	ANN_Matlab	SVM_Matlab	libsvm_L	libsvm_P	True Index	ANN_Matlab	libsvm_L	libsvm_P	True Index
1	-1	1	-1	-1	1	1	1	1	1	1	1	3	3	3	3
2	-1	1	-1	-1	1	1	1	1	1	1	1	3	3	3	3
3	-1	1	-1	-1	1	1	1	1	1	1	1	3	3	3	3
4	-1	1	-1	-1	1	1	1	1	1	1	1	3	3	3	3
5	-1	-1	-1	-1	1	1	1	1	1	1	1	3	3	3	3
6	-1	1	-1	-1	1	1	1	1	1	1	1	2	2	3	2
7	-1	-1	-1	-1	1	1	1	1	1	1	1	3	3	3	2
8	-1	1	-1	-1	1	1	1	1	1	1	1	2	2	3	2
9	-1	-1	-1	-1	1	1	1	1	1	1	1	2	2	3	2
10	-1	1	-1	-1	1	1	1	1	1	1	1	2	2	3	2
11	-1	1	-1	-1	1	1	1	1	1	1	1	2	3	3	2
12	-1	1	-1	-1	1	1	1	1	1	1	1	2	3	3	2
13	-1	1	-1	1	-1	-1	-1	-1	-1	1	-1	1	1	3	1
14	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	1	1	3	1
15	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	1	1	3	1
16	-1	1	-1	-1	-1	-1	-1	-1	-1	1	-1	1	1	3	1
17	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	1	1	3	1
18	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	1	3	3	1
MSE_testing	2.667	1.111	2.667	2.889	0.000	0.000	0.000	0.000	0.000	1.333		0.056	0.389	1.72	
MSE_learning	0.570	0.793	0.410	0.447	C_Value=0.45		C=0.7, Sigma=0.4								
Output Layer	Sigmoid	Sigmoid	Tanh	Tanh											
Hidden Layer	Sigmoid	Tanh	Tanh	Sigmoid											
Correct	6	13	6	5	18	18	18	18	18	12		17	14	5	
Misclassified	12	5	12	13	0	0	0	0	0	6		1	4	13	
Correct %	33	72	33	28	100	100	100	100	100	67		94	78	28	
Misclassified %	67	28	67	72	0	0	0	0	0	33		6	22	72	

Table A.2. Outputs of all designed models on mechanical stuck datasets of Group_A

Group_A_MS Datasets											
Testing Data	F1	F2	F3	F4	SVM_linear	SVM_rbf	SVM_Matlab	ANN_Matlab	libsvm_L	libsvm_P	True Index
1	-1	-1	-1	-1	1	1	1	1	1	1	1
2	-1	-1	1	-1	1	1	1	1	1	1	1
3	-1	-1	-1	-1	1	1	1	-1	1	1	1
4	-1	-1	-1	-1	1	1	1	1	1	1	1
5	-1	-1	-1	-1	1	1	1	1	1	1	1
6	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1
7	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1
8	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1
9	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1
10	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1
11	-1	1	-1	-1	1	1	-1	-1	1	1	-1
MSE_testing	1.818	2.182	1.455	1.818	0.364	0.108	0.000	0.364	0.364	2.182	
MSE_learning	0.655	0.514	0.337	0.336	C_Value=0.5	C=1, Sigma=0.37					
Output Layer	Sigmoid	Sigmoid	Tanh	Tanh							
Hidden Layer	Sigmoid	Tanh	Tanh	Sigmoid							
Correct	6	5	7	6	10	10	11	10	10	5	
Misclassified	5	6	4	5	1	1	0	1	1	6	
Correct %	55	45	64	55	91	91	100	91	91	45	
Misclassified %	45	55	36	45	9	9	0	9	9	55	

Table A.3. Outputs of all designed models on differential stuck datasets of Group_A

