Anomaly Detection in Oil Well Drilling Operation Using Artificial Intelligence-Based Approaches

Thesis submitted in partial fulfillment of the requirements for the award of the degree

of

Doctor of Philosophy

in

Computer Science and Engineering

Submitted by

Achyut Mani Tripathi

Under the guidance of

Dr. Rashmi Dutta Baruah

Department of Computer Science and Engineering

Indian Institute of Technology Guwahati

January, 2020

Abstract

Artificial intelligence (AI) based approaches and in particular machine learning techniques have been extensively applied in domains such as health care, computer vision, and network security to build complex and accurate models that can produce more efficient solutions. Oil and gas sector is an area that generates massive and high volume data during the extraction of oil and gases. Stuck pipe, borehole instability, washout, and kick are among the more recurrent problems that occur during drilling operation and cause enormous financial loss to the oil and gas industries. In the ongoing situation, these problems are solved by first principle models and also appeal highly experienced drillers who can prevent such unwanted situations. Machine learning and AI techniques have shown tremendous performance to solve various research problems that involve massive real-time data, but their capabilities have not been explored entirely in the domain of oil industries. Still, there is a requirement of data-driven models that can solve the oil well drilling complications.

The oil well drilling process needs a mechanical framework, also known as a rig. The rig contains different functional units having multiple sensors that provide the measurements of different hydraulic and mechanical parameters further helpful to monitor the oil well drilling process. The data measured by the rig sensors are stored in a database known as supervisory control and data acquisition (SCADA) system. The data stored in the SCADA system is multivariate time series data. The multivariate time series data stored in the SCADA system can be utilized to develop various machine learning models that can accurately provide the ongoing insight of the oil well drilling process. These data-driven supervisory models can be used for identifying oil well drilling complications. This research work primarily aims at developing

Ai-based models that can be used to realize systems capable of automatically detecting anomalies during oil well drilling operations. The focus is on stuck pipe anomalies that care recurrent during the drilling operation. The above mentioned aim is attained through the following three contributions:

The first contribution is development of a two-level classifier that identifies oil well drilling activities from the real-time oil well drilling data and also provides a detailed report that shows the percentage of time the drilling activity is performed in one complete cycle of the oil well drilling. The second contribution describes a novel probabilistic model that combines Dynamic Naive Bayesian Classifier and Fuzzy AdaBoost to identify the anomalies that lead to stuck pipe complication during the oil well drilling process. The last contribution explains novel Contextual Dynamic Bayesian Network that detects contextual anomalies that occur during the oil well drilling process. All the developed models have been tested using real data from various wells located in Assam. The activity detection module has also been validated by deploying it at the well sites and the results are satisfactory.

Declaration

I certify that:

a. The work presented in this thesis is original and has been done by me under the guidance of my supervisor.

b. The work has not been submitted to any other Institute for any degree or diploma.

c. I have followed the guidelines provided by the Institute in preparing the thesis.

d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Achyut Mani Tripathi

Copyright

Signature of Author.....

Achyut Mani Tripathi

Certificate

This is to certify that this thesis entitled, "**Anomaly Detection in Oil Well Drilling Operation Using Artificial Intelligence-Based Approaches**", being submitted by **Achyut Mani Tripathi**, to the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, for partial fulfillment of the award of the degree of Doctor of Philosophy, is a bonafide work carried out by him under my supervision and guidance. The thesis, in my opinion, is worthy of consideration for award of the degree of Doctor of Philosophy in accordance with the regulation of the institute. To the best of my knowledge, it has not been submitted elsewhere for the award of the degree.

.....

Dr.Rashmi Dutta Baruah
Assistant Professor
Department of Computer Science and Engineering
IIT Guwahati

.....

Dedicated to
Chiku

Acknowledgments

The journey of my Ph.D. life would not have been possible without the help and contribution of various people. I am thankful to all those people who devoted their time and efforts throughout the completion of this journey.

At first, I would like to express my innermost acknowledgment towards my thesis supervisor **Dr. Rashmi Dutta Baruah**, for guiding me to learn the basics of research and teaching. Her advice, inspiration, and assistance at every step always directed me towards the fruitful directions. Her down to earth behavior and unconditional support always kept me grounded and helpful to make this journey memorable. I would also like to thank her for believing me and providing a dedicated research environment and vast resources that are helpful to do my research tasks.

I would also like to pay my thanks to **Prof. S.B. Nair, Dr. Sanasam Ranbir Singh, and Dr. Senthilmurgans S** for their critical reviews and comments that are helpful to overcome the weaker parts of my research. During this journey, I spent significant time with my research fellows **Sonia, Pallabi and Dipankar** who helped me and would like to thank all of them for providing professional support during this journey. I want to give special thanks to **Selva Senthil, Vishwanath, and Balavignesh** for their help and discussions during the completion of the ONGC project. Special thanks to my seniors **Shashi Sekhar Jha, Shirsendu Das and, Shilpa Budhkar** for helping me in all aspects. Along with this, I appreciate the CSE dept staff for their technical support.

I would also like to appreciate my friends **Niraj Kant Sinha, Sammer A. Pandit and Pratik Agrawal, Konark** for their help and guidance during the preparation of the GATE examination. Your friendship always motivated me to do more better in my life. I would also like to thank my friends in

IIT Guwahati **Bumboo, Naveen, Vasudevan, Ashwathy, Aaakansha, Rakesh, Piyush, Prateek, Puneet, Rohit, Jumbo, Promit, Pathak, Shivshant, Manish, Lakshya, Ashish, Pankaj and Panthadeep** who fulfilled my life with positivity and happiness. You people are among the few ones who touched my soul with your beautiful nature and friendship. I want to thank the sports board of IIT Guwahati for providing an excellent sports facility inside the campus. I would like to thanks all my **Hockey Team members** with whom I played different versions of the Inter IIT Sports meet. I would like to thank IDT (ONGC) Dehradun, India for providing real-time drilling data to validate models proposed in this thesis.

Last but not least, I express my deep indebtedness to my parents **Dr. Radheshaym Mani Tripathi and Smt. Sudha Tripathi** for all the motivations, lessons, support, and encouragement. I also want to thank my sister **Vishnukanta Tripathi** and my twin **Aaditya Mani Tripathi** for the unconditional support, love, and beautiful memories we enjoyed together till date.

Contents

List of Figures

List of Algorithms

List of Tables

List of Acronyms

AB *AdaBoost*

AI *Artificial Intelligence*

ANFIS *Adaptive Neuro Fuzzy Inference System*

ANN *Artificial Neural Network*

BD *Bit Depth*

Belief *Belief of State*

BHA *Bottom Hole Assembly*

BP *Block Position*

CBR *Case-Based Reasoning*

CRWO *Circulation Without Rotation*

CxDBN *Contextual Dynamic Bayesian Network*

DBN *Dynamic Bayesian Network*

DDR *Daily Drilling Report*

DI *Inlet Density*

DNBC *Dynamic Naive Bayesian Classifier*

DO *Outlet Density*

DRWO *Drilling Without Rotation*

DRWR *Drilling With Rotation*

DT *Decision Tree*

DTW *Dynamic Time Wrapping*

F1 *F1 Score*

FAB *Fuzzy AdaBoost*

FOU *Footprint of Uncertainty*

FPR *False Positive Rate*

FRB *Fuzzy Rule-Based*

GTO *Geo-Technical Order*

GUI *Graphical User Interface*

HL *Hookload*

HMM *Hidden Markov Model*

IVFS *Interval Valued Fuzzy Set*

KNN *K Nearest Neighbours*

ML *Machine Learning*

MTS *Multivariate Time Series*

NOP *No Operation*

NPT *Non-Productive Time*

OCSVM *One Class Support Vector Machine*

ONGC *Oil and Natural Gas Corporation of India*

PAA *Piecewise Aggregated Approximation*

RECI *Reciprocation*

RF *Random Forest*

ROFB *Rotation Off Bottom*

RONB *Rotation On Bottom*

ROP *Rate of Penetration*

RPM *Rotation Per Minute*

RTDD *Real-Time Drilling Data*

SCADA *Supervisory Control and Data Acquisition System*

SPM *Stokes Per Minute*

STP *Stand Pipe Pressure*

SVM *Support Vector Machine*

TD *Total Depth*

TPR *True Positive Rate*

TRIWO *Tripping In Without Rotation*

TRIWR *Tripping In With Rotation*

TROWO *Tripping Out Without Rotation*

TROWR *Tripping Out With Rotation*

TS *Time series*

TSK *Takagi-Sugeno-Kang*

WOB *Weight ON Bit*

List of Symbols

$\mathbf{x}$	Data instance
w	Weight of each data sample
σ	Degree of dispersion of data along dimension
σ^{max}	Maximum value of degree of dispersion
σ^{min}	Minimum value of degree of dispersion
$R2$	Upper range of FOU
$R1$	Lower range of FOU
FOU	Footprint of uncertainty
$e(k)$	Misclassification Error of k^{th} DNBC
$\overline{\mu}_k$	Upper membership FOU of error
$\underline{\mu}_k$	Lower membership FOU of error
C_k	Confidence of k^{th} DNBC
$\overline{\mu}_{k+1}^{w_i}$	Upper membership of $(k+1)^{th}$ DNBC
$\underline{\mu}_{k+1}^{w_i}$	Lower membership of $(k+1)^{th}$ DNBC
$\overline{Z}_k$	Normalization constant for upper membership

$\underline{Z}_k$	Normalization constant for lower membership
λ	Defuzzification parameter
$y(x_i)$	Actual class label of data x_i
$I(x_i)$	Hypothesis for x_i from k^{th} DNBC
n	Total number of data samples
k	Total number of weak classifiers
$Rule^i$	i^{th} rule
A^i	Antecedent of i^{th} rule
y^i	Output of i^{th} rule
τ	Firing strength of rule
L	weighted aggregated sum
O	Observations
$P(S)$	Probability of state S
π	Initial probability of states
N	Number of hidden states
M	Number of different observation symbols
a_{ij}	Transition probability between state i and j
b_{im}	Emission probability of observation symbol m
α	Forward probability
β	Backward probability
θ	Parameters of HMM model
L	Dimension of observation sequence

R	Number of attributes
X	Random variable X
$Par(X)$	Parents of attribute X
$P(X_1, .., X_R)$	Joint probability
t	Time stamp
$Belief(State)$	Belief of state
W_s	Window size
$Sign(\mathbf{x})$	Sign of $\mathbf{x}$
V_{min}	Minimum value of $\mathbf{x}$
V_{max}	Maximum value of $\mathbf{x}$
b_i	i^{th} cluster center
F_{Low}	Low fuzzy set
F_{Mid}	Medium fuzzy set
F_{High}	High fuzzy set
$\mu_{F_{Low}}(\mathbf{x})$	Membership of $\mathbf{x}$ for low fuzzy set
$\mu_{F_{Mid}}(\mathbf{x})$	Membership of $\mathbf{x}$ for medium fuzzy set
$\mu_{F_{High}}(\mathbf{x})$	Membership of $\mathbf{x}$ for high fuzzy set
I	Increasing trend
D	Decreasing trend
C	Constant trend
H, M, L	High, Medium and Low values
TS_i	i^{th} time series

a_{nn}	Transition probability from normal state to normal state
a_{ns}	Transition probability from normal state to stuck
a_{sn}	Transition probability from stuck state to normal state
a_{ss}	Transition probability from stuck state to stuck state
Y	Drilling Activity
X_{Train}^{Layer1}	Training data at layer 1
X_{Test}^{Layer1}	Test data at layer 1
X_{Train}^{Layer2}	Training data at layer 2
X_{Test}^{Layer2}	Test data at layer 2

Citation to Published Work

Chapter 3 is based on following patent:

- Senthilmurugan S, Rashmi Dutta Baruah, Munawar A. Shaik, **Achyut Mani Tripathi**, Senthil Selvaraju, Viswanth Ramba, Bala Kumara Vignesh M, Amol Musale, Gauba SK, Samal KB, "Decision Support System for Oil and Gas Well Drilling". (**Application No. 201911040595**)

Chapter 4 is based on following paper:

- **Achyut Mani Tripathi**, Rashmi Dutta Baruah, "Anomaly Detection in Multivariate Time Series Using Fuzzy AdaBoost and Dynamic Naive Bayesian Classifier", IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2019), Bari, Italy, October 6-9, 2019.

Chapter 5 is based on following paper:

- **Achyut Mani Tripathi** and Rashmi Dutta Baruah, "Contextual Anomaly Detection in Time Series Using Bynamic Bayesian Network", Asian Conference on Intelligent Information and Database Systems, Phuket, Thailand, March 26-29, 2020.

Chapter 1

Introduction

The increasing demand for sensor-equipped infrastructure for monitoring industrial processes requires an anomaly detection framework as a basic unit to trigger anomalous situations. High dimensional data gathered by the multiple sensors are mined with machine learning techniques to identify the transition of dynamical systems into a precarious state. Any dynamical system's current status can be well identified by mining multivariate time series data generated from numerous integrated sensors. A framework is highly needed to minimize the error and improve the performance of the monitoring system through automation. Every monitoring system has a fundamental unit to detect anomalies. Evolving behavior and pace of time series data can make detection of anomalies more challenging as compared to detection of anomalies in static data. The existing framework of the monitoring applications provides data preprocessing and visualization of the dynamic system. Handling such a massive and high volume of data is challenging and needs effective data handling techniques that can model the uncertainties present in the data received by the numerous sensors while detecting the anomalies.

Anomaly detection mainly deals with identifying data samples that show significant deviation against the other data samples or unforeseen act. The unexpected behavior of data samples is known as anomaly or outlier [52]. The classification or detection of the anomalous pattern from the massive data generated by the domain such as industrial monitoring will result in valuable information. For example, anomalous MRI images identified by computer vision techniques can be a prime indication of tumors, or un-

expected deviation in sensors reading in industrial monitoring may indicate some fault in industrial process. Identification of anomalies was started in the early days of 19^{th} century. Initially, numerous methods based on statistical computations were proposed to detect the anomaly [67]. According to the survey of Varun et al. [52] there are three type of anomalies.

- **Point Anomaly**

  Point anomaly is an anomaly when single data point shows unexpected behavior. Figure 1.1 shows an example of point anomaly where the data sample at index 52 shows anomalous behavior.

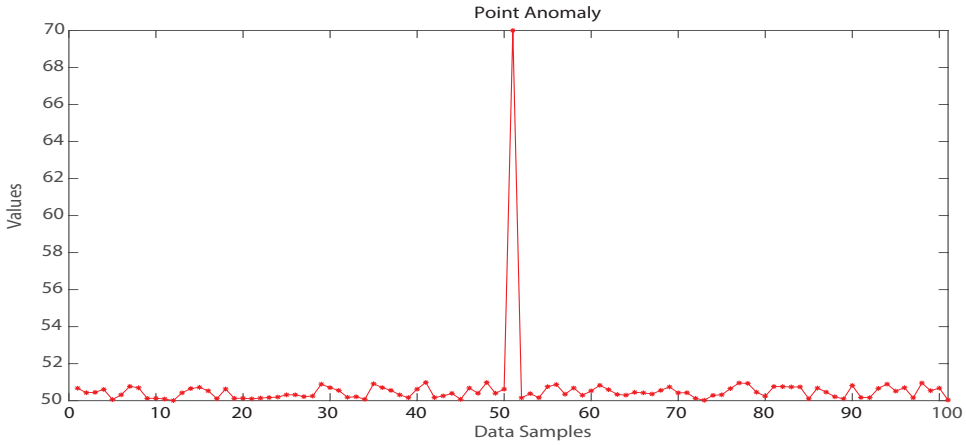


Figure 1.1: *Point Anomaly*

- **Collective Anomaly**

  Unforeseen behavior shown by collection of data points is known as collective anomalies. In Figure 1.2 the data samples from index 50 to 60 are the collective anomalies.

- **Contextual Anomaly**

  Contextual anomalies are the special type of anomalies where the unexpected behavior of the data samples is defined based on the given context. Figure 1.3 shows the contextual anomalies where the value of temperature sensor is declared as anomalous when the season is summer.

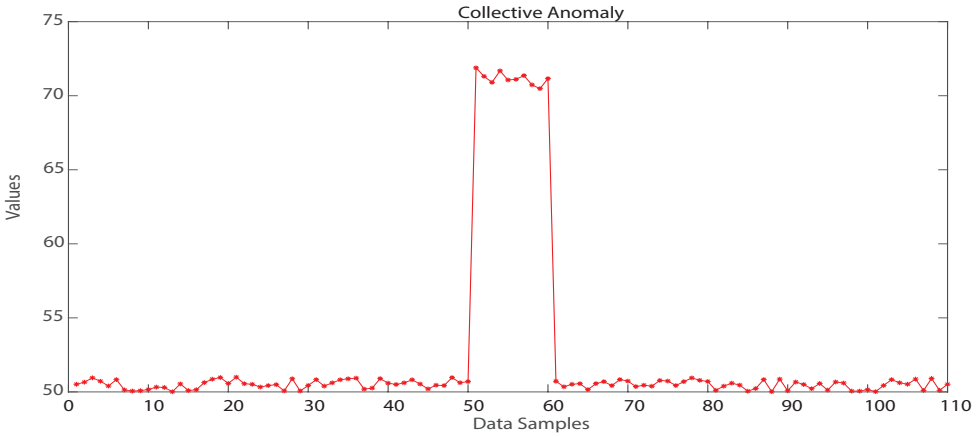


Figure 1.2: *Collective Anomaly*

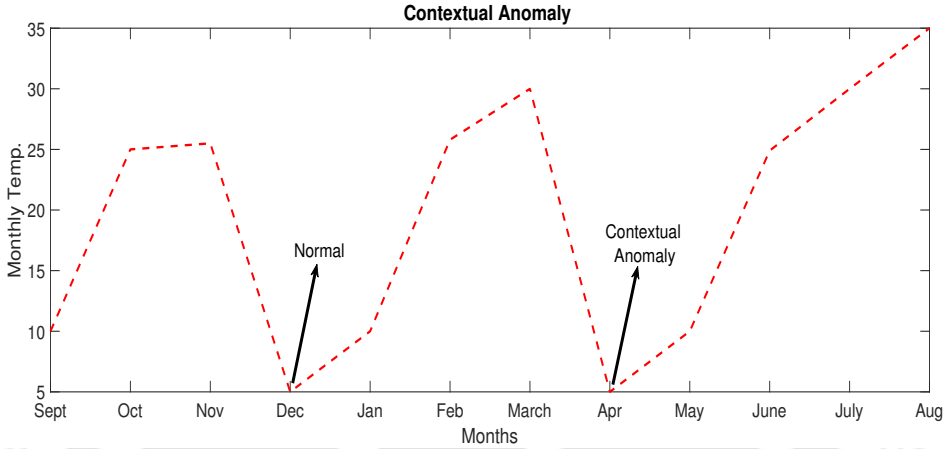


Figure 1.3: *Contextual Anomaly*

Extraction of oil and gas needs a mechanical framework. i.e. rig, to drill the deep formations. Figure 1.4 shows a typical structure of the oil well rig. Seismic survey of the thick soil layers provides all necessary details useful to decide well regions prosperous for the production of oil and gases. The drilling process involves huge expenditure and complications. Improvement in technologies boost the use of modern drilling apparatuses and sensor-equipped monitoring infrastructures to diminish the drilling cost and complexities. The Oil well rig accommodates various sensors integrated to different working units of the rig to provide better insight into the ongoing drilling process. Currently oil industries have deployed Supervisory Control, and Data Acquisition (SCADA) [92] system to

Figure 1.4: *Oil Well Drilling Rig*

collect various drilling parameters. Supervision of ongoing drilling process is achieved by continuous monitoring of the drilling parameters like Hookload (HL), Weight on Bit (WOB), Rotation Per Minute (RPM), Torque, Rate of Penetration (ROP), Mud Properties, Total depth (TD), Bit depth (BD), Block Position (BP), Inlet and Outlet densities collected in the SCADA system. Optimized and safe execution of the drilling process requires continuous monitoring of various drilling parameters. The SCADA engineers continuously monitor the parameters mentioned above and generate alerts whenever the drilling process makes an entrance into problematic situations. The continuous monitoring performed by the SCADA engineers is exhaustive and may cause a high false alarm rate. Stuck pipe [220], [185], the condition can be considered as an anomaly that can be identified through the acquired SCADA drilling parameters. One way to detect these anomalies is to inspect the massive and high dimensional multivariate time series (MTS) data collected in the SCADA system using artificial intelligence (AI) and machine learning (ML) techniques. Hole deviation [217], Loss circulation [77], Drill pipe failure [110], Washout trouble [215], Mud contamination [115], Hole cleaning [225] and deterioration of drilling equipments [4] are the other complications associated with the oil well drilling process. To resolve the problem of high false alarm rate AI and ML techniques are used to develop models using the high dimensional oil well drilling data.