Group_A_DS Datasets											
Testing Data	F1	F2	F3	F4	SVM_linear	SVM_rbf	SVM_Matlab	ANN_Matlab	libsvm_L	libsvm_P	True Index
1	1	1	1	1	1	-1	1	1	1	-1	1
2	-1	-1	-1	-1	1	1	1	1	1	-1	1
3	1	1	1	1	1	-1	1	1	1	-1	1
4	1	1	1	1	1	-1	1	1	1	-1	1
5	1	-1	-1	-1	1	1	1	1	1	-1	1
6	-1	-1	-1	-1	-1	1	1	1	-1	-1	1
7	-1	-1	-1	-1	-1	1	1	1	-1	-1	1
8	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
9	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
10	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
11	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
12	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
13	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
MSE_testing	0.923	1.231	1.231	1.231	0.615	0.923	0.000	0.000	0.615	2.154	
MSE_learning	0.511	0.776	0.497	0.512	C_Value=0.35	C=0.5, Sigma=0.33					
Output Layer	Sigmoid	Sigmoid	Tanh	Tanh							
Hidden Layer	Sigmoid	Tanh	Tanh	Sigmoid							
Correct	10	9	9	9	11	10	13	13	11	6	
Misclassified	3	4	4	4	2	3	0	0	2	7	
Correct %	77	69	69	69	85	77	100	100	85	46	
Misclassified %	23	31	31	31	15	23	0	0	15	54	

Table A.4. Outputs of all designed models on datasets of Company_X Field no. 6

Company_X_Field no. 6											
Testing Data	F1	F2	F3	F4	SVM_linear	SVM_rbf	SVM_Matlab	ANN_Matlab	libsvm_L	libsvm_P	True Index
1	-1	-1	-1	-1	-1	1	-1	-1	-1	1	1
2	1	-1	-1	-1	1	1	1	1	1	1	1
3	1	-1	-1	-1	1	1	1	1	1	1	1
4	1	1	-1	-1	1	1	1	1	1	1	1
5	1	-1	-1	-1	1	1	1	-1	1	1	1
6	-1	-1	-1	-1	-1	1	1	1	1	1	1
7	1	-1	-1	-1	1	1	1	-1	1	1	1
8	1	-1	-1	-1	1	1	1	1	1	1	1
9	1	-1	-1	-1	1	1	1	1	1	1	1
10	1	1	1	-1	1	1	1	1	1	1	1
11	1	-1	-1	-1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1
13	1	-1	-1	-1	-1	1	1	-1	1	1	-1
14	-1	-1	-1	-1	-1	1	-1	-1	-1	1	-1
15	-1	-1	-1	-1	-1	1	-1	-1	-1	1	-1
16	-1	-1	-1	-1	-1	1	-1	-1	-1	1	-1
17	1	-1	-1	-1	1	1	1	1	1	1	-1
18	1	-1	-1	-1	-1	1	-1	-1	1	1	-1
19	-1	-1	-1	-1	-1	1	-1	-1	1	1	-1
20	1	-1	-1	-1	-1	1	1	-1	1	1	-1
21	1	-1	-1	-1	-1	1	-1	-1	1	1	-1
22	1	-1	-1	-1	1	1	1	1	1	1	-1
23	-1	1	-1	-1	-1	1	-1	-1	-1	1	-1
24	-1	-1	-1	-1	-1	1	-1	1	1	1	-1
MSE_testing	1.333	1.667	1.667	1.833	0.667	2.000	0.833	1.000	1.500	2.000	
MSE_learning	0.800	0.643	0.348	0.396							
Output Layer	Sigmoid	Sigmoid	Tanh	Tanh							
Hidden Layer	Sigmoid	Tanh	Tanh	Sigmoid							
Correct	16	14	14	13	20	12	19	18	15	12	
Misclassified	8	10	10	11	4	12	5	6	9	12	
Correct %	67	58	58	54	83	50	79	75	63	50	
Misclassified %	33	42	42	46	17	50	21	25	38	50	

Table A.5. Optimization of SVM_Linear model for Group_A datasets

SVM_Linear Optimization for Group_A										
All Datasets				MS				DS		
C_Values	MSE	Accuracy		C_Value	MSE	Accuracy		C_Value	MSE	Accuracy
0.01	1.285	54		0.45	0.470	91		0.1	0.8538	46
0.1	0.897	54		0.5	0.343	91		0.3	0.3639	85
0.2	0.568	54		0.55	0.368	91		0.31	0.3495	85
0.25	0.443	69		0.9	0.478	91		0.32	0.3239	85
0.3	0.345	92		1	0.588	91		0.34	0.2972	85
0.35	0.274	92						0.35	0.2911	85
0.4	0.230	100						0.4	0.4252	85
0.41	0.224	100						0.5	0.5121	85
0.42	0.219	100						0.9	0.7655	85
0.43	0.216	100								
0.44	0.214	100								
0.45	0.212	100								
0.46	0.230	100								
0.47	0.218	100								
0.5	0.225	100								
0.6	0.272	100								
0.8	0.623	100								
1	0.804	100								

Table A.6. Optimization of SVM_RBF model for Group_A datasets

SVM_RBF Optimization for Group_A													
All Datasets					MS					DS			
C_Value	Sigma	MSE	Accuracy		C_Value	Sigma	MSE	Accuracy		C_Value	Sigma	MSE	Accuracy
0.1	0.1	0.8491	72		0.2	0.1	0.5411	92		0.2	0.1	0.6174	62
0.5	0.1	0.8536	72		0.3	0.1	0.5411	92		0.4	0.1	0.6174	62
0.5	0.2	0.6481	100		0.4	0.1	0.5411	92		0.4	0.2	0.4922	62
0.2	0.2	0.6274	100		0.4	0.2	0.3621	92		0.5	0.2	0.4922	77
0.2	0.3	0.4145	100		0.7	0.2	0.3621	92		1	0.3	0.3821	77
0.2	0.4	0.5185	83		0.5	0.3	0.1828	92		0.5	0.3	0.3821	77
0.5	0.4	0.2743	94		0.5	0.4	0.1884	92		0.5	0.35	0.3613	77
0.6	0.4	0.2697	94		0.5	0.35	0.1513	92		0.5	0.34	0.3571	77
0.6	0.35	0.3033	94		0.6	0.33	0.1449	92		0.5	0.33	0.3555	77
0.4	0.39	0.2882	94		0.6	0.35	0.1361	92		0.5	0.32	0.3562	77
0.8	0.4	0.2817	100		0.7	0.35	0.1269	92					
0.9	0.4	0.2935	100		0.8	0.35	0.1205	92					
0.7	0.4	0.2732	100		1	0.35	0.1147	92					
0.6	0.4	0.2697	94		0.9	0.33	0.1307	92					
0.7	0.5	0.4712	83		0.9	0.34	0.1216	92					
					1.2	0.37	0.1084	92					
					1.5	0.37	0.1084	92					
					1.1	0.37	0.1084	92					
					1	0.37	0.1084	92					
					0.9	0.37	0.1136	92					

Table A.7. Optimization of SVM_Linear and SVM_RBF models for Company_X
Field no. 6 datasets

SVM_Linear and SVM_Linear for Company_X Field no. 6							
Linear				RBF			
C_Value	MSE	Accuracy		C_Value	Sigma	MSE	Accuracy
0.1	9.8	50		0.1	0.1	1.2333	79
0.3	9.83	50		0.4	0.1	1.5655	79
0.5	9.86	58		0.4	0.2	1.2036	79
0.9	9.9348	75		0.4	0.3	1.4822	79
0.8	9.9218	75		0.9	0.4	2.4212	79
0.7	9.9045	75		0.6	0.2	1.3209	79
0.65	9.8959	71		0.35	0.2	1.1989	79
1	9.9564	83		0.35	0.19	1.1815	79
0.9	9.9348	75		0.35	0.18	1.1674	79
				0.34	0.18	1.1641	79
				0.33	0.18	1.164	79
				0.32	0.18	1.1638	79
				0.31	0.18	1.1637	79
				0.3	0.18	1.1635	79
				0.29	0.18	1.1634	79
				0.28	0.18	1.1632	79
				0.275	0.18	1.1642	79