1.1 Research Challenges in Oil Well Drilling

This research work focuses on some of the challenges that need to be addressed to realize a system to identify abnormal conditions in the oil well drilling process automatically. The major challenges involved in the oil well drilling process are as follows:

1. The oil well drilling process involves more than 20 drilling activities whose sequential execution advances the drilling process. Recognition of oil well drilling activities is a crucial task as it allows for the identification of nonproductive time (NPT). Activity recognition is also essential as it can be part of a complete oil well drilling monitoring system. The SCADA data contains data that belong to multiple activities such as drilling or tripping. Symptoms of the anomalies differ during the execution of different activities. For anomalies, a framework is required which can identify the oil well drilling activities preferably in real-time and make them available to the anomaly detection modules

2. Stuck pipe problem is the most recurrent problem in oil well drilling process. It occurs when the drilling pipe is stuck, and the drill bit is not able to move while the drilling operations. Deviation in a single parameter present in the oil well drilling data does not confirm the occurrence of stuck pipe anomalies. The multiple drilling parameters can be considered as multivariate time series data recorded at a regular time interval. Continuous monitoring of the drilling parameters provides a better insight of the stuck pipe anomalies. The anomaly detection method can be used to finding the pattern that confirms the occurrence of stuck pipe anomalies. Performance of the anomaly detection method may deviate in the presence of different contexts. For example, type of soil is an essential factor that affects the performance of the anomaly detection models created to supervise the anomalies, i.e., stuck pipe complications in the oil well drilling process. The major challenge is to incorporate the contextual information in the anomaly detection model to detect the contextual anomalies that may occur during the oil well drilling process.

3. The available drilling data is of large volume (in this work we used the data of ten

rigs, and each rig has the drilling data more than one year approximately 8640000 data samples of dimension 20). The extensive SCADA data contain sensor errors and noise which requires prepossessing of the SCADA data before the development of supervised models. Further, the developed approaches would require to handle the uncertainties present in the data.

1.2 Contributions

- In the first contribution, a two-level classifier is proposed that combines the fuzzy rule-based and random forest classifier at stacked layer to identify the various oil well drilling activities from the real-time drilling data. The proposed model is efficient in identifying various oil well drilling activities for the drilling process.
- In the second contribution, a novel Fuzzy AdaBoost ensemble-based Dynamic naive Bayesian Classifier (DNBC) is proposed to identify the stuck pipe anomaly in the real time drilling data. In this method, a parameter, footprint of uncertainty (FOU), is initialized using the data's statistical properties that belong to the normal drilling operation.
- The behavior of the model developed to identify the stuck pipe anomalies is highly influenced by the presence of different types of soil and result in a high false alarm rate to identify the stuck pipe complications. To overcome this drawback, in the last contribution, a Contextual Dynamic Bayesian (CxDBN) is presented that is capable of identifying the contextual anomalies that occur during the oil well drilling process.

1.3 Organization of The Thesis

This section provides the details of the organization of the thesis.

- **Chapter 1**: *Introduction*

  This chapter presents the basics of anomalies and existing challenges related to anomaly detection in oil well drilling.

- **Chapter 2**: *Literature Review*

  This chapter presents the literature work on the anomaly detection techniques, and the later part of this chapter deals with the basics of the oil well drilling process for the extraction of oil and gas and explains various ML and AI-based techniques used to resolve the various oil wells drilling complications.

- **Chapter 3**: *Oil Well Drilling Activity Recognition* This chapter presents a novel two-level framework that stacks fuzzy rule-based classifier and random forest classifier to detect the numerous oil well drilling activities. Besides, we also provide a detailed description of the percentages of time the specific drilling activity is performed in one complete oil well drilling cycle. This work's primary objective is to provide the drilling activity label to the oil well drilling data sample that is further used to create the anomaly detection models.

- **Chapter 4**: *Dynamic Naive Bayesian Classifier and Fuzzy AdaBoost Technique* This chapter discusses a novel combination of dynamic naive Bayesian classifier [29] and fuzzy AdaBoost [43] technique where the footprint of uncertainty associated with the weights are initialized using the statistical properties of the data that belong to the normal class. The efficacy of the proposed anomaly detection method is shown using real data of stuck pipe anomalies.

- **Chapter 5**: *Contextual Anomaly Detection* This chapter presents a new framework that incorporates contextual information with the dynamic Bayesian network. The causal relationship between the state and the contextual information is used to design the dynamic Bayesian network. The proposed DBN network's performance is shown with the case study of contextual anomaly detection in the oil well drilling process.

- **Chapter 6**: *Conclusion and Future Works*

  This chapter discusses conclusion and future work related to the proposed work.

Chapter 2

Literature Review

The first part of this chapter provides a literature review of various anomaly detection techniques. The second part of this chapter provides a brief introduction of the oil well drilling process and presents various functional units of an oil well drilling rig. It also provides details of stuck pipe anomalies, their types and discusses various techniques that have been proposed to prevent the occurrence of the stuck pipe during the oil well drilling.

2.1 Anomaly Detection

Anomaly detection is a fundamental task of data mining and extensively used to identify the anomalies in real-world data produced by numerous applications. Early identification of the anomalies helps to make critical decisions for various applications. Detection of the outliers plays a vital role to extract useful information for multiple research areas such as fraud identification [158], [161], cybersecurity [18], and health supervision system [83]. The anomaly or outlier can be seen as the data point that shows significant deviation against the other data instances. The identification of the anomalies by inspecting the normal patterns is challenging because of the following reasons:

- Lack of a definition of a precise boundary to discriminate normal and abnormal patterns.
- Nature of the application to which anomaly detection is applied, the methods work

brilliantly in one field perform poorly in another domain.

- The presence of noise behaves the same as the anomalies.

The reasons mentioned above poses challenges for the anomaly detection methods to detect the outliers. Several anomaly detection methods have been proposed to address the issues described above while detecting the anomalies [104]. The type of anomaly, nature of the data, available class labels, and a computational cost are the significant issues that make the design of the anomaly detection method more challenging [11], [22], [42], [46].

Early identification of anomalies is essential as it provides useful information to make decisions. The increasing importance of the anomaly detection approaches in diverse research domains inspires the need for comprehensive and structured analysis of the anomaly detection methods. Various researchers have made significant contributions to the study and analysis of the anomaly detection methods and presented several surveys. Some of the significant survey papers on the anomaly detection can be found in [52], [168], [178], [200], [226], [237], [9], [98], [127], [95], [49], [89], [15]. Figure 2.1 shows classification of the anomaly detection techniques.

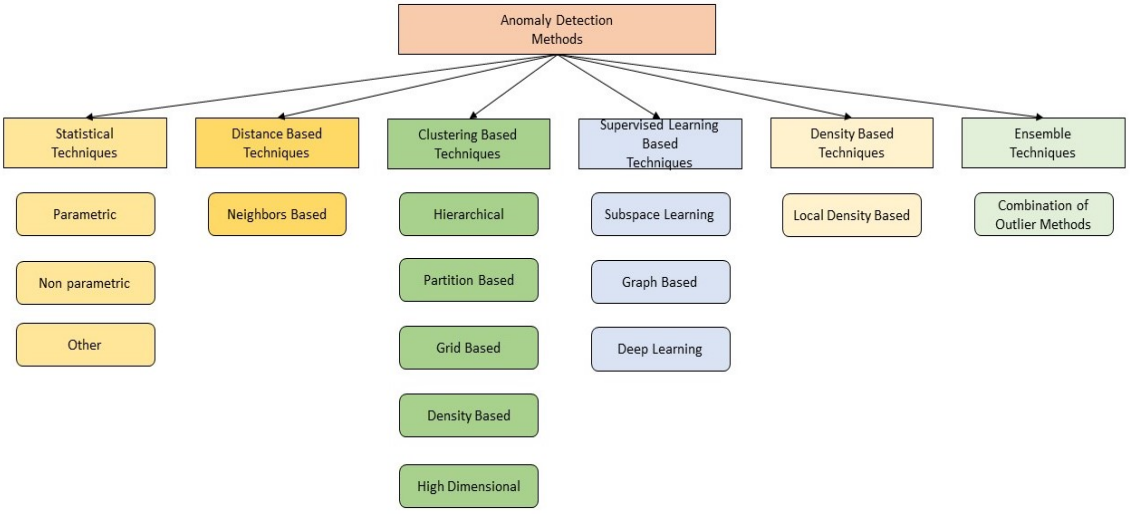


Figure 2.1: *Classification of anomaly detection methods*

2.2 Anomaly Detection Approaches

2.2.1 Supervised Learning-Based Methods

The learning-based approaches for anomaly detection have been successfully applied to detect the anomalies in diverse research domains. The learning-based methods can be subdivided into three major classes the deep learning methods, subspace learning-based methods, and graph-based methods. In the upcoming section, we will present the state of the art techniques that are successfully applied to detect the anomalies using the three techniques mentioned above.

2.2.1.1 Deep Learning Techniques

The deep learning methods have gained considerable attention from the research community. The significant reason behind the deep learning methods' success is its capability to learn highly discriminating features from the given data. Several attempts are made to apply the deep models for the identification anomalies [109], [114], [119], [127], [130], [141]. The recent survey of applying the deep learning models is presented in [49]. They presented an extensive survey of various deep learning models for anomaly detection and their effectiveness in detecting the anomalies. The application of the deep learning methods to detect the anomalies is effective due to the following reasons:

1. The deep models are capable of detecting the anomalies in large scale data.
2. The deep models have shown outstanding performance in extracting high discriminating features from the given data.
3. It provides better solutions to set the discriminating boundaries between normal and abnormal data.

Majorly the deep learning methods for anomaly detection can be sub-grouped into three categories:

The supervised deep learning requires training of binary classifiers using the training data that contains data that belong to the binary classes, i.e., the normal and abnormal

class. The supervised methods trained using the binary class data are well used to identify the outlier detection in the health care database [49]. Nevertheless, the popularity of the supervised deep learning methods to detect the anomaly is less than other versions of the deep methods as it suffers from the problem of availability of the training data and class imbalance issues. The semi-supervised deep learning methods can be easily trained using the data set containing large data instances of the normal class compared to the anomalous class data samples. The deep autoencoders are well suited for this task. The autoencoders are trained using the data that belong to the normal class and are further used to detect the anomalies.

The unsupervised deep learning techniques for anomaly detection mainly aims to use the informative feature of the data instances to provide the labels to the unlabelled dataset present to the unsupervised deep learning model. The autoencoder based model mainly computes reconstruction error to identify the anomalies. The autoencoder techniques are well in two ways to perform anomaly detection tasks. In a first way the autoencoders are used as feature extractor and second combined with other models to detect the anomalies [238], [54], [21], [70], [102]. In [65], author presented a Deeplog framework to identify the anomalies in system log. This method adopts Long Short-Term Memory (LSTM) in the framework, and the autoencoder provides encoding and decoding of the system log messages. In another work [39], the author proposed the anomaly detection method that utilizes the deep autoencoder to detect the anomalies for High Performance Computing (HPC). It was the first work proposed to identify the anomalies in supercomputer nodes. The autoencoders are trained on the normal class data and further applied to detect the anomalies. In various scenarios, the deep learning techniques are used to create the multi-level deep learning models that use the deep learning models like the autoencoders to extract the features from the data instances belong to the normal class, and extracted features are further used as input to the existing machine learning models like one-class support vector machine. However, these models' performance depends on the extracted features by the hidden layers of the deep learning method. Two effective approaches have been proposed Deep One-Class Classi-

fication [179] and One Class Neural Network [50] to overcome the issues related to the deep models. The One Class Neural Network uses the efficient feature representation capability of deep learning to create the decision function to separate the normal class data instance from the outliers.

2.2.1.2 Graph-Based Techniques

The graph is a popular tool to show relationship behavior and is widely applied to solve various research challenges. The graphs are capable of extracting and denoting inter-dependent relations of the data samples to detect the anomalies. In [15], the author presented a detailed and organized review of the anomaly detection methods that involve the graphical representation to identify the abnormal behaviors. They focused on studying the research challenges, performance analysis, and future scope of the state of the art methods. The first pioneer work to discover the anomalies using the graph-based technique was performed by Moonesingh et al. [148]. They proposed a random walk technique that models the interdependency of the data into an undirected graph and uses the Markov random walk to compute the outlier score of the data instance. In recent work, [212], the author combined the local neighborhood information with the graphical representation to detect the anomalies. The method also resolves the problem of a high false alarm rate present in the graph-based techniques. In [213], the same author extends the work mentioned above to detect the outliers by discovering the local information of the dataset using multiple neighborhood graphs, and at last random walk is used to compute the outlier score. However, the computational cost of the graph-based method makes them less accessible for the anomaly detection task, especially in the area that requires the timely detection of anomalies.

2.2.1.3 Subspace Based Techniques

The features used while detecting the anomalies play an essential role in identifying the abnormal behavior of the data instances. The majority of the existing anomaly detection methods involve consideration of all the features of the dataset to identify the anomalies. In [236], the author proposed a technique that includes a subset of the feature set to

locate the anomalies in the data set. They presented the comparisons to show that the performance achieved by the subset of features is comparable and sometimes better than the performance shown by the whole feature set. Considering the entire feature set may delay the process of outlier detection, so it will be an exciting direction to explore the subspace-based techniques to detect the anomalies.

Identification of the subspace and its application is widely explored to detect the anomalies in the various domains that generate high dimensional data. The primary aim of these methods is to identify the different sets of meaningful and efficient features to detect the outliers. The subspace-based approaches are further classified into two classes relevant subspace discovery [106], [149], [148] and sparse subspace discovery [227], [66]. The latter approach projects the high dimensional data into lower dimensions to discover the subspace capable of identifying the anomalies. However, the significant drawback of these methods is the high computational cost to learn the subspace. To overcome the complication of high computational cost in [11] author proposed a technique that uses an evolutionary approach to explore the subspaces that perform the detection of anomalies with the low computational cost, but the performance of this method highly influenced by the initial parameters used in the evolutionary method. In [227], the author used the concept of lattice to define the subspace relations. However, this method suffers from high computational cost and less efficient due to the lattice-based presentation of the relations. In the [66], the author used a high dimensional linear transformation method to detect the outliers, and the developed framework is named a sparse encoding framework. In [106], the relevant subspace-based technique is proposed to detect the outliers, and the results show the method is efficient to detect the outliers using the relevant subspace discover by the algorithm. The method inspects the correlation of the data instances from its neighbors and uses the same to define the proximity of the data instance while obtaining the outer score. In [149], the author discovered the subspace-based on a variance in between the features to detect the anomalies. The demand for the high computational cost makes these methods less popular to detect anomalies. The principal component analysis is another well-explored method to detect

the anomalies. This method also explores relevant subspaces to detect the normal and outliers data instances [125]. In [208], a hybrid approach is proposed that applied the Monte Carlo searching to discover the high contrast subspace, and later, the local outlier factor (LOF) in combination with the identified search subspace is applied to compute the outlier ranking of the data instances. Other methods that are using the relevant subspace-based outlier detection can be found in [118], [125]. The subspace learning-based methods' great performance is well known to solve various anomaly detection problems, but these methods suffer from high computational cost.

2.2.1.4 Discussion On Learning-Based Approaches

These methods offer less time complexity to detect anomalies. The models are trainable easily with less training data for anomaly detection. The Graph-Based techniques inspect the interconnectivity between the data samples using the graphs and use the learned representation to detect the outliers. The deep learning methods show the promising capability to learn the discriminating features from the data and further provides a feature extraction preprocessing unit for the traditional anomaly detection models like the one-class support vector machine (OCSVM). The deep model is more efficient in detecting the anomaly in large-scale datasets and learning the better representation of the decision boundary to discriminate the outliers and normal class data instances. But the subspace learning method suffers from the problem of high computational cost while identifying the subspace to detect the anomalies. The large scale dataset influences the deep learning-based anomaly detection framework's performance as they involve complex relationships to define the normal behavior of the data samples.

Still, a lack of precise representation of boundaries can discriminate the normal and abnormal data instances. Exploring the existence of the separation boundary would be an excellent research scope. A few reviews have been proposed for unsupervised anomaly detection using Recurrent Neural Network (RNN), Deep Belief Network (DBN). For a more detailed study of the deep learning-based anomaly detection techniques, we recommend the surveys provided by [49] and [127]. Exploring the direction of combining fuzzy methods with the deep learning methods to detect the anomalies would be another

future scope. The methods developed in combination with the fuzzy and deep models can handle the uncertainties present in the high dimensional time series.

2.2.2 Ensemble Learning Techniques

The ensemble learning techniques are well used to boost the performance of a specific machine learning method. The ensemble method performs a superior grouping of the numerous anomaly detection approaches such as the distance-based method, the density-based method, and the linear-model dependent techniques. The ensemble techniques are very successful in solving the classification and clustering challenges. The ensemble technique's significant advantage is that it removes the dependency of the model from the data set and combines different heterogeneous machine learning models to perform the desired task. Recently numerous methods have been proposed to achieve classification using Boosting [169] and Bagging methods [129]. Isolation Forest [133] has been submitted for parallel anomaly detection. Extreme Gradient Boosting [231] method was applied for the anomaly detection and, hybrid models are developed based on the Bagged Outlier Ensemble technique [170].

Very initial work to detect the anomalies using the ensemble methods was performed by Lazarevic et al. [129]. The feature bagging method technique was applied to model the high dimensional data. The model combines the decision of several anomaly detection models, and each model selects its random set of features from the available feature set to train the model. All the data samples are assigned the outlier score, and assigned outlier scores by the various anomaly detection methods are combined to generate the ensemble-based outlier score. From the experiments of Lazarevic et al. [129], it is clear that the ensemble methods outperform compared to individual anomaly detection techniques. The effect of change in the data distribution while detecting the outliers could be a significant future work of this technique. In [8], Agrawal et al. presented a survey of anomaly detection using the ensemble methods. The survey in [8], motivates some more pioneer studies [45], [120] of the ensemble techniques for the anomaly detection. In [8], the author presented various ensemble-based outlier detection methods and included the impact of analysis of these ensemble methods to detect the anomalies. The major works

were proposed in [45] and [129] to perform the ensemble-based classification. In [178], authors have explored multiple directions of the ensemble methods like an independent ensemble, data specific ensemble, the model-based ensemble, and sequential ensemble of the methods. They also presented the effect of the independent and dependent assumption between the methods while developing the ensemble techniques for the outlier detection. The boosting procedure creates dependency among the models to perform anomaly detection. The bagging technique assumes the independent characteristics among the anomaly detection models. The model-based ensemble technique contains the sequential and independent aspects. However, the assumption of dependent and independent highly depends on the nature of the application that generates the data. Other surveys [126], [187] present numerous challenges faced by the ensemble techniques used for the anomaly detection. They describe various problems associated with the comparison of anomaly scores using several functions and mixture models to combine the scores. In [187], the author proposed a similarity-based approach to compute the combined outlier score and ranking of the different methods. In [38], a novel framework was intended to combine the heterogeneous anomaly detection models to identify the high dimensional dataset anomalies. The framework enables the anomaly detection models to approximate the outlier score instead of repeatedly applying the same anomaly detection to compute the outlier score. The framework used to ensemble the outlier detection technique is named Heterogeneous Detector Ensemble on random Subspaces (HeDES). The model is efficient in increasing the anomaly detection accuracy in the high dimensional data set collected for real-world applications. One possible future direction of this work is to extend the experiments and analysis for massive and high dimensional datasets belong to numerous real-world challenges that involve anomaly detection tasks.

In 2013, Zimek et al. introduced a random sub-sampling methodology that computes the local density based on the nearest neighbors of the data sample. They created numerous data sets to train different anomaly detection methods. The random sub-sampling method assumes sampling without replacement policy to build the training datasets. The models trained based on the random sub-sampling show the high efficiency

to computes the outliers. The same author extended their previous work based on the learning mechanism. The data permutation approach is applied to perform the ensemble of numerous outlier detection methods. The approach majorly focuses on the estimation of the outlier score using distance and density estimation. Small noise is added to each of the feature values while creating the training data sets using the perturbation approach. The obtained training datasets are further used by the numerous anomaly detection methods to estimate the outlier score. The estimated outlier scores produce the combined score to provide the outlier ranking to the data samples. In [157], the author proposed a hybrid technique that combines the free bagging method and subsampling technique to train the different anomaly detection methods. The free bagging method selects different combinations of feature sets while building different data sets to train the various anomaly detection models. The subsampling method follows sampling without replacement policy while the sampling of the data instances. The feature bagging technique used in this method was failed to obtain the variance of the data samples, and the performance of the technique above depends on the data set created with the subsampling method. In recent work, Zhao et al. [232], proposed a framework that works in an unsupervised fashion and combines the outlier scores of the different anomaly detection methods. The proposed framework solves the challenge of aggregation of outlier scores and selecting outlier detection methods under the unified framework. The technique initially provides the label to the test data instances based on its K nearest neighbors' label. The framework then inspects the locality of the data sample using the K nearest neighbors to identify the best anomaly detection method. The same author extended the method mentioned above to design the four variations of the Locally Sensitive Combination in Parallel Outlier Ensembles Framework to detect the anomalies [233]. The book published by Agrawal et al. [12] can be used to understand more diversity of application of different techniques used to ensemble outlier methods. The book includes a detailed analysis of the numerous methods used to create the ensemble classifiers for outlier detection. A complete and comprehensive understanding of the various ensemble methods to detect the anomalies is presented in this book.

2.2.2.1 Discussion On Ensemble-based Anomaly Detection Techniques

The ensemble methods are robust and reduce the machine learning model's dependency on the type of dataset. The bagging and boosting techniques boost the anomaly detection techniques' performance by a large margin compared to the particular anomaly detection technique. These methods are well-suited techniques to detect the anomalies in the high dimensional dataset. The bagging methods select the best set of features to create the various anomaly detection models that are the further ensemble to perform effective anomaly detection. The ensemble methods can efficiently deal with factors like noise, fast computational processing, and quality of the data, thus performing better than the individual anomaly detection methods.

Poor development of the ensemble techniques to detect the anomalies and lack of the right strategy to evaluate the features of the ensemble. results in overfitting when applied to detect the anomalies in real-world datasets are the major drawbacks of the ensemble techniques.

The ensemble techniques have shown remarkable performance in dealing with the challenge of noise and fast computation by combining the different techniques into the single ensemble framework. However, there is still a scope to identify the strategy that can describe the best combination of the anomaly detection models. New approximate methods are required to perform the meaningful combination of the outlier ranking provided by various anomaly detection methods. Data locality is another open research issue that has been less investigated by the existing ensemble-based anomaly detection techniques. The data sample's locality needs to be considered while computing the outlier ranking of the data instance using the ensemble techniques.

2.2.3 Density-Based Techniques

The density-based techniques assume anomalies in the low-density region, and the non-anomalous data belong to high-density regions. The data instances present at the greater distance from their neighbors are flagged as anomalous data points. The mechanism used in the density-based anomaly detection methods is more complex than the distance-based

outlier detection techniques. The clarity and great performance of the density-based methods to detect the anomalies make them popular to detect the outliers in diverse research domains. Some of the significant contributions that have been made and serve as fundamental concept for the new density-based anomaly detection techniques can be found in [42], [113], [228], [31] and [201], [124], [155].

Initially, Breuning et al. [42] introduce the first density-based anomaly detection method named the Local Outler factor (LOF). The concept of the LOF score depends on the K nearest neighbors. The KNN of the data instance is utilized to compute local reachability distance (lrd). The lrd is computed as follows:

$$lrd(x) = \frac{1}{\frac{\sum_{q \epsilon KNN(x) reach-dist_K(x \leftarrow q)}}{|KNN(x)|}} \tag{2.1}$$

Where,

$$reach - dist_K(x \leftarrow q) = max(K - distance(q), d(x, q)) \tag{2.2}$$

$$LOF_k(x) = \frac{1}{|KNN(x)|} \sum_{q \epsilon KNN(x)} \frac{lrd_k(q)}{lrd_k(x)} \tag{2.3}$$

$lrd_k(q)$ and $lrd_k(x)$ are the local reachability distance of q and x respectively. x is the data instance, and q is the nearest neighbor. The major aim of the LOF is to inspect the clustering structure of the neighborhood to detect anomalies. The test data is said to be anomalous if the value of the lrd is low compared to its neighbors. The computational complexity of the LOf method is quadratic in time. Schubert et al. [120] proposed a method that repaced the lrd with the distance between the K nearest neighbors distance, and results show the method gained improvement in performance as compared to the LOF. In [202], a changing density-based method is proposed and named as a Connectivity-based Outlier Factor (COF) to identify the anomalies. The method is similar to the LOF, and the only step that differs is the computation of density estimation. In this method, the shortest path-based changing density is used to detect the outlier score. The method shows improved results as compared to the [42] and [188]. The major disadvantage of the COF method is the assumption of data distribution. The method

proposed a concept of isolativity that defines the connectivity of the data instance with the other data instances. The COF is computed as follows:

$$COF_k(x) = \frac{|N_{K(x)}|ac - dist_{N_{K(x)}(x)}}{\sum_{q \epsilon N_{K(x)}} ac - dist_{N_{K(q)}(q)}} \tag{2.4}$$

Where $(ac - dist_{N_{K(x)}})$ is a average distance of the input data x from the neighbors. The COF uses a minimum spanning tree (MST) of the KNN to define the density of the KNN. The computational complexity of the LOF and COF is quadratic time. The techniques mentioned above compute the outlier scores but failed to define the precise range of the outlier score to identify the threshold useful to detect the anomalies. In [124], the author introduced Local Outlier Probability (LoOP) that defines the outlier score in terms of probabilities. The probability aspect of this method arranges the outlier score in between 0 to 1. The accuracy of the LoOP is comparable with the LOF. This method provides a better comparison of the anomalous data in different datasets. The LoOP score is expressed as follows:

$$LoOP_L(O) = max\left\{0, erf\left(\frac{PLOF_{\lambda,L}(Q)}{nPLOF * \sqrt{2}}\right)\right\} \tag{2.5}$$

Where $PLOF_{\lambda,L}(Q)$ is the probabilistic LOF of the data instance, and $nPLOF * \sqrt{2}$ is the aggregated LOF. This method is different from the LOF as it assumes the distance from the nearest neighbor that follows the Gaussian distribution, and computed distance is termed as the probabilistic distance. After the calculation of the probabilistic distance, the same procedure as LOF is applied to calculate the anomaly score and normalization, and the Gaussian error function is used to convert the anomaly score into the probability values. The computational complexity of the LoOP is the same as the [188]. The LOF and COF failed to handle the issue of multi-granularity. To resolve this issue, Papadimitriou et al. [155] proposed Local Correlation Integral (LOCI) that uses a multi granularity deviation factor (MGDF) while computing the outliers. The outliers of the data instance are defined by three standard deviation distance from the MGDFs neighbors. This method is efficient in dealing with the variations in the local densities

of the feature space. The MDFE is computed as follows:

$$MDEF(x_j, R, \beta) = 1 - \frac{N(x_i, \beta_R)}{N(x_{j,R,\beta})} \tag{2.6}$$

Where $N(x_i, \beta_R)$ and $N(x_{j,R,\beta})$ are the number of β_R neighbourhood data instances and average of all the data instances x in the R neighbourhood of data instance x_j. The performance of all the techniques mentioned above depends on the selection of the number of nearest neighbors, and this issue was well resolved in LOCI. In the LOCI [155], a maximization technique was used to select the number of nearest neighbors. A similar assumption of Gaussian distribution made in the LoOP was used in LOCI to estimate the density. The method utilized to estimate the local density in LoOP depends on two sets of neighborhoods. The results show LOCI yields better performance as compared to the LOF and LoOP but suffers from the issue of long runtime. To remove the drawback of the LOCI mentioned earlier, Papadimitriou et al. [155] proposed an approximate LOCI to identify the anomalies.

Another anomaly detection technique that shows competitive behavior against the existing density-based anomaly detection methods was proposed by Ren et al. [173]. They introduced the Relative Density Factor (RDF) technique and employed the P-Tree to detect the anomalies. This method is more scalable than the other methods and less sensitive towards the increase in the size of the dataset. The value of RFD is high for the anomalous data. The value of RDF is computed as follows:

$$RDF(x, R) = \frac{DF_{neighbor}(x, R)}{DF(x, R)} \tag{2.7}$$

Where $DF(x, R)$ is the density ratio of the data instance x using its neighborhood data instances located at the distance of radius R and $DF_{neighbor}(x, R)$ is the density factor of the neighborhood of the x.

In [75], Jin et al. proposed an INFLUenced Outlierness (INFLO) that uses symmetric neighborhood dependencies to identify the local anomalies. The LOF suffers from an inaccurate space representation, resulting in incorrect anomaly detection in datasets containing nearly related clusters with varying densities. The INFLO resolves this drawback by assuming various descriptions of the nearest neighborhood set. The

set of K-nearest neighbors and a reverse set of nearest neighbors are used to calculate the INFLO anomaly score. The INFO anomaly score is computed as follows:

$$INFLO_K(x) = \frac{\sum_{q \epsilon LS_k(x)} Density(q)}{|lS_K(x)| Density(x)} \tag{2.8}$$

Where $Density(q)$ and $Density(x)$ are the densities of q and x , and $LS_K(x)$ is average density of data instance x.

In [47], a novel method was introduced that identifies the outliers in the uncertain data. The method is named as density-based outlier detection in uncertain data (UDLO). Instead of applying the naive method for computing the K nearest neighbors to calculate the outlier score, this method uses an exact algorithm to compute the nearest neighbors. This method employed Euclidean distance to identify the anomalies. Exploring the effects of other distance measurement techniques on the performance of the UDLO would be an excellent future scope. As mentioned earlier, various methods like LOF, LOCI, and INFLO were proposed to detect the outliers, but they share a similar drawback: the distance calculation in the high dimensional space. In [118], the author proposed a high contrast subspace method to detect the anomalies in the high dimensional dataset. Campello et al. [44] introduced a Global-Local Outliers score method that detects the global and local anomalies using a single framework. The method uses statistical computations to detect the anomalies. The results shown in [44] show that the method yields comparable results to detect the anomalies against the other existing methods. Momtaz et al. [143] proposed a Dynamic-Window Outler Factor (DWOF) to detect the top n anomalies. This method was motivated by the work proposed by Fan et al. [76], where they proposed the Resolution-based Outlier Factor (ROF) method to detect the anomalies. The DWOF resolves the shortcomings like the sensitivity of the parameters and low accuracy of the ROF method and yields better performance than the ROF.

The identification of anomalies in big data is a challenging task. Wu et al. proposed the RS-Forest method to detect the fast and accurate anomalies in the big data streams [216]. A parallel anomaly detection method was proposed in [31] and named Distributed LOF Computing (DLC). This method works in two-phase. The first phase is

grid partitioning, and the second phase in the local outlier factor computation. Lozano et al. [136] proposed a new method based on the parallel implementation of the LOF. In [201], Tang et al. applied the concept of KDE to detect the anomalies. They applied the KDE to estimate the local density and also employed the use of reverse nearest neighbors during the computation of the anomaly scores. The Euclidean distance was used while computing the outlier score.

Intending to reduce the quadratic computational cost imposed by numerous density-based anomaly detection techniques, Vazquez et al. [209] proposed an algorithm called Sparse Data Observers (SDO). This method yields comparable results against the existing anomaly detection methods with a low computational cost. The concept of relative density was used in [152] to detect the anomalies. EsDLOS [198] is an anomaly detection method proposed by Su et al. to detect the anomalies in a scattered dataset. They deducted the steps of the LOF and proposed a robust clustering technique that uses multiple queries to remove the non-anomalous data from the data used by the LOF to detect the anomalies. Thus, the size of data needs to be processed reduced by this method, which makes the computation fast compared to the LOF. The accuracy and running time of this algorithm beats the performance of the LOF.

2.2.3.1 Advantages of Density-Based Techniques

1. The density-based anomaly detection methods have shown excellent performance in identifying the local outliers. These methods do not require any prior assumption like the underlying data distribution and require tuning a single parameter to detect the anomalies.

2. These methods involve computation of the local neighborhood density that results in more accurate identification of the various anomalies missed by the other anomaly detection methods.

2.2.3.2 Disadvantages of Density-Based Techniques

1. The performance of these methods highly depends on the parameter K, i.e., the number of nearest neighbors.

2. The computational cost and time are high as compared to the statistical-based anomaly detection methods.

3. They performed poorly when tested to detect the outliers in the data streams due to change in the density over time.

4. The Computed outliers scores for the normal and anomalous data instances are nearby for the high dimensional dataset.

2.2.4 Clustering-based Techniques

The clustering-based anomaly detection techniques majorly depend on several clusters created by applying various clustering techniques. The created clusters are used to define the normal characteristics of data. In a popular choice, the clusters with a small number of data instances or smaller sizes are assumed as the outliers. It is required to note that anomaly detection and clustering are two different tasks and dissimilar to each other. The clustering method's primary aim is to detect the clusters, and the anomaly detection method focuses on identifying the anomalies. The performance of the clustering-based anomaly detection highly depends on the ability of the clustering method to detect the structural property of the clusters [17]. The clustering-based techniques do not require any labeled data and work unsupervised to detect the outliers. Different versions of the clustering-based methods to identify the anomalies are presented in [226]. As most clustering-based techniques do not belong to a current decade, we would like to guide the readers to the survey of [226] and not provide details of the working of these methods to detect the anomalies.

Among the various clustering-based anomaly detection methods [104], [52], [226], [89] proposed by various researchers, the two clustering algorithms stand on top ranking and widely applied to detect the anomalies. The first method is Denstream [46], and the second is D-Stream [55]. Both methods are a two-phased algorithm and apply density-based clustering to detect the abnormalities in an online and offline fashion. The initial step of the Denstream clustering records the summary of the data stream, and the later phase performs the clustering of data instances by using the summary recorded

previously. Potential micro clusters are utilized to detect the anomalies. The assigned weight is used to identify the anomalies by comparing it against the density threshold of the microclusters. Another popular clustering-based anomaly detection method is a CluStream [10], but the method requires more memory than the DenStream clustering to detect the anomalies.

Nevertheless, the clustering techniques suffer from the problem of the arbitrary shape of the cluster and lack of an adaptation of the dynamic characteristics of the data streams during the anomaly detection. The method proposed in [55] repeats the same initial procedure to detect the outliers as offered in [46] but differs in the clustering technique where it uses a grid partition-based approach to cluster the data stream. Identification of the anomalies is less complicated as compared to [46] and [10]. The grid-based methodology defines the noise using the dense and sparse grid used for the clustering step. The grids with less density than the defied threshold is considered as the anomaly. The Denstream and D-Stream both show better performance as compared to the CluStream. In [174], SDstream is proposed that detects the anomaly using sliding window technique in data streams. In [28], AnuOut was recommended to identify the abnormalities in the data streams, and this method is fast as compared to the methods mentioned above.

k-means clustering was applied in [68] to detect the outliers by processing the data streams using the window-based method. The results obtained in [68] were compared with the existing clustering-based anomaly detection techniques [23], [160], and the comparison shows the method is efficient as compared to the current techniques to detect the outliers in the data streams. The development of hybrid methods by combining the clustering and distance-based anomaly detection methods would be a major future research work. In [139], the K-means clustering is used along with weight assigning methods to detect the anomalies in the data streams. In [205], the author proposed a framework that assigns weights to the features while performing the clustering. The technique is compared with the LOF method and found it more efficient to detect the anomalies with a less computational cost. In [145], the author proposed the clustering-

based method to identify the abnormalities in data streams using an incremental version of the K means clustering. Bhosale et al. proposed a hybrid framework that combines partition-based clustering [151], [116], and density-based clustering to detect the outlier in the data streams. The results show that the hybrid framework yields better performance than [68]. The author also discussed using the proposed method to identify the anomalies in the datasets containing categorical and mixed attributes as future work. In [146], change in the mean and covariance matrix is incrementally updated to detect the anomalies. The data samples outside the cluster boundaries are considered as anomalies. The same author extended their work and proposed an elliptical fuzzy rules-based eTSAD method to identify the anomalies in the data streams [147]. In [181], the author presented the ensemble-based clustering technique for the data streams. The clusters created using the ensemble technique are further used to detect the anomalies. In [57], Chenaghlou et al. proposed an active clustering-based method to identify the outliers in the data streams. This method takes less memory and time to detect the anomaly. The sliding window was used to cluster the incoming data streams. In [175], the author applied the optimization technique to identify the outliers in the small and large clusters. In [58], Chenaghlou et al. extended their work in [57] to detect the anomalies in real-time data streams, and this method also models the sequential characteristics of the clusters. In [222], a Gibbs sampling technique is applied to detect the outliers in a text document. The following factors need to be inspected while using the clustering-based method to identify the anomalies:

1. How to define the anomaly when the data instance does not belong to any cluster or outside the clusters?

2. How to use the distance between the data sample and the cluster center to identify the anomalies.

3. How to define the anomaly based on small clusters created after the clustering method?

2.2.4.1 Advantages of Clustering-based techniques

1. They work in the unsupervised fashion and do not require any prior knowledge to detect the anomalies. The clusters are created from the data, and a new data sample is inserted to test the label of the data sample as normal or outlier. They work fast, so popular choice to detect the anomalies in the data streams.

2. The clustering-based techniques are robust towards the various data types. The hierarchical clustering method provides a nested partition of the data and provides a facility to use any partition level to perform the similarity measures to detect the anomalies.

3. The partition-based clustering techniques are scalable and simple, thus well applied to detect the anomalies in datasets that contain well separated compact spherical clusters.

2.2.4.2 Disadvantages of the Clustering-based Techniques

1. The majority of the clustering-based anomaly detection techniques require initialization of different parameters like the number of clusters, distance threshold, assumption of the shape of the clusters. The performance of these methods highly influenced by the initial values of the parameters.

2. The density-based clustering methods are sensitive towards the initialized parameters. The curse of dimensionality is another factor that causes high computation time and sometimes results in a high alarm rate to detect the outliers in the high dimensional dataset.

3. The cost of clustering is high in some of the hierarchical clustering-based anomaly detection applications [224], [85].

2.2.4.3 Research challenges

1. The selection of the appropriate width of the cluster and declaration of the distance of the data sample from the cluster center in case of the multivariate data is a

significant challenging area that still needs more research study.

2. Need for the hybrid framework to combine the various clustering techniques to overcome the drawbacks of individual clustering methods while identifying the anomalies. For example, the density-based clustering techniques can be combined with partition-based clustering to handle the noise.

3. Lack of the evaluation technique and the benchmark dataset to compare the performance of the various clustering-based anomaly detection techniques.

2.2.5 Statistical Methods

Anomaly detection using statistical approaches can employ supervised, unsupervised, and semi-supervised techniques. In the supervised methods, underlying distribution is assumed, and the learned parameters for the assumed distribution model are used to detect the anomalies. The statistical models are further classified into two major groups non-parametric and parametric models. The former technique does not require any assumption related to the distribution of the data samples. The parametric model assumes predefined distribution for the given data samples. In this survey, we will see some of the recent research contributions proposed to detect the anomalies using the statistical-based approaches. The method under Parametric Approaches mainly involves two types of model regression and Gaussian Mixture models.

2.2.5.1 Regression Techniques

The regression models are among the well-explored techniques to detect the anomalies. The selection of the regression model can be non-linear or linear and depends on the type of problem. Initially, the regression model is trained on the training data and later applied to the test data. The detection of anomaly is performed by measure the significant deviation of the test data from the expected output. Mixture models, Robust least-square [41], Vibrational Bayesian technique [226] are among the popular regression-based models used to identify the anomalies. In [186], the linear regression-based anomaly detection model was proposed by Satman et al. This method uses the

least trimmed square error method to detect the outliers. The method consumes less memory and time to detect the anomalies. In [156], the author proposed the regression-based method that uses a weighted summation technique to detect sensor data. In [61], the author proposed a survey of the various linear and non-linear models to detect the anomalies. The survey provides a performance comparison of different regression-based models using the receiver operating characteristics (ROC) curve. This survey shows that the performance of non-linear regression models is better than the linear regression models when applied to detect the outliers in real datasets.

2.2.5.2 Gaussian Mixture Model

The Gaussian Mixture models are well applied parametric statistical anomaly detection method that aims to identify the parameters such as the mean and variance of the GMM models by fitting the maximum likelihood estimation [38] technique on the training data. Later, the identified parameters are used for detecting the outlier score of the test data instances. In [219], the author proposed the unsupervised technique to identify the anomalies. This method applies the Exampler-Based Gaussian Mixture Model to learn the parameters from the training data. Unlike to methods proposed in [42], [202], [155] the technique proposed in [219] explores global properties of the dataset to detect the anomalies. The anomaly detection technique proposed in the [219] also shows robustness towards the noise present in the dataset. However, this method also suffers from high computational complexity to detect the outliers compared to other GMM-based techniques. In [203], Tang et al. proposed the hybrid model that combines the regression model and subspace learning method into a single framework to detect the anomalies. The method used locality projection-based subspace learning to detect the outliers and also shows better results as compared to the PCA –based method proposed in [180]. The technique used in [203] fills the disadvantage of the LOF and low density-based anomaly detection approaches by model the multi Gaussian states present in the dataset.

2.2.5.3 Non-Parametric Approaches

The kernel density estimation technique is the popular non-parametric anomaly detection method [159]. The unsupervised technique to detect the anomalies is presented in [128]. The anomaly is identified based on the local density estimation of the data instance, and the results are compared with the techniques proposed in [42], [155]. However, factors like high dimensionality and the massive size of the dataset majorly affect the performance of the proposed method. The variable kernel density estimation-based approach provides an automatic weight-based selection of the parameter K used in the LOF. In [184], the author proposed a novel method to identify the anomalies in the sensor nodes. In [37], the author applied the KDE method to detect the anomalies in the data streams. In [206], the anomalies in power grid-related applications are identified using the KDE technique.

A robust outlier detection methods were proposed in the [37] that uses Adaptive KDE to learn the probability distribution of the data, and further learned distribution is used to detect the anomalies. The results in the [37] show the method outperformed compared to the standard KDE method. The work of [37] can be further extended to detect the anomalies in the multivariate data streams. In [234], the KDE-based method is proposed to detect the anomalies in the multimedia networks. The non-parametric-based anomaly detection method was proposed by Smithy et al. in [196]. Zhang et al. [229], proposed the Gaussian kernel-based anomaly detection technique for the data streams. Unlike the methods mentioned above in the [163], a novel local anomaly semantic framework was proposed that uses the KDE to detect the anomalies in the extreme paced data streams. The methods mentioned above have shown the popularity of the KDE method to detect anomalies in various application areas, but the curse of dimensionality and high computational complexity is the drawback of the KDE-based method.

2.2.5.4 Alternate Statistical Techniques

Trimmed mean, Boxplot, Extreme Studentized test, histogram [10], Dixon text [211] are among the other popular statistical anomaly detection methods that have been proposed

to detect the anomalies. In [32] author discussed various optimization-based methods to detect the anomalies. In [90], a Histogram-based method was proposed to detect the outliers. In this paper, the author proposed two techniques, static and dynamic selection of bins, to define the outlier score of the data instances. The method is computationally efficient as compared to the COF [202], LOF [42], and INFLO [113]. However, the method lacks to detect local anomalies. A broad discussion on the statistical methods to detect the anomalies was suggested in [177]. In [103], Hido et al. proposed the statistical-based technique to detect the anomalies using the linear density estimation method. The ratio among the training and test data is used to compute the outlier ranking of the data instances. The results reported in the [103] show that the proposed statistical method performed better as compared to the KDE method and other existing methods. In [64], Du et al. proposed a novel Robust Local Outler Detection (RLOD) method for anomaly detection. They inspected the scope of detecting the global anomalies, unlike methods proposed in [82] and [26] that focus on detecting the local outliers. They presented a study of the sensitivity of the setting of the parameters used to detect the anomalies in the [82] and [26]. The framework used in [64] divides the anomaly detection procedure into three steps. In the first steps, the density peaks are identified using the three standard deviation rules. The data instances are assigned to the nearest clusters with maximum density. In the third and final step, Chebyshev inequality was applied to compute the outlier ranking. The running time and detection rate of the method are better than the techniques provided in [42], and [226]. The future score of this method is toward the detection of anomalies in the parallel and distributed systems.

2.2.5.5 Advantage of Statistical Approach

1. They assume the underlying probability distribution and learn the model's parameters using the training data. They are fast to compute and show improved performance when applied to detect anomalies.

2. The implementation complexity is low for the statistical models.

3. These models have shown excellent performance to model the real quantitative

values, ordinal values.

2.2.5.6 Disadvantage of Statistical Approach

1. They assume the underlying distribution of the data and learn the models' parameters for the given dataset. The performance of the model depends on the assumption and learned parameters.

2. The majority of the statistical-based anomaly detection methods are developed for the univariate data and suffer from the curse of dimensionality when applied to detect the anomalies in real-time multidimensional data.

3. The Histogram-based outlier methods are failed to model the interrelation between the multiple features of the multidimensional data and show poor performance when applied to detect the anomalies.

4. Shows Poor performance in identifying anomalies in the time series data.

2.2.5.7 Research Challenges

1. The majority of the statistical methods suffer from high computational cost, thus combining dimensionality reduction methods with the statistical methods to detect the anomalies would be an exciting research task.

2. The performance of these approaches depends on the availability of the anomaly free data. The collection of anomaly free data is itself a significant challenge, especially in real-world datasets.

2.2.6 Distance-Based Approaches

The distance-based anomaly detection is performed using the distance between the data instances. The data instance falls apart from the reachability of the nearest neighbors is considered as the anomalous data. KNN [62] is a primary and popular anomaly detection method that has been well used to detect the anomaly using the K nearest neighbor data instances. Different definitions of the distance-based anomaly detection

have been proposed, some of the popular definitions can be found in [22], [206], [122], [167]. The distance-based anomaly detection methods are more flexible and robust than the statistical-based methods and capable of dealing with the large dimensional data. Numerous distance-based outlier detection methods can be sub-categorized into three major classes pruning techniques, data stream techniques, and K-nearest neighbor dependent techniques.

2.2.6.1 K-Nearest Neighbor Approach

The K-nearest neighbor approach is a well-used approach to detect anomalies. The approach is different from the KNN classification. It first finds the K-nearest neighbor of the data instance, and then the local density of the data instance is defined in terms of the identified nearest neighbors. The primary aim of this method is to detect the global anomalies present in the dataset. The techniques under this category explore the neighborhood distance to identify the outliers. The first significant contribution to detecting the outliers using the K nearest neighbors is made in [121] and [167]. The proposed methods inspect large datasets to detect the anomalies. The non-parametric method proposed by Knorr et al. in the [121] has a similarity with the techniques proposed in [219] and [186]. The method suffers from high computational complexity, and in the [167], Ramaswamy et al. resolve the issue of the high computational cost using the cell-based approach. In the [122], a KD-tree, X-tree method was introduced to identify the top N anomalies in the data set. However, this method's performance degraded with an increase in feature dimensions. In [122], Angiulli et al. detect the top N outliers in the given input and provides weights to the training data instances of the dataset. For detecting the anomaly in the test data, the test input data instance weight is computed and compared against the weights of the nearest neighbors data instances present in the training data. A novel Recursive Binning and Re-Projection [87] were proposed by Ghoting et al. that deals with the drawbacks of the [121], [24]. This method enhances the computation speed of anomaly detection in the massive and high dimensional dataset. This method differs in terms of utilizing the nearest neighbors for the anomaly detection, and the method uses the approximate nearest neighbors to

compute the outliers. Likewise, the existing distance-based anomaly detection methods that focus on detecting the global anomalies in 2009 Zhang et al. [188] proposed a Local Distance Outlier Factor method to detect the local anomalies in the dataset using the distance between the nearest neighbors. This method shows improvement in performance to detect the anomalies compared to the traditional the LOF [42] technique. This method is less sensitive towards the selection of initialized parameters, and later Liu et al. [134] proposed a method to deal with the uncertainty of the LOF method. In [106], Huang et al. proposed a technique named a Rank-Based Detection Technique that ranks the nearest neighbors. The results show the method efficiently identified the anomalies in the high dimensional dataset. Bhattacharya et al. combine the rank of nearest neighbors and recursive set of nearest neighbors to identify the anomalies [36]. In [62], a method was proposed to detect the anomalies in the traffic data set. This method explores the relationship between the data instances present in the nearest neighbor set. The results show that the method yields better performance as compared to parametric methods like KDE and GMM. The significant drawback of the [62] is the assumption of a single distance metric to compute the outliers. The performance was miserable for the high dimensional dataset. In [214], Minimum Spanning Tree was used to detect the anomalies. Radovanovi'c et al. [166] proposed the reverse nearest neighbor set-based approach to detect the anomalies, and the method also tackles the problem of curse of dimensionality. The method is shown satisfactory performance to detect the anomalies in the small and large datasets. Different versions of the anomaly detection methods that utilize the concept of the nearest neighbors were proposed to detect the outliers. The performance of these methods depends on the selection of the number of nearest neighbors. The other K-nearest neighbor-based anomaly detection methods can be found in [97], [107] and [201].

2.2.6.2 Pruning Techniques

In [34], the author proposed a method that uses pruning rules and randomization to identify the anomalies in quadratic time complexity. Various assumptions related to the method leads to poor performance. In [24], Augillli et al. proposed the Detect-

ing Outliers Pushing method that resolves issues like CPU computation time, memory requirement, and minimization of input-output costs that remained untouched by the methods studied in the [122], [121], and [167]. The By-neighbor technique with pruning rules was proposed in [172] that computes vertical nearest neighbors to detect the anomalies. The method's performance was better than the method introduced by Ramaswamy et al. [172]. The method works in two steps and also uses P-Tree to detect the anomalies. The authors also suggested applying the P-Tree to detect the anomalies by combining the density-based anomaly detection method with the P-Tree. The other pruning rule-based anomaly detection methods can be found in [172] and [210].

2.2.6.3 Anomaly Detection in Data Streams

The extreme pace of data streams and requirement of quick computation are significant factors that make detection of the anomalies challenging in the data streams. In [193], the author studied the effect of the factor like concept drift, the uncertainty, the high dimensionality on the distance-based outlier detection for the data streams. The challenges above motivated the research community to design the anomaly detection methods for the data streams. The data streams are processed in the window-wise fashion and need fast computation as the arrival rate of the data streams imposes various difficulties in detecting the outliers [25]. In the [25] author introduced two types of technique to process the data streams window wise and landmark-based processing. The landmark technique processes the data between the last time point and the current time point. The windows–based technique considers the data between two data points of the considered window. In the [25], the author proposed a novel distance-based anomaly detection method that performed a one-time investigation of the data streams and named Stream outlier Miner (STORM). They proposed three query-based techniques to detect the anomalies. One technique performs the exact anomaly query, and the other two methods extract the approximate outcome of the asked query. The significant drawback of the exact query-based method is high memory utilization due to the storage of all the processed window objects. The approximate method resolves the issue mentioned above by applying approximate methods to reduce the utilization of the memory by the data

in the current window and its neighbors. The neighbor-based incremental detection of anomalies is performed by Yang et al. [218]. They proposed various methods like Exact-M, Extra-N, and Abstract-M, to detect the anomalies in the data streams. The method outperforms as compared to incremental DBSCAN [226]. Among the three mentioned methods, the Extra-N and Exact-M are based on density clusters, and the remaining method uses the concept of distance-based outlier detection. Event detection in data streams was performed by Kontaki et al. [123]. They focus on resolving the complications of the continuous event detection in data streams faced by the methods proposed in the [23] and [218]. In the [23], the author introduced the algorithms to reduce memory consumption. The methods are Micro Cluster-based Outlier Detection(MCOD), Continuous Outlier Detection(COD) and Advance Continuous Outlier Detection(ACOD). All the three method uses two major parameters the number of nearest neighbors and distance-based thresholding. The difference and comparison between these three methods are well explained in the summary table. The COD takes less memory space as compared to STORM and Abstract-C methods. The extreme-paced data stream was mined by Cao et al. [48] to detect the anomalies. In [200], Tamboli et al. presented a detailed comparative analysis of the distance-based outlier detection methods for the data streams.

2.2.6.4 Advantages of Distance-Based Approaches

1. They do not assume any underlying distribution of the data, thus easy to process and quick to detect the anomalies.

2. The scalability of these methods to model the multidimensional data is better than the statistical methods.

2.2.6.5 Disadvantages of Distance-Based Approaches

1. They also suffer from the curse of dimensionality same as the density-based and the statistical-based methods.

2. The use of KNN based neighborhood search for the high dimensional dataset is

expensive and time-consuming.

3. The majority of the distance-based method is failed to detect the anomalies in the data streams due to a lack of preserving the local distribution of the neighborhood data instances and computation of the KNN in the fast arrival data streams.

4. Show poor performance when applied to detect the anomalies in the time series data.

2.2.6.6 Research Challenges

1. Designing of the methods that can reduce the computational cost of the distance-based outlier detection methods.

2. Still a Requirement of a distance-based outlier detection methods for identifying the anomalies in the evolving data streams.

3. Identifying the local anomalies using the distance-based anomaly detection methods.

4. Incremental models are required to detect the anomalies in the univariate and multivariate data streams along with the low computational cost.

The previous sections presented literature on various anomaly detection techniques that have been proposed to identify the anomalies. The techniques mentioned above are well used to detect the anomalies in multivariate time series data by window-based processing. The features (statistical or temporal) are extracted for each window to detect the anomalies in the MTS.

2.3 Oil Well Drilling

The oil well drilling refers to creating a hole through rock and soil to make an entrance in geologic reservoirs that contain the gases and oil. The geologic reservoir includes the oil and gases that are extracted using the planned drilling process. The drilling process requires the construction of a circular section in the soil and rock, also known

as well. The place of existence of hydrocarbons majorly requires two types of drilling. The first type of drilling is known as onshore drilling that needs penetration of the rock and earth. The second type of drilling is offshore drilling that requires the drilling of the deep seabed. Figure 2.3 shows the onshore drilling oil well site, and Figure 2.2 shows the oil extraction using offshore drilling. This research mainly focuses on onshore drilling.

Figure 2.2: *Offshore drilling [1]*

Figure 2.3: *Onshore drilling [3]*

2.3.1 Onshore Drilling

The significant steps performed during oil well drilling are as follows:

1. **Seismic Survey:** Initially, a seismic survey is conducted to identify the potential reservoir locations luxurious for production of the oil and gases. This step creates a Geo-Technical Order (GTO) report that is further required for the oil well planning and design.

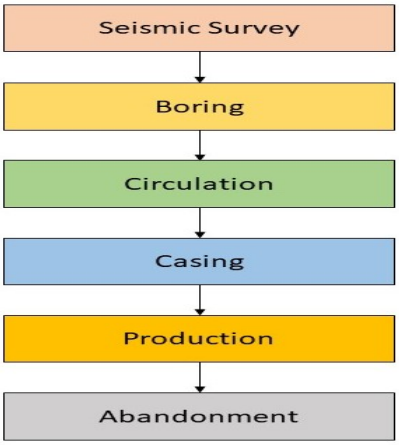


Figure 2.4: *Steps of oil well drilling*

2. **Boring:** Boring is a process of creation of the hole in the desired reservoir location. Drilling pipe and drill bit are involved in creating the hole in the rock. In the presence of a human population area near the reservoir, the boring is performed at an angle, and this process is known as directional drilling.

3. **Casing:** In this process, cementing of the created hole is performed after the drilling successfully reached the desired depth. The cementing process is used to prevent the soil layers from collapsing inside the shaped hole.

4. **Circulation:** The drilling mud flows inside the hole and back to the surface along with the soil cuttings produced during the drilling process. The circulation process also helps to maintain the desired pressure and temperature inside the oil well.

5. **Production:** In this step, the oil and natural gases are extracted from the reservoir.

6. **Abandonment:** After the successful extraction of the oil and gases, the drilled oil well is closed and safely abandoned.
   Figure 2.4 shows various steps of the oil well drilling process.

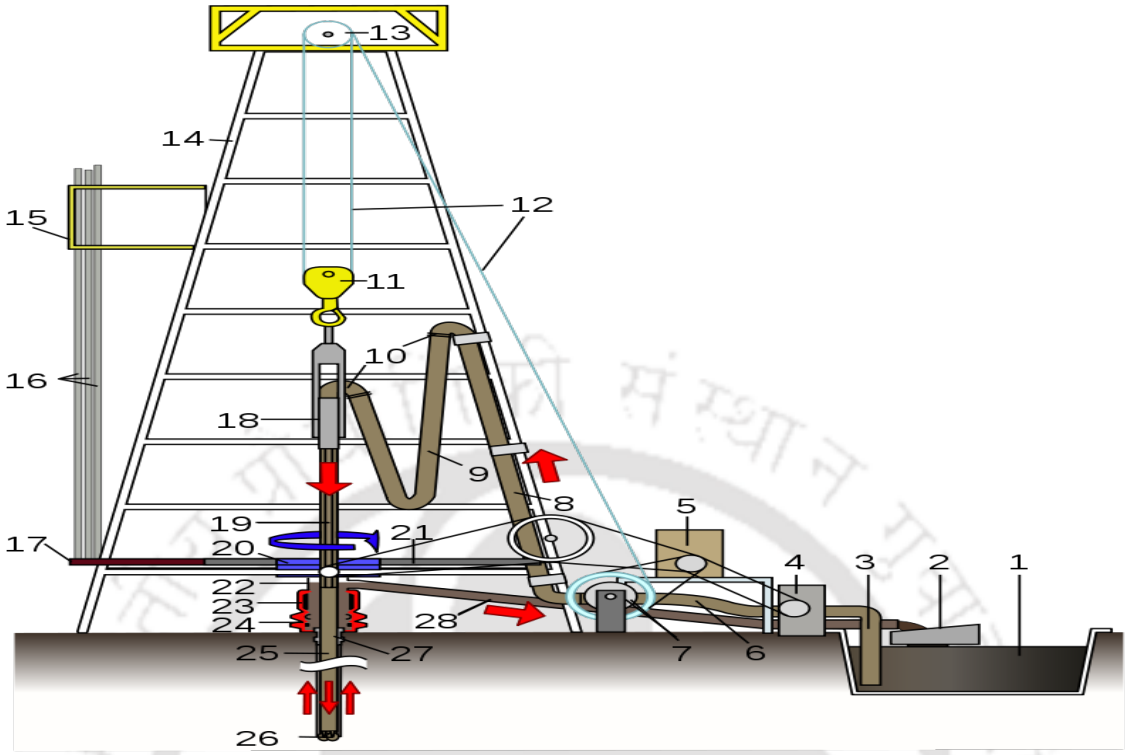


Figure 2.5: *Drilling rig [2]*

2.3.2 Drilling Rig

The rig is a mechanical framework that is used for the drilling process to make the hole in the rock and soil layers. Figure.2.5 shows typical structure of oil well drilling rig. Various working units of the rig are as follows [2]:

1. **Mud Tank :** This tank contains the drilling mud circulated during the drilling process to make the soil cuttings out from the hole.

2. **Shale Shakers :** This unit separates the soil cuttings from the drilling mud during the circulation.

3. **Suction Line :** This is a Connection of pipelines between the mud tank and mud pumps.

4. **Mud Pump :** A pump is used to insert the drilling fluid inside the borewell from

the mud tank.

5. **Power Source :** It supplies power to the mud pumps during the circulation.

6. **Hose :** It is a flexible tube used to transfer the mud from one location to another location.

7. **Draw-works :** This component is helpful for an up and down movement of a traveling block

8. **Standpipe :** A standpipe is a metallic pipe connected with the drilling pipe and useful for the circulation task.

9. **Kelly hose:** Kelly hose is the hose that connects the standpipe and Kelly.

10. **Goose-neck :** It is a connection joint between the standpipe and drilling pipes.

11. **Traveling block :** The traveling block is a free-moving part of the rig that contains pulley and drilling strings to move the drilling pipes up and down during the drilling process.

12. **Drill line :** It is a multi-threaded wire involved to make an upward and downward movement of the drilling pipes.

13. **Crown-block :** It is a permanent part of the traveling block that contains the pulley and the drilling string.

14. **Derrick :** It is a lifting device that contains supporting poles to lift the various components of the drilling rig.

15. **Racking Board:** This is a place where the drilling workers stand during removal or insertion of the drilling pipes.

16. **Stand :** It is a place where all the drilling pipes are collected for the drilling operations.

17. **Setback :** It is a place where the driller sits to conducts and monitors the drilling process.

18. **Swivel :** It is a mechanical device used to rotate the Kelly.

19. **Kelly drive:** It is an internal component of the rotary table that moves vertically during the drilling.

20. **Rotary table:** A mechanical framework of the drilling rig that rotates the drilling pipe while the drilling process.

21. **Drill floor:** This is named as a heart of the rig and used to perform the connection of various tools like drilling pipes and drilling bit.

22. **Bell nipple :** It is a large diameter pipe connected with a top section of a blowout preventer and facilitates mud circulation to the mud tanks.

23. **Blowout preventer :** It is a large valve to prevent control of the blowout situations generated during the drilling operations. The installed valve prevents the unwanted escape of gas and oil from the bore well.

24. **Blowout preventer pipe** : It is the pipe connected to the valve of the blowout preventer.

25. **Drill string :** These are the drilling pipes that transmit the drilling mud and connected sequentially to reach the desired drilling depth to perform the drilling. The drill bit is connected with the lowest drill string pipe and moves downwards while the drilling.

26. **Drill bit :** It is the mechanical tool designed to perform the hole in the deep soil layers during the drilling. The drill bit is located at the lowest part of the drilling pipes.

27. **Casing head :** It is the mechanical framework connected with the top of the hole to perform the cementing of the drilled hole.

28. **Flowline :** The pipe connects the bell nipple and shale shaker to facilitate the flow of the mud into the mud tanks.

The oil well drilling rig contains multiple functional units. Supervision of the drilling process is performed by monitoring various hydraulic and mechanical parameters gathered through the sensors attached to the different functional units of the rig. The data recorded with the sensors are stored in a Supervisory control and data acquisition (SCADA) system. The SCADA system is a database that stores the massive drilling data produced during the drilling process. The SCADA system also facilitates a graphical user interface (GUI) to supervise the different drilling parameters. The supervision of the drilling process requires continuous monitoring of the drilling parameters through the SCADA GUI interface. An alarm is triggered by the SCADA engineers whenever the drilling process makes an entrance into the problematic conditions.

2.4 Stuck Pipe Complication and Its Types

The stuck pipe complication is the most recurrent problem of oil well drilling. The stuck pipe problem causes enormous financial loss to oil industries [176]. Numerous studies denote the stuck pipe problems cost is higher than 250 million dollars per year [176]. The significant consequences of the stuck pipe problem include damage to drill string and complete loss of the oil well. According to shiver et al. [192], every fourth oil well rig struggled with a stuck pipe problem. A survey performed by British Petroleum that includes more than 700 well indicates a loss of more than $170,000$ dollars during 1985-1988 [40]. The effect of the stuck pipe problem varies according to the drill site. For example, a case study was reported in [192] in which a team of drillers and researchers conducted research with the aim to reduce the occurrence of the stuck pipe complications in the Gulf of Mexico and the North Sea during the 1980s. The team successfully identified the reason behind the appearance of the stuck pipe anomalies and reduced the causes of stuck pipe complications.

The stuck pipe problem is a state of the oil well rig in which the up or down or rotation of the drilling pipe gets restricted during the drilling process. The restriction

in the movement of the drilling pipe causes non-productive time (NPT) for the drilling process. In the literature, various reasons have been mentioned by the researches for the occurrence of the stuck pipe complications [108]. Some of the significant factors that lead the drilling process towards the stuck pipe complications are imprudent fluid properties, improper hole cleaning, and type of soil formation drilled [108]. The root causes of the stuck pipe problem can be applied to classify the stuck pipe complications into two types first is a mechanical stuck pipe, and the second is a differential stuck pipe.

2.4.1 Mechanical Stuck Pipe

Improper hole cleaning, essential seating, the collapse of the casing, plastic formation, and junk are among the significant factors that cause the mechanical stuck pipe complications during the drilling process. While the mechanical stuck pipe improper cleaning of the BHA causes the soil cuttings to contaminated over the drilling pipe and restrict the motion of the drilling pipe. Figure.2.6 shows the mechanical stuck pipe problem during the oil well drilling process. The mud filter cake created due to the contamination of the soil cutting restricts the drilling pipe rotation.

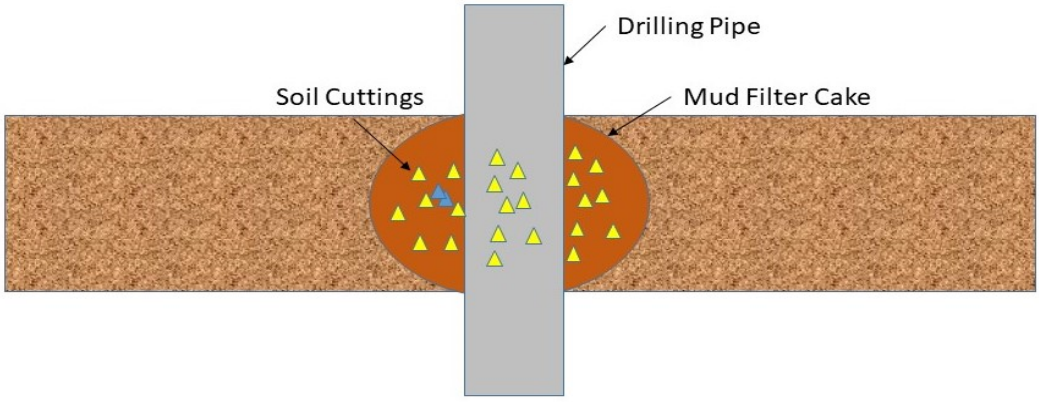


Figure 2.6: *Mechanical stuck pipe [111]*

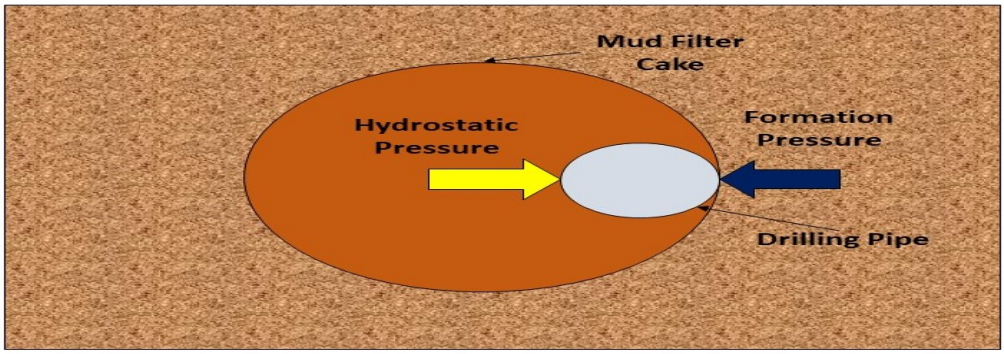


Figure 2.7: *Differential stuck pipe [111]*

2.4.2 Differential Stuck Pipe

As the name states, this complication arises due to the difference created between formation pressure and hydrostatic pressure inside the Bottom Hole Assembly (BHA). The mud inserted during the drilling process is absorbed by the soil formation, leading to the creation of mud filter cake around the drilling pipe. Due to pressure difference, the drilling pipe gets shifted towards one side of the wall, and the sticking of the drilling pipe with the wall restricted the motion of the drilling pipe. Figure. 2.7 shows a top view of the oil well drilling process also shows the shift in the drilling's location pipe towards the one side of the wall while the occurrence of the differential stuck pipe.

Proper oil well design, balanced mud design, and correct well planning are the other solutions that protect the oil well drilling against the stuck pipe complications. However, in [63] author addresses the issues where the stuck pipe happens though all aforementioned preventive measures have been used.

2.5 Methods to Avoid Stuck Pipe

Numerous techniques have been proposed to prevent the drilling process against the stuck pipe situations while drilling. Many of them result in inactive methods as these approaches were useful after the happening of the stuck pipe problem. But the majority of procedures are capable of protecting against the stuck up. The preventive mechanisms that have been proposed to prevent the stuck pipe anomalies in oil well drilling are

discussed in the latter part of this chapter. The duration of the stuck pipe complication is unpredictable. The first action performed as a remedy for the stuck up is an idea to free the stuck up as quickly as possible. The most general practice to release the stuck pipe is firing the jars [99]. The stuck up zone of the BHA is identified to release the stuck up, and the jar is moved upward or downward to release the drilling pipe. The jar-based method's application fails when the stuck-up zone in the BHA is miss identified. Another disadvantage of the jar based approach is the duration of jar based method as it increases the NPT in the oil well drilling [91]. Another technique as an alternative to the jar method was a resonant vibration method that takes less time to free the stuck up condition [91]. The resonant process's significant advantage is it is operated from the surface and does not require any intervention of the stuck pipe zone [91]. An additional method well applied to cure the differential stuck pipe is a continuous supply of chemical pills inside the BHA. These pills are inserted inside the BHA to soak the mud filter cake that causes the stuck pipe, but later, the use of the pills was banned due to its toxic effects on the surrounding environment when applied during the offshore drilling operations [100]. In [91], the author proposed a method that uses chelation fluid to crack the mud filter cake inside the BHA. All the techniques above need the use of the pills to diminish the effect of filter cake on the movement of the drilling pipe. The pills based method are extensively used to free the stuck pipe complication and attracted the attention of researchers from the oil industries. Design of the drill collars plays a vital role in preventing the risk of stuck pipe problems. The well-designed drill collar reduces a contact area between the wall and the drilling pipe, resulting in less occurrence of stuck up situations [7]. In [60], the author has made a significant contribution by designing a computer-based monitoring tool named Drilling Adviser that provides information about the stuck-up and subsequent events. This system takes manually entered data from the drillers to identify the complications. The prime objective of this tool was to assist the drillers during the drilling operations. In 1983 Love et al. proposed a new factor, "Stickiness Factor," that indicates a probability of freeing the stuck-up situations [135], and the model was tested in the Gulf of Mexico to identify the stuck

pipe complications. When all the methods above failed to free the stuck up situations backing off operations were performed to release the stuck up [171]. Pipe stretching computations [154] and Free point Logs [197] are the significant activities involved in freeing the drilling pipe. However, in case of failure of the backing off procedure, the pipes are cut, and a fishing [105], or a sidetracking [150] is performed to release the stuck cases. The fishing and sidetracking take several days to free the stuck up situations, thus result in increases of the NPT for the drilling.

In the previous section, we presented the most common approaches performed to free the drilling pipe during the stuck pipe conditions. Nevertheless, most of the methods mentioned above were performed after the occurrence of the stuck pipe complication. It is better to predict the occurrence stuck pipe situation ahead and take preventive measures to avoid the stuck pipe instead of applying the methods to free the drilling pipes after the happening of the stuck pipe [144]. Numerous methods have been proposed to identify the stuck pipe situation during the drilling using computational intelligence and statistical models. In the upcoming sections, we will see the approaches that have been submitted to determine the stuck pipe anomalies in oil well drilling using machine learning-based techniques.

2.6 A Review of Stuck Pipe Identification Approaches

The approaches to identify the stuck pipe anomalies can be sub-divided into two groups. The first groups includes the approaches that use statistical analysis-based techniques to identify the stuck pipe conditions during drilling. The latter group consists of machine learning techniques such as artificial neural networks, support vector machine, case-based reasoning, rule-based classifiers to detect the stuck pipe anomalies.

2.6.1 Statistical Approaches To Recognize Stuck Pipe Anomalies

At first, Hemkins et al. in 1987 used Multivariate Analysis to predict the stuck pipe complications for the drilling process [101]. They mentioned 131 cases of the stuck pipe in the Gulf of Mexico between 1981 to 1984. This study requires a total of 20 drilling

parameters to identify the stuck pipe anomalies. The relationship between the multiple drilling parameters was studied and utilized to develop the model to detect the stuck pipe complications. The method reported the success of 81-87 %. Howard et al. followed the same technique to determine the stuck pipe cases in the Gulf of Mexico. However, they collected the drilling data of more than 1000 wells with the 40 drilling parameters and identified the stuck pipe cases. The success rate of this method was 75%. Biegler et al. proposed their approach to identifying the stuck pipe-cases based on multivariate statistical analysis. In this approach, they inspected the combinations of physical and drilling parameters to build the model. The model proposed by Beigler et al. has two advantages. First, the model was capable of predicting the stuck pipe anomalies in the drilling data, and second, the model also reveals the root cause behind the stuck pipe's occurrence. However, the model developed by Beigler et al. has limitations, such as the model can identify the stuck pipe-cases only in the presence of water-based mud. In 2011 Shoraka et al. proposed a method that includes regression and discriminant analysis. The regression model identified the stuck pipe anomalies, while the discriminant analysis was used to determine the set of drilling parameters that are useful to build the anomaly detection model. The discriminant analysis based model correctly predicted the occurrence of the stuck pipe-cases before the actual appearance of the stuck pipe however, the regression model predicted the same before the one day. In [183], the author proposed a novel framework that measures deviation in the drilling parameters and assigned a weight to each of the variations predicted for each window. The predicted difference in the drilling parameters' values, the torque, the ROP, and the HL is combined to identify the stuck pipe anomalies. All the methods described above require optimization of drilling parameters to predict the transition of the drilling process from the regular drilling to the stuck pipe situations. The optimization process requires the computations of various parameters related to the mud properties that take a long duration if needed in real-time.

2.6.2 Recognition of Stuck Pipe Using Supervised Machine Learning Approaches

The methods mentioned previously require optimization of various mud parameters to detect the stuck pipe cases, and the performance of the models depends on the appropriate selection of the different drilling parameters. In [194], Pall et al. developed a knowledge database named as Case-Based Reasoning (CBR) system. The CBR system contains 50 different cases of drilling problems relevant to the oil well drilling process. All the cases were collected for the drilling of the North Sea drilling operations. Any new oil well-drilling cases are matched against the available cases in the CBR system to identify the causes and type of drilling problem. The CBR model's small size and static nature make it less popular to detect the stuck pipe problems for the oil well drilling. In 2001, Ali et al. proposed expert-system decision trees that identify the lost circulation index and formation damage factor to determine the stuck pipe problems in oil well drilling. The proposed expert-system decision tree uses fuzzy logic to deal with drilling problems that contain partial truth [80]. In [81], an underbalanced drilling technology (UBD) was proposed to screen the optimal UBD technique for the drilling procedure. This work was an extension of the previous work performed in the [80] and capable of identifying the stuck pipe anomalies. In [138], Robert et al. proposed a recommendation framework that predicts the rig's future behavior and provides available solutions to prevent the stuck pipe complications. The artificial neural network (ANN) was trained using the drilling and mud parameters to predict the dynamic behavior of the drilling process. In [207], the author proposed a framework that combines ANN and the CBR approach to identify the deviation in different drilling parameters using the ANN and later match the stuck pipe complications cases against the cases previously stored in the CBR. In 2009, a hybrid method was developed by Roar et al. to identify the stuck pipe problem during the drilling process. The first principle methods and AI techniques are integrated to detect the drilling complications [153]. In [53], the drilling parameters are optimized, and existing drilling monitoring software like Drill Edge is used to perform the case studies of the occurrence of the stuck pipe in the Gulf of Mexico. Wang et

al. [112] proposed an adaptive neuro-fuzzy inference system (ANFIS) based hierarchical framework that generates the alarm by monitoring the deviation between the predicted and actual values of the drilling parameters. Kadir et al. proposed the statistical-based method that computes the outliers in the drilling data based on the mean and standard deviation of the data samples. The statistical method proposed by Kadir et al. successfully identified the stuck pipe complication by monitoring the strokes per minute (SPM) parameter [117]. In the same year chamkalani et al. [51] proposed a hybrid model that combines support vector machine (SVM) and simulated annealing to detect the stuck pipe cases. Gundersen et al. proposed a decision framework that uses a fuzzy rule-based (FRB) classifier and the CBR system to identify the earlier symptoms of the stuck pipe anomalies [94]. Skalle et al. created the CBR systems that check the deviation in the drilling parameters of the SCADA system, and the deviation is checked against the predefined cases of the stuck pipe in the CBR to alert the stuck pipe complications [195]. Bach et al. [30] have applied the clustering technique to identify the predefined cases of the stuck pipe stored in the CBR and claimed that the proposed method is less expansive in terms of time complexity when compared with the traditional CBR system. Goebel et al. proposed an ensemble framework that combines the decision of the ANN, the SVM, a Bayesian Network (BN), and a Rule-based classifier to identify the occurrence of the stuck pipe cases for the drilling process [88]. Gundersen et al. proposed the decision support system to identify the stuck pipe complications during the drilling process. The proposed decision support system contains the CBR system to identify the stuck pipe anomaly symptoms using the deviation in the drilling parameters [93]. In [74], the statistical feature from the drilling parameters are extracted, and the AI-based models SVM, ANN, random forest (RF) were trained and further used to identify the stuck pipe complications. The method above assumes the drilling parameter as the time series data. In [182], the author identified percentages of change in deviation in the values of the drilling parameters during the different oil well drilling activities, and changes in the deviation are further utilized to detect the stuck pipe complications. They presented two case studies where they successfully identify the occurrence of the stuck pipe while

the drilling operations. Zhao et al. proposed a framework that considers the drilling parameters as time series and extracts features from the drilling parameters using the SAX technique. The SAX-based extracted features are clustered and later used by Dynamic Time Wrapping (DTW) to identify the occurrence of the stuck pipe anomalies [230]. Ambrus et al. integrated the drilling parameters and the contextual information like soil properties and the mud properties in the BN to identify the probability of occurrence of the stuck pipe during the drilling process [20]. Gharbi et al. inspected the use of metaheuristic approaches to improve the quality of the raw data received for the drilling operation. The proposed model removes the noise present in the drilling data and improves the performance of the AI-based model to predict the stuck pipe cases. The deviation in the value of the Hookload parameter was predicted to identify the symptoms of the stuck pipe complication [16]. Priyadarshy et al. proposed GUI-based framework that computes an average value of the drilling parameter and displays the deviation in the values of the drilling parameter in a circular GUI interface. This interface also displays the external drilling parameters' different external values like the mud properties, and plasticity [162]. In [221], the author combined the mathematical model and ANFIS model to predict the change in the rate of penetration (ROP) parameter. The change in the ROP parameter is further helpful to identify the stuck pipe-cases using the mathematical model. In [69], the author used the support vector regression (SVR) to predict the value of the ROP parameter helpful to identify the poor hole cleaning that causes the stuck pipe problems. Li et al. [131] designed an Online Sequential Extreme Learning Machine (SELM) that does not require the historical drilling data to identify the stuck pipe cases. The online data received by the SCADA system was used to update the model on the fly. The model outperforms compared to the traditional SVM classifier widely used by the various researchers to detect the stuck pipe cases. Alshaikh et al. proposed an automation tool that identifies the stuck pipe anomalies during the drilling process. The proposed automation tool integrated a physics-based method and the CBR approach to detect the earlier symptoms of the stuck pipe problem [19]. In [137], the author used a Deep Neural Network (DNN) based approach to extract different drilling

inputs available in the Daily Drilling Report (DDR). The proposed method applied text mining to extract the DDR information and extracted information as the input parameters to the stuck pipe detection models like the SVM and ANN. Gurina et al. proposed the anomaly detection framework that computes similarity scores between the drilling data processed window-wise. Moreover, the feature extracted from the multiple drilling parameters is used as an input to the decision tree (DT) to identify the stuck pipe complications [96]. In [13], the author predicted the rock and fluid properties using Fuzzy Logic and Neural Network, and identified properties were further used to detect the stuck pipe anomalies using AI and ML techniques. In [84], an ensemble framework is proposed based on the RF classier, the SVM Classifier, and the logistic regression. The decision of these models was combined to predict the symptoms of the lost circulation that cause the stuck pipe anomaly. The ensemble classifier is trained using the parameters of the seismic data collected for the rig. Abbas et al. used the SVM and ANN models trained using the drilling parameters and lithology information to identify the lost circulation problem. The models were trained using the data of 385 rigs and successfully tested to detect the symptoms of lost circulation problem that leads to stuck pipe conditions in the Iranian oil fields [6], [5]. Abrar et al. trained the SVM and ANN models using the drilling data of the oil well rigs to identify the stuck pipe complications. They performed the ten-fold cross-validation to test the robustness of the developed models. In [140], the author proposed a novel framework for earlier detection of the stuck pipe anomalies. The model is trained using the drilling and rheology parameters. The training data used to train the model contains the data belong to the normal drilling operations and the stuck pipe cases.

2.7 Summary

This chapter started with the literature review, advantages, disadvantages, and research challenges associated with the various anomaly detection methods. The later part of this chapter discussed the basics of oil well drilling. Then it presented the rig and various working units of the rig to conduct the oil well drilling process. In continuation, it

Table 2.1: *Summary of Deep Learning Techniques for Anomaly Detection Techniques*

Method	
	Advantages
Deep Learning Techniques [21], [54], [70], [102], [109], [114], [119], [127], [130], [141] [238]	1. Learn more representative features for classification.
	2.Efficiently captures a relationship between multiple features.
	3.Automatically extracts features from given input data.
	4.Generative models can generate synthetic data samples from anomalous class
	5.Efficient in handling temporal characteristics of time series.
	Disadvantages
	1. Deep Learning models are less interpretable
	2. Valunarable to adversarial attacks.
	3. Demands huge amount to labeled data for training.
	4.Demand training of high number of parameters.
	5. Require re-training on arrival of data samples follow unseen distribution.

Table 2.2: *Summary of Graph-Based Techniques for Anomaly Detection Techniques*

Methods	
	Advantages
Graph-Based Techniques [15], [148], [212], [213]	1.Encodes relationship between data samples in a form of graph.
	2.Identifies anomaly by exploring neighborhoods.
	3.Uses graph-clustering techniques to identify the anomalies in large dataset.
	4. Uses undirected graphs and random walk to identify outier score of a data sample.
	Disadvantages
	1. Suffer from a high computational cost.
	2. Less efficient to identify anomalies in data streams.
	3. Poor performance to identify outliers in time series.

presented the stuck pipe problem and its types. Finally, a literature review of different approaches such as the manual approaches, the statistical techniques, and computational intelligence-based approaches to identify the stuck pipe complication during the oil well drilling process are discussed. From the literature, it is found the research challenges associated with supervision of the drilling process requires the techniques that are effi-

Table 2.3: *Summary of Other Supervised Anomaly Detection Methods*

Method	
	Advantages
Ensemble Techniques [45], [120], [126], [129], [187], [233] Statistical Techniques [37], [42], [155], [196], [202], [206], [226], [229]	1.Ensemble techniques develop more robust classifiers for anomaly detection.
	2.Bagging and boosting techniques are employed to develop ensemble models.
	3.Ensemble model provides a unified framework that combine multiple anomaly detection methods.
	4.Low computational complexity to implement statistical models.
	5 Statistical methods are computationally fast compared to unsupervised techniques..
	Disadvantages
	1. Both techniques suffer from curse of dimensionality .
	2. Poor performance to identify anomalies in time series.
	3. Additional time latency during ensemble of different models to identify the outliers.
	4. Data locality is unexplored during computation of outlier scores.
	5. Performance of statistical model highly depends on assumed underlying data distribution.

Table 2.4: *Summary of Su-Space Techniques for Anomaly Detection Techniques*

Method	
	Advantages
Sub-Space Techniques [66], [106], [118], [125], [148], [149], [227],	1.Projection of high dimensional data into lower dimension.
	2.Some of the techniques select sub-set of features to detect the anomalies.
	3.Sparse encoding framework to detect the anomalies..
	4.Explore variance-based sub-space to detect the anomalies.
	5.Principal component analysis to detect the anomalies.
	Disadvantages
	1. Suffer from high computational cost.
	2. Poor performance to identify anomalies in time series.
	3. Retraining is required on arrival of unseen data.
	4. Less interpretable models.

cient to deal with temporal characteristics and uncertainty present in the drilling data and at the same time capable of incorporating the expert drilling knowledge with the developed supervisory models. Table 2.1, Table 2.2, Table 2.3, Table 2.4, and Table 2.5 show a brief summary of deep learning, graph-based techniques, ensemble and statisti-

Table 2.5: *Summary of Unsupervised Anomaly Detection Techniques*

Method	
	Advantages
Density-Based Techniques [31], [42], [113], [124], [155], [201], [228] Clustering Techniques [10], [52], [68], [89], [104], [226] Distance-Based Techniques [22], [122], [167], [206].	1.Do not assume a prior underlying distribution of data for anomaly detection.
	2.Incorporate locality information while detection of outliers.
	3.Clustering methods works in an unsupervised fashion.
	4.Require initialization of less parameters.
	5.Computation of outlier scores demands calculation of neighborhood density.
	Disadvantages
	1. Suffer from curse of dimensionality.
	2. Poor performance to identify anomalies in time series.
	3. Cost of clustering is high for some techniques.
	4. Non-interpretable models.
	5. Outlier score depends on choice on parameter K (number of nearest neighbors)

cal techniques, sub-space learning techniques, and unsupervised techniques for anomaly detection.

Chapter 3

Oil Well Drilling Activity Recognition

Recognition of oil well drilling activities is a crucial task as it allows for identification of the nonproductive time (NPT). The activity recognition is also important as it can be part of the complete oil well monitoring system. This chapter presents a novel two-level classifier that consists of Fuzzy Rule-based (FRB) and Random Forest (RF) Classifiers to recognize different drilling activities during the oil well drilling process. The novel two-layered classifier is designed by stacking the FRB and RF classifiers to achieve classification of the various drilling activities with high accuracy. The evaluation of the proposed method is performed with real drilling data from oil well rigs in the state of Assam, India. Further the proposed classifier generates an accurate report of time spent in executing different drilling activities in a complete cycle of the oil well drilling process. Empirical evaluation of the real drilling data shows the efficiency of the proposed method.

3.1 Related Work

Some of the significant contributions that address the issue of identification of drilling activities in this section are, in [191] author proposed a method based on support vector machine to classify the drilling activities. In [189] drilling activities are broadly categorized into six drilling activities using ant colony method. In [204], drilling expert knowledge is transformed into rules and developed rules are applied to identify the drilling activities in a mud logging data. In [190], the artificial immune-based classifier is employed to determine the drilling activities. In [71], [75], statistical properties of

the drilling data are investigated and used as a feature to classify the drilling activities. In [73] trend and value pair features is proposed to identify various drilling activities. In [27] a distributed approach is introduced to detect multiple states of the drilling operation. In [72] a hidden Markov model was trained to memorize the last drilling activity and used to classify the drilling activities in the drilling data.

The major shortcomings of the above mentioned techniques are: first, most of the algorithms process the incoming data in windows (or data frames) for developing the classifier model and also for final classification. Thus, the accuracy of the algorithms are sensitive towards the selection of an appropriate size of the window. Second, the methods were trained over a small data set, which do not capture the different situations that may arise during drilling, and never been tested for the massive real drilling data. Third, most of the methods have limited themselves in identifying three main drilling activities, namely, drilling, tripping, and circulation but did not take into account many other activities such as drilling with or without rotation and tripping in with or without rotation. Due to these requirements, there is a need of new approaches that can efficiently detect various kinds of drilling activities.

3.2 Preliminaries

This section presents a description of various drilling activities of the oil well drilling process. The latter part of this section introduces the basics of the FRB and RF classifier.

The drilling process is a sequential arrangement of different drilling activities and execution of these drilling activities advances the drilling process. The drilling expert manages duration of execution of different drilling process.

3.2.1 Oil Well Drilling Activities

1. **Drilling with rotation (DRWR) (Vertical drilling):** This drilling activity is performed for the vertical wells. In this activity, drill bit advances downwards in a straight path, thus the Bit depth and Total depth increases and the drill string pipes are rotating continuously. Figure 3.1 shows the DRWR activity.

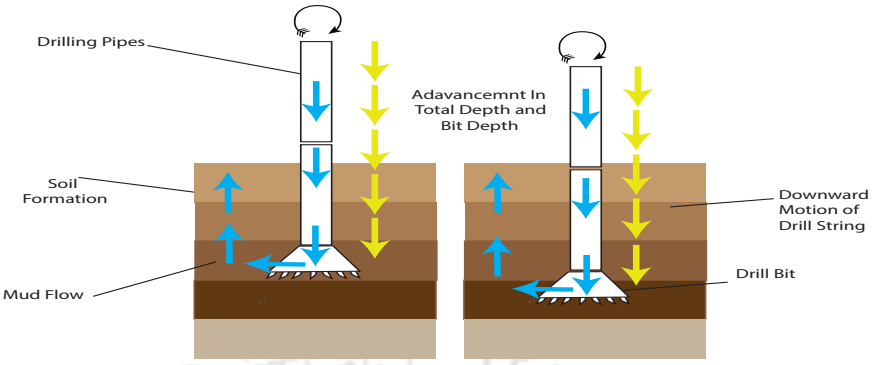


Figure 3.1: *Drilling with rotation*

2. **Drilling without rotation (DRWO) (Horizontal drilling):** During this activity, the drilling is performed by moving the drill string pipes at a path inclined with a predefined angle. The Total depth and Bit depth remain increasing, and rotation in the drill string is zero. Figure3.2 shows the DRWO activity in the drilling rig.

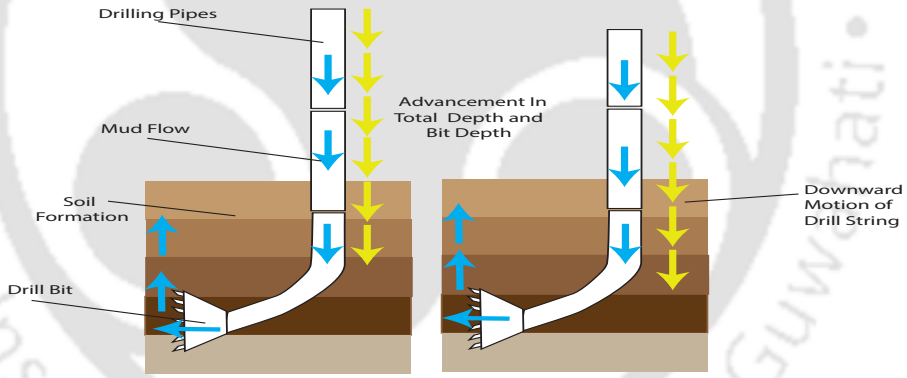


Figure 3.2: *Drilling without rotation*

3. **Tripping out with rotation (TROWR) (Back reaming):** In this activity, the drill string pipes are taken out from the bore well by rotating the drill string pipes continuously. The Bit depth decreases and the Total depth remains constant during this operation. This operation is also known as back reaming. Figure 3.3 denotes the TROWR activity in the drilling rig.

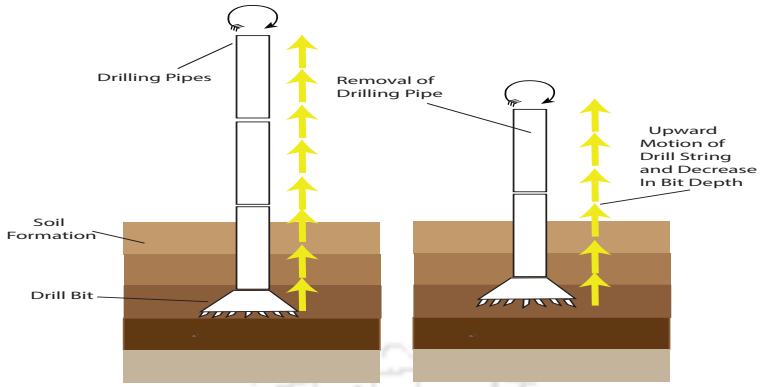


Figure 3.3: *Tripping out with rotation*

4. **Tripping out without rotation (TROWO):** In this activity, the drill string pipes are taken out from the bore well without rotation. Due to upward motion, the Bit depth decreases and the Total depth remains constant. Figure 3.4 shows the TROWO activity.

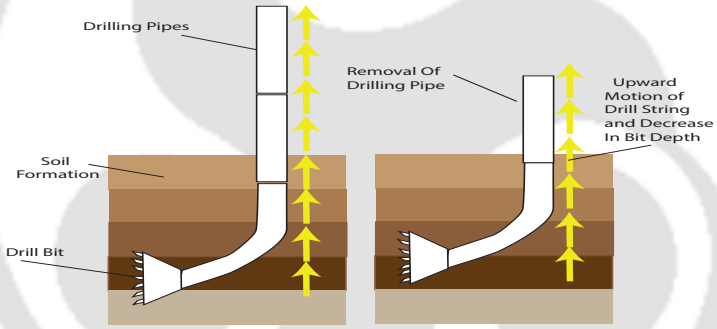


Figure 3.4: *Tripping out without rotation*

5. **Tripping in with rotation (reaming) (TRIWR):** During this activity, the drilling pipes are inserted inside the bore well with rotation. Due to downward movement, the Bit depth increases and the Total depth remains constant. Figure 3.5 shows the TRIWR activity.

6. **Tripping in without rotation (TRIWO):** During this activity the drill string pipes are inserted into the bore well without rotation. The Bit depth increases but

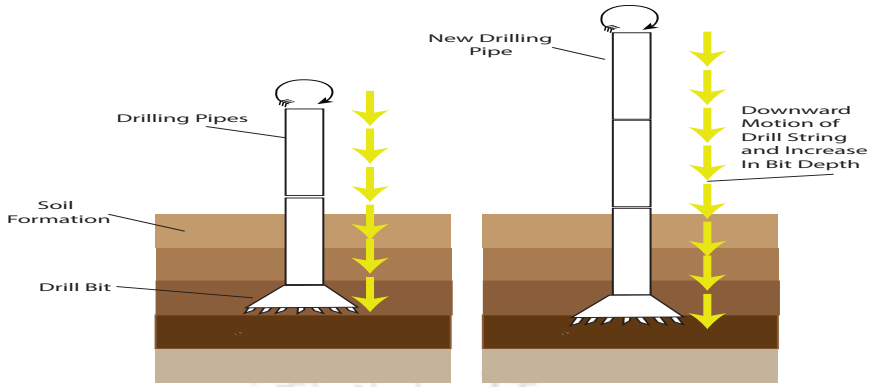


Figure 3.5: *Tripping in with rotation*

the Total depth remains constant. Figure 3.6 shows the TRIWO activity in the drilling rig.

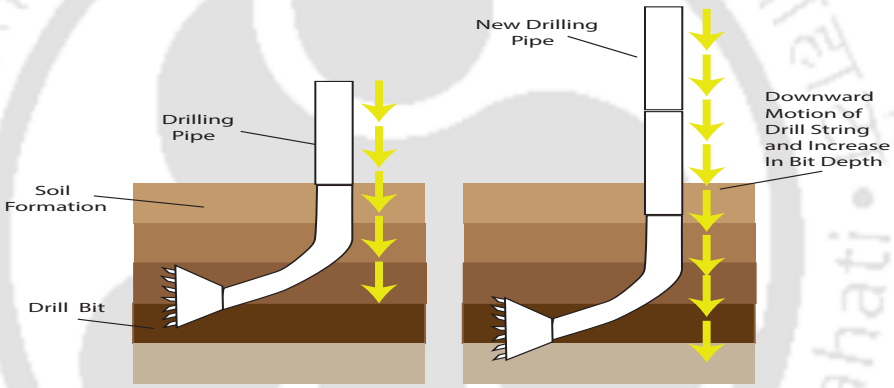


Figure 3.6: *Tripping in without rotation*

7. **Rotation on bottom (RONB) (Circulation with rotation):** During rotation on the bottom, the mud fluid is circulated inside the bore well to take soil cuttings out from the bore well. The Bit depth and Total depth remain constant, and the drill string pipes are rotated continuously. This process is also known as Circulation with rotation. Figure 3.7 shows the RONB activity.

8. **Circulation without rotation (CRWO):** In this activity, the mud is circulated inside the bore well to take the soil cuttings out from the bore well. Bit depth and Total depth remain constant, and the rotation of the drill string is zero. Figure

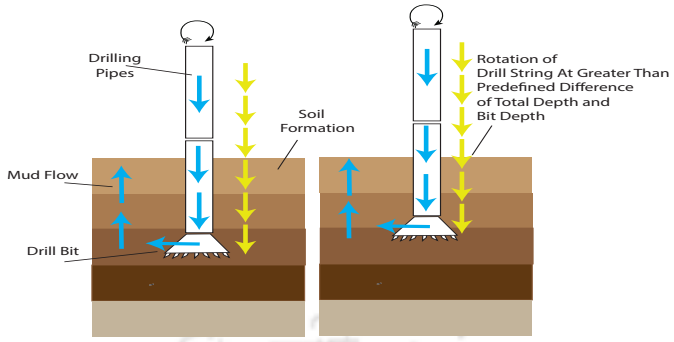


Figure 3.7: *Rotation on bottom*

3.8 shows the CRWO activity.

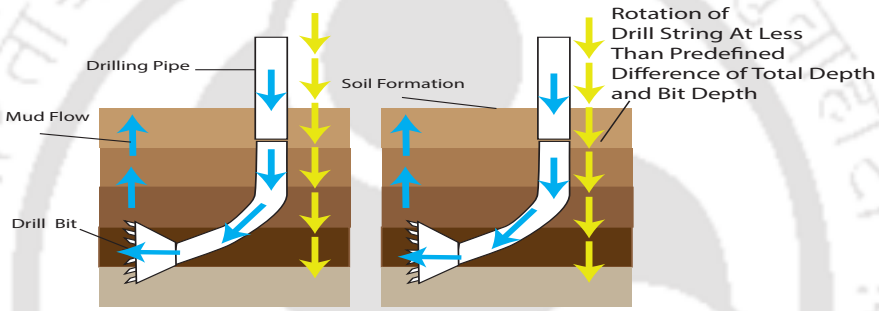


Figure 3.8: *Circulation without rotation*

9. **Rotation off bottom (ROFB):** In rotation off bottom, the mud is circulated inside the bottom hole to take the soil cuttings out from the bottom hole. The Bit depth and the Total depth remain constant, but the drill string pipes are rotated continuously. Figure 3.9 shows the ROFB activity.

10. **Reciprocation (RECI):** During the reciprocation, the drill string pipe moves upward and downward direction along with the continuous mud circulation. The Total depth remains constant, and the Bit depth increases or decreases. Figure 3.10 denotes the RECI activity.

11. **No operation (NOP):** This activity belongs to the non-productive state of the rig like pipe connection, casing, sensor errors, etc.

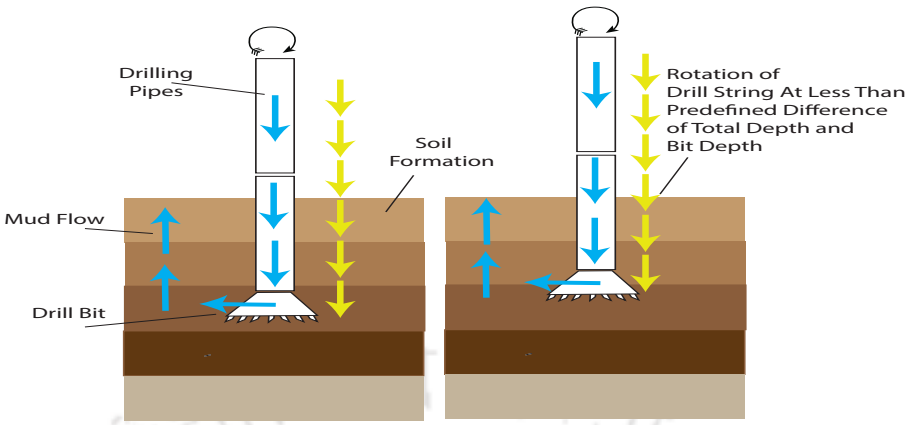


Figure 3.9: *Rotation off bottom*

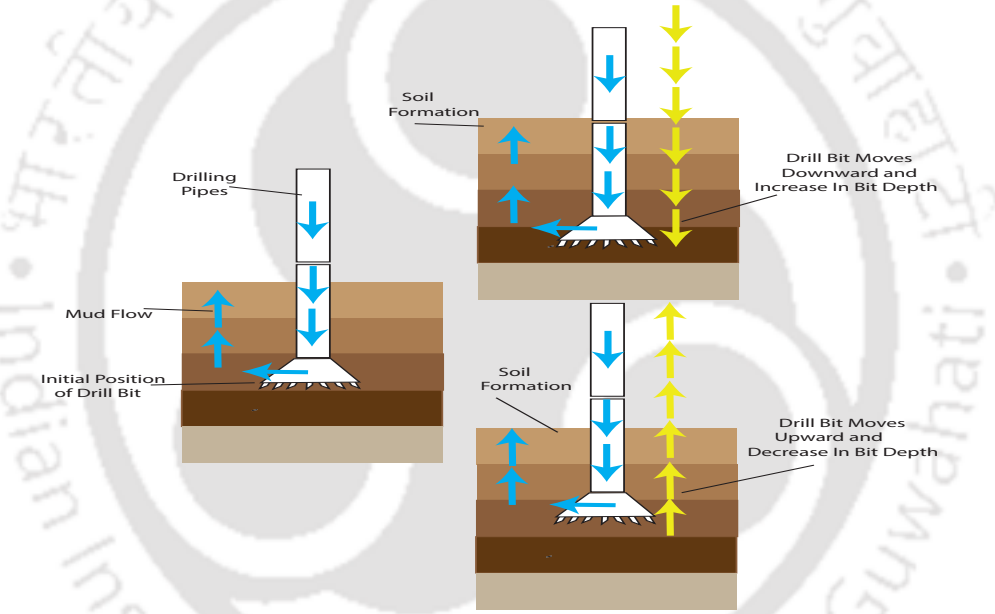


Figure 3.10: *Reciprocation*

Table 3.1 shows variation in the drilling parameters for the above-mentioned drilling activities. Figure 3.11 shows a hierarchy of oil well drilling activities.

3.2.2 Fuzzy Rule-Based (FRB) Classifier

The fuzzy rule-based system represents human reasoning as a collection of IF-THEN rules. Fuzzy rules are basically used to model the non-linear valued function such as

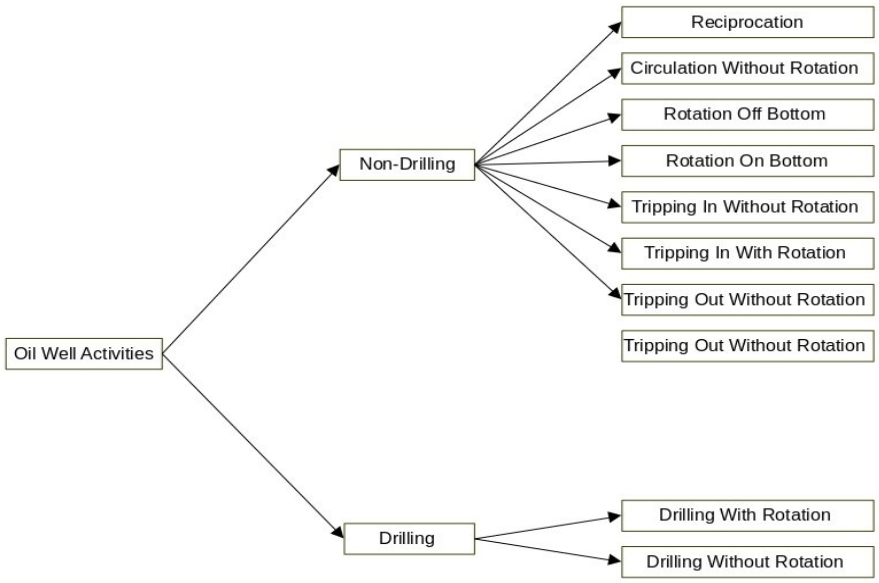


Figure 3.11: *Hierarchy of Oil Well Activities*

Table 3.1: *Variation in drilling parameters during different drilling activities*

Parameters	Drilling Activities									
	DRWR	DRWO	TROWR	TROWO	TRIWR	TRIWO	RONB	ROFB	RECI	CRWO
TD	I	I	C	C	C	C	C	C	C	C
BD	I	I	D	D	I	I	C	C	I/D	C
HL Difference	GTZ	GTZ	GTZ	GTZ	GTZ	GTZ	Z	Z	GTZ	Z
RPM	GTZ	Z	GTZ	Z	GTZ	Z	GTZ	GTZ	Z	Z
TD-BD Difference	S	S	H/S	H/S	H/S	H/S	GTPV	LTPV	H/S	LTPV
SPM	GTZ	GTZ	Z	Z	Z	Z	GTZ	GTZ	GTZ	GTZ

I= Increases, D= Decreases, C= Constant, GTZ= Greater Than Zero, Z=Zero
H= High, S=Small, GTPV= Greater Than Pre-Defined Value
LTPV= Less Than Pre-Defined Value,TD=Total Depth,
BD=Bit Depth, HL=Hookload, SPM=Strokes Per Minute, RPM=Rotation Per Minute

expert knowledge and logics. A Fuzzy rule-based classifier uses fuzzy rules and an inference mechanism to yield either soft or crisp class label for a given input. Here, we are using Takagi-Sugeno-Kang (TSK) [199] rules of the following form:

$$Rule^i : IF\ x_1 is\ A_1^i\ AND.... x_n is\ A_1^i\ THEN\ y_k^i \tag{3.1}$$

$$y_1^i = \sum_{j=1}^{N} a_{j1}^i x_j \ AND .... y_m^i = \sum_{j=0}^{N} a_{jm}^i x_j \tag{3.2}$$

Where $x_j,\ j = 1, 2, .., N$ is the j^{th} input variable, A_1^i is a antecedent fuzzy set of rule i, $y_k^i, i = 1, 2, .., M$, is the output rule i for each class and a_{jk}^i is the j^{th} consequence parameter of the output k of the rule i. In this model every rule votes for all the classes. Among several methods, the output can be determined using the weighted sum aggregation method.

$$L_k(x) = \frac{\sum_i^N y_k^i \tau^i(x)}{\sum_i^N \tau^i(x)} \tag{3.3}$$

where τ^i is the firing strength of rule i. For a crisp label, x is assigned to the class with maximum value of $L_k(x)$.

3.2.3 Random Forest (RF) Classifier

Random forest classifier [132] is an extensively used supervised machine learning algorithm that produces promising results without tuning of hyperparameters as in case of the other supervised machine learning techniques. It is well suited to perform the classification and regression tasks. The random forest consists of collection of numerous decision trees that are created and ensemble to perform the classification. Typically bagging technique is used to train the decision trees of the RF classifier. Unlike the decision tree method, instead of selecting the best feature to split the node, random forest creates the subset of features and choose the best feature to split the node. RF uses a weighted majority voting scheme to classify the unseen data instances. Figure 3.12 shows the classification procedure of the RF classifier.

3.3 Methodology

This section describes the novel two-layer framework that ensembles FRB and RF classifiers at the stacked layers to identify the drilling activities. The FRB classifier efficiently captures uncertainty present in drilling data. The fuzzy rules used in FRB classifier are

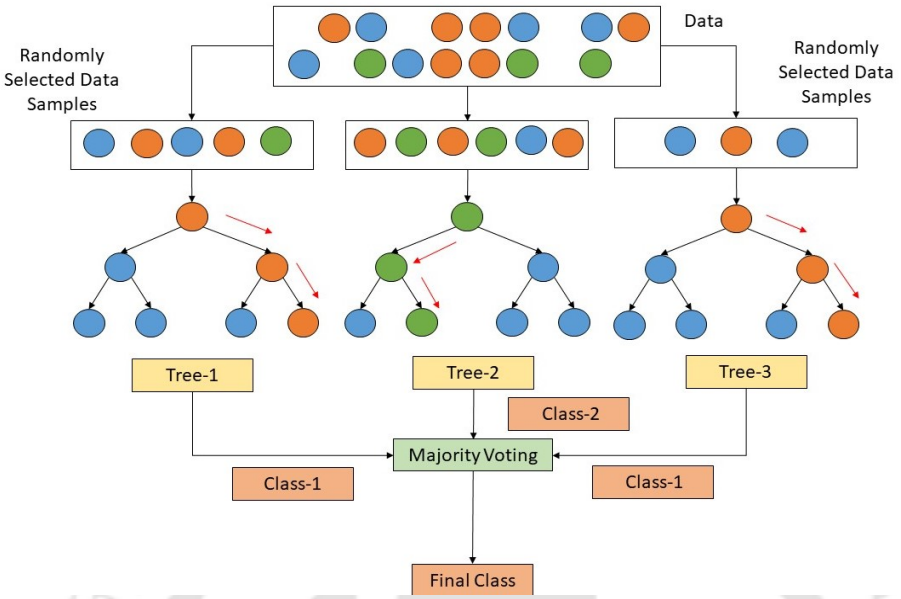


Figure 3.12: *Random forest classifier*

interpretable and easy to understand. The random forest classifier is an ensemble classifier that aggregates decision of multiple decision trees. The expert drilling knowledge can be easily expressed in the form of decision tree. These decision trees are easy to read and interpret, without even requiring statistical knowledge. However, the second layer is employed as the sub-activities under drilling and non-drilling were not classified with required accuracy at the first level. Different models are tested at the second layer and random forest layer obtained the best results while classification of the ten drilling activities in combination with FRB classifier. The reasons mentioned above motivated use to select and stack these classifier to develop the proposed two-level framework that identifies the drilling activities. Figure 3.13 shows the novel two-layer framework for the classification of the various drilling activities. Significant steps to define the working of the proposed framework are as follows:

1. The two-layer framework is divided into two layers, at the first layer FRB classifier identifies broad drilling activities (tripping or drilling) associated with the data instance received from the SCADA system.

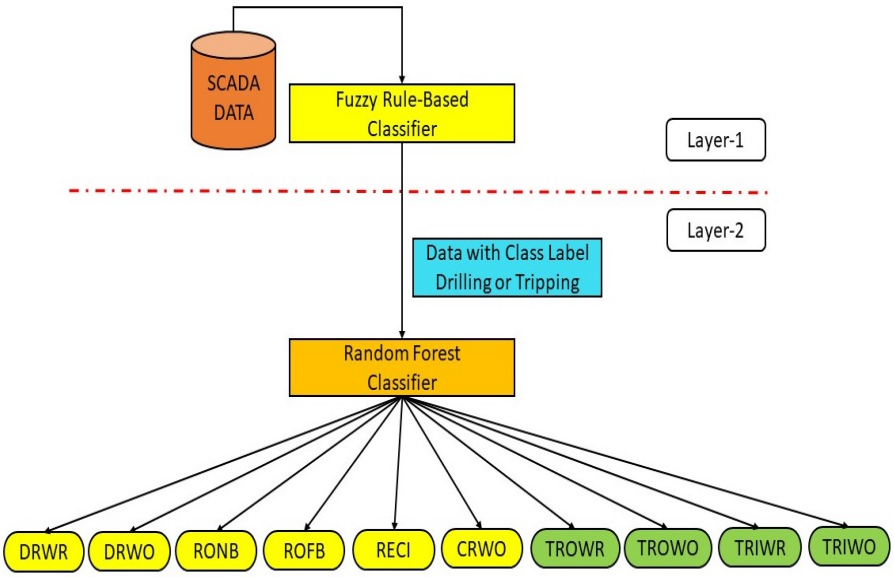


Figure 3.13: *Stacked two-level classifier to classify various drilling activities*

2. The data instance classified at the first layer is used as an input to the Random forest classifier at the second layer. Random forest classifier again sub-classifies the data instance as the specific drilling activities as shown in Table 3.1.

Algorithm (1) provides the details of various steps of the proposed framework. Figure 3.13 shows how various drilling and tripping activities identified in the first layer are sub-classified to other drilling activities in the second layer. All the yellow colored activities belong to the drilling activities, and all the green activities denote the tripping activities.

ALGORITHM 1: Classification of drilling activities using proposed framework

Input: SCADA Data, Training Data Layer-1(X_{Train}^{Layer1}),Training Data Layer-2(X_{Train}^{Layer2}), Test Data Layer-1(X_{Test}^{Layer1}), Test Data Layer-2 (X_{Test}^{Layer2}), Hookload (HL), Strokes Per Minute(SPM), Total Depth (TD), Bit Depth(BD), Rotation Per Minute (RPM), Flowout (FO)

Output: Drilling Activity (Y)

Use drilling expert rules of Table 3.1 to prepare X_{Train}^{Layer1} data from SCADA data to develop FRB classifier at layer-1.;
Select HL, FO, TD and BD from X_{Train}^{Layer1} and create fuzzy sets using values of HL, FO, and difference of TD and BD.;
Use the fuzzy sets obtained in previous step to create the FRB classifier to classify the drilling and tripping activities. ;
Use the above FRB classifier to separate the data belong to drilling or tripping activity in the SCADA data.;
Again use the expert drilling rules in Table 3.1 on the data obtained in previous step to prepare the (X_{Train}^{Layer2}) data to train the RF classifier at layer-2;
Select the HL,TD, BD and FO parameters from SCADA data to prepare test data (X_{Test}^{Layer1}) for the FRB classifier;
Classify (X_{Test}^{Layer1}) at the layer 1 using the FRB classifier ;
Assign class label Drilling' or 'Tripping' to (X_{Test}^{Layer1}) by the FRB ;
if *Class label == Drilling' or 'Tripping'* **then**
    Select HL, SPM, RPM, TD and BD parameter from SCADA data to prepare test data (X_{Test}^{Layer2}) for the RF classifier;
    Extract derived features from the (X_{Test}^{Layer2})data using conditions in Table 3.8.;
    Classify (X_{Test}^{Layer2}) at layer 2 using the RF classifier ;
    Assign drilling activity Y to (X_{Test}^{Layer2}) ;
else
    Discard SCADA data
Return Drilling Activity (Y);

3.4 Experiments and Results

This section first provides description of the data set and then explains the conducted experiments and obtained results. All experiments are conducted in a computer with Windows 10 operating system, 16 GB RAM and cpu with 3.4 GHz frequency. All scripts are written using MATALB 2018 A version.

3.4.1 Data set Description

The Major objective of the proposed method is to classify the ten drilling activities as mentioned in the Table 3.1. To accomplish the task we considered the SCADA data of four oil rig wells recorded at an interval of seven seconds and provided by Oil & Natural Gas Corporation Limited (ONGC), India. The description and location of wells selected for the validation are shown in Table 3.2. Training data contains total 2619654 data instances. Table 3.3 shows the eleven drilling parameters and their units obtained from the SCADA system of the rigs. These drilling parameters are used as features for learning the supervised models for classification of the drilling activities. The test data consist of total 246654 data samples. Distribution of data samples used to train the proposed method is shown in Table 3.4.

Table 3.2: *Specification of four rig wells*

S.No	**Rig Name**	Location	**Starting Date**	**Ending Date**
1	A	Assam (India)	04/02/2017	28/06/2017
2	B	Assam (India)	25/01/2017	05/05/2017
3	C	Assam (India)	01/01/2016	02/02/2017
4	D	Assam (India)	01/12/2016	01/03/2017

3.4.2 Preprocessing of Drilling Data

The drilling data gathered in the SCADA system contains various sensor errors, noise and missing values. The rules of preprocessing are applied to remove the noise, sensor errors and missing values present in the data to get the clean and well organized data to

Table 3.3: *Units of different drilling parameters*

S.No.	Parameter	Unit
1	Hookload (HL)	$tonfus$
2	Stand Pipe Pressure (SPP)	$1kgf/cmA$
3	Strokes Per Minute (SPM)	spm
4	Weight on Bit (WOB)	$tonfus$
5	Rotation Per Minute (RPM)	rpm
6	Flowout (FO)	$rel\ \%$
7	Rate of Penetration (ROP)	$meter/hour$
8	Total Depth (TD)	$meter$
9	Bit Depth (BD)	$meter$
10	Inlet Density (DI)	g/cm^3
11	Outlet Density (DO)	g/cm^3

Table 3.4: *Data distribution for various drilling activities in training and test data set*

Drilling Activities	Number of Data Samples
DRWR	269660
DRWO	256612
TROWR	262364
TROWO	251638
TRIWR	253640
TRIWO	255645
RONB	264660
ROFB	274649
RECI	250630
CRWO	279886
Total	2619654

train the ML models. Table 3.5 shows various expert rules given by the expert drillers to perform preprocessing of the drilling data.

Table 3.5: *Expert drilling rules for preprocessing*

Preprocessing Drilling Rule	Descriptions
Rule-1	Remove all data rows with negative values
Rule-2	Remove all data rows having values beyond the minimum and maximum measurement range of sensors
Rule-3	Remove all data rows having missing values

3.4.3 Feature extraction

In the proposed method two different feature extraction techniques are used to train the FRB and RF classifiers.

Table 3.6: *Fuzzy sets created from the drilling parameters for FRB classifier at layer 1*

S.No	Parameters	Fuzzy Sets
1	HL	High HL, Medium HL, Low HL
2	FO	High FO, Medium FO, Low FO, Very Low FO
3	TD and BD	High , Medium and Very Low difference of TD and BD

3.4.3.1 Feature extraction for the FRB classifier

Among the drilling parameters as shown in the Table 3.3 Hookload (HL), Flowout (FO), Total Depth (TD) and Bit Depth (BD) are considered as the features to train the FRB classifier. Fuzzy sets defined for the respective drilling parameters are given in Table 3.6. Fuzzy sets are formed for the HL, FO and difference of TD and BD parameters as shown in the Figures 3.14-3.16. Further, the created fuzzy sets with a Gaussian membership function are used to create the fuzzy rules of the FRB classifier.

3.4.3.2 Feature extraction for the RF classifier

The RF classifier is trained using the drilling parameters Hookload (HL), Bit Depth (BD), Total depth (TD), Rotation Per Minute (RPM) and Strokes Per Minute (SPM). Different alphabetical symbols are assigned to the drilling parameters based on the various conditions satisfied by the values of the drilling parameters as shown in Table 3.8. For example, it is shown in the Table 3.8 how the two different symbols c and d are used to denote the zero and non zero values of the SPM parameter.

3.4.4 Selection of various hyperparameters

The clustering and expert knowledge both have been employed to fix the number of rules and the fuzzy set. We first applied subtractive clustering [59] and adjusted the

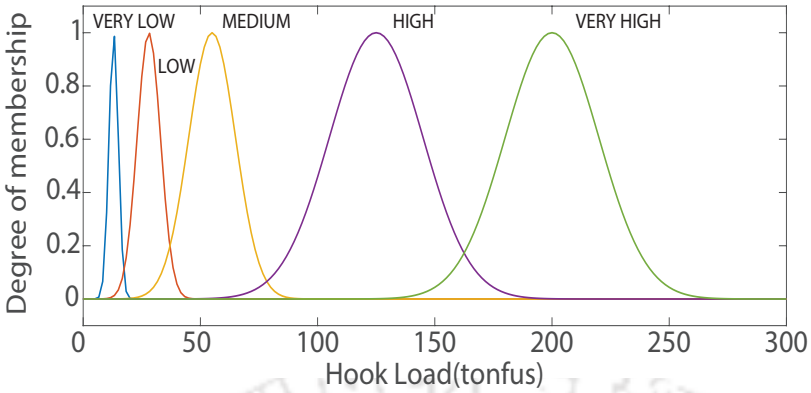


Figure 3.14: *Fuzzy set for the Hookload Parameter*

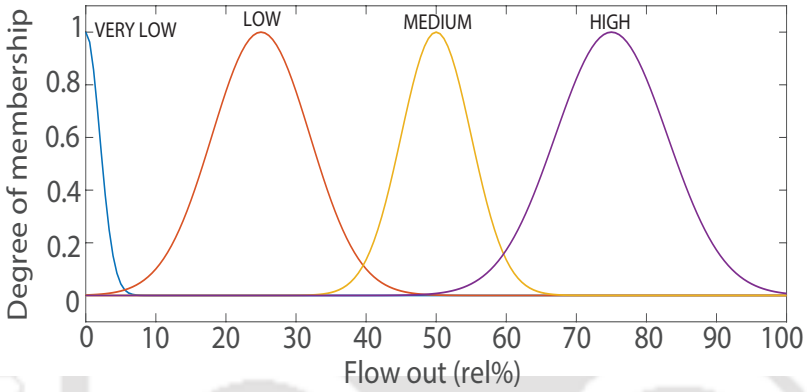


Figure 3.15: *Fuzzy set for the Flowout Parameter*

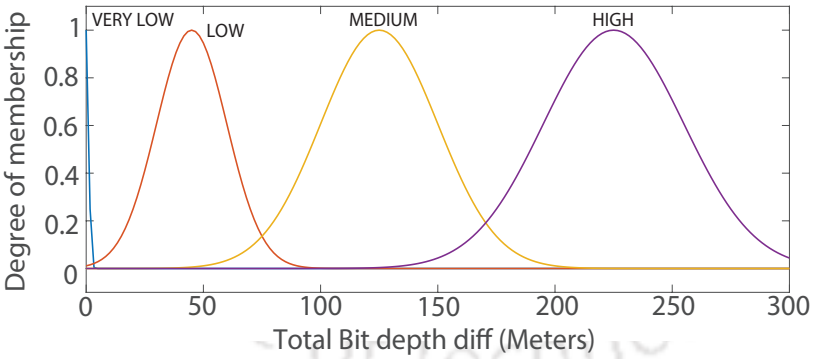


Figure 3.16: *Fuzzy set for the difference of Total Depth and Bit Depth parameters*

parameters using expert knowledge. The consequent parameters are determined using recursive least-squares approach [33]. Training of RF classifier needs fine-tuning of different hyperparameters. The number of trees in a random forest classifier is selected to

Table 3.7: *Fuzzy rules formed to identify drilling and tripping activities at the layer 1*

S.NO	Hookload	Flowout	Difference of Total Depth and Bit Depth	Activity
1	High	Low	Very Low	Drilling
2	High	Medium	Very Low	Drilling
3	High	High	Very Low	Drilling
4	Medium	Low	Very Low	Drilling
5	Medium	Medium	Very Low	Drilling
6	Medium	Low	Very Low	Drilling
7	High	Very Low	High	Tripping
8	High	Very Low	Very High	Tripping
9	Medium	Very Low	High	Tripping
10	Medium	Very Low	Very High	Tripping
11	Low	Very Low	High	Tripping
12	Low	Very Low	Very High	Tripping

Table 3.8: *Derived features from the drilling parameters for RF classifier at layer 2*

S.No	Parameters	Derived Feature	Condition	Symbol
1	**HL**	Zero HL Difference	$(HL_t - HL_{t-1} = 0)$	a
		Non Zero HL Difference	$(HL_t - HL_{t-1} > 0)$	b
2	**SPM**	Zero SPM	$(SPM_t = 0)$	c
		Non Zero SPM	$(SPM_t > 0)$	d
3	**RPM**	Zero RPM	$(RPM_t = 0)$	e
		Non Zero RPM	$(RPM_t > 0)$	f
4	**TD**	Increasing TD Difference	$(TD_t - TD_{t-1} > 0)$	i
		Decreasing TD Difference	$(TD_t - TD_{t-1} < 0)$	j
		Constant TD Difference	$(TD_t - TD_{t-1} = 0)$	k
		Difference with BD	abs $(TD_t - BD_t) \geq thr$	l
5	**BD**	Increasing BD Difference	$(BD_t - BD_{t-1} > 0)$	m
		Decreasing BD Difference	$(BD_t - BD_{t-1} < 0)$	n
		Constant BD Difference	$(BD_t - BD_{t-1})$	o
		Difference with TD	abs $(TD_t - BD_t) < thr$	l

reduce the computational cost. Large number of trees produce more generalized result. The number of trees is directly proportional to the computational cost. In our study, the RF is trained with 70 number of trees, and the trained RF is capable of providing a class label to the data samples before the arrival of the next data sample. The duration in between the arrival of the two data samples is 7 seconds. The value of the maximum number of samples in the leaf node is selected as 300 beyond that model starts over-fitting. The value of the maximum value of terminal nodes is set as 15 to prevent the overfitting issue.The value of the minimum sample split is used as 1. All the values of the hyperparameters are selected to prevent the RF classifier from overfitting or underfitting issues to classify the given drilling data

3.4.5 Identification of the drilling activities

The FRB and RF classifiers are trained over the training data using the extracted features as discussed in the earlier section. The FRB classifier contains a total of twelve fuzzy rules as shown in Table 3.7. The RF classifier is trained using seventy decision trees. The test data contains the drilling data of four different wells, but here we have shown various drilling activities identified for the rig A. We developed three more models Decision Tree (DT), Random Forest(RF) and Multiclass Support Vector machine (SVM) to compare the performance of the proposed model. We selected these three models as they are well applied to detect various drilling activities [73]. The models are evaluated for five-fold cross validation setting using nested cross validation technique [35]. We used the same technique as used in the [73] to extract features from the multivariate time series of the drilling data. We selected the window size of 10 data samples to extract the features from the drilling data. The models are trained using the data belonging to each activity. Figures 3.17-3.26 show the different drilling activities identified by the proposed method for the continuous five days of drilling in the month of March for rig A. In the Figures 3.17-3.26 per 500 data samples denote one-hour drilling. Total ten colors are used to represent the different drilling activities. The Figures 3.17-3.26 show the behavior of the Hookload (HL), Total depth (TD) and Bit depth (BD) parameters when the rig A enters to different states of the drilling activity. In the Figure 3.17 the

data range 5600 to 7499 shows the ROFB activity and data samples range from 7500 to 8000 show the TROWO activity. For the same range the Figure 3.18 shows the TD is constant and BD is decreasing. Both the figures mentioned above validate the expert knowledge of variation in drilling parameters of the TROWO activity as given in Table 3.1. The TRIWO activity can be seen for the data range 1000 to 1600 in Figures 3.23-3.24. All the data samples with the dark green and yellow colors in Figures 3.17-3.26 denote the DRWR and CRWO activities.

Figure(3.30), Figure(3.31), Figure(3.32) and Figure(3.33) show multi-label classification matrix obtained for the classification of the test data set by DT, RF, SVM and the proposed classifier. Figure 3.27 and Figure 3.28 show comparison of precision and recall of the four models. Figure 3.29 shows