Table A.8. Summary of runs by ANN_MATLAB and SVM_MATLAB on different fields of Company_X datasets

Run#	S/N	Type	Learning								Testing								SVM_Matlab			ANN_Matlab			Average	Difference	% diff. increase			
			Field Selected								Field Selected																			
			1	2	3	4	5	6	7	8	Total	Total	1	2	3	4	5	6	7	8	Total	Total	Corr.	Misclass.				Accu.	Corr.	Misclass.
1	S							10		10	20							7		7	14	12	2	86%	9	5	64%	75%	21%	33%
	N							10		10								7		7										
2	S							10	8	18	28							7		7	14	10	4	71%	10	4	71%	71%	0%	0%
	N							10		10								7		7										
3	S							10	8	18	36							7		7	14	9	5	64%	11	3	79%	71%	14%	22%
	N							10	8	18								7		7										
4	S							10	6	16	32							7	2	9	18	10	8	56%	15	3	83%	69%	28%	50%
	N							10	6	16								7	2	9										
5	S							10	4	6	20	40						7	2	9	18	11	7	61%	14	4	78%	69%	17%	27%
	N							10	4	6	20							7	2	9										
6	S							10	4	6	20	40						7	1	2	19	12	7	63%	15	4	79%	71%	16%	25%
	N							10	4	6	20							7	2	9										
7	S	Packoff	1	1	1	1	1	2	1	2	9	18						3		3	6	5	1	83%	5	1	83%	83%	0%	0%
	N							6	2	1	9							3		3										
8	S	Packoff	1	1	1	1	1	1	2	7	14		1					3	1	5	10	8	2	80%	9	1	90%	85%	10%	13%
	N							5	2	7								1	4	5										
9	S	Packoff	1			1	1	1	1	5	10		1	1				2	2	6	14	10	4	71%	11	3	79%	75%	7%	10%
	N							4	1	5								3	4	1										
10	S	Packoff		1				3		1	5	10	1	1	1	1	1	2	7	14	9	5	64%	10	4	71%	68%	7%	11%	
	N							1	4	5								5	2	7										
11	S							17		17	34		6	5	10	1		3	11	36	36	27	9	75%	23	13	64%	69%	11%	17%
	N							17		17										0										
12	S							17		17	34		6	5	10					21	37	24	13	65%	25	12	68%	66%	3%	4%
	N							17		17		1	6	4	5					16										
13	S							17		17	34									0	35	25	10	71%	21	14	60%	66%	11%	19%
	N							17		17		1	6	4	5	2		9	8	35										
14	S							17		17	34		6	5	10					21	36	31	5	86%	35	1	97%	92%	11%	13%
	N							17		17								7	8	15	36	31								
15	S							18		12	30	60						20	8	10	50	35	15	70%	31	19	62%	66%	8%	13%
	N							14	9	7	30							7	2	3										
16	S							17		17	34							21		21	25	21	4	84%	20	5	80%	82%	4%	5%
	N							17		17								4		4										
17	S							17		17	29							21		21	30	22	8	73%	21	9	70%	72%	3%	5%
	N							12		12								9		9										
18	S							12		12	24							26		26	35	25	10	71%	26	9	74%	73%	3%	4%
	N							12		12								9		9										
19	S							26		26	35							12		12	24	19	5	79%	18	6	75%	77%	4%	6%
	N							9		9								12		12										
20	S							6		6	12							16		16	19	14	5	74%	13	6	68%	71%	5%	8%
	N							6		6								3		3										
21	S		11		11			25	6	15	68	116	6	6	12	2	7	33		33	53	34	19	64%	37	16	70%	67%	6%	9%
	N		6		5			20	11	6	48		2	3	7	5	3	20		20										
22	S		8		8			17	6	10	49	97	9	9	20	2	12	52		52	72	49	23	68%	54	18	75%	72%	7%	10%
	N		6		5			21	10	6	48		2	3	6	6	3	20		20										
23	S							17		6	46							21	4	25	44	25	19	57%	31	13	70%	64%	14%	24%
	N							17		6								16	3	19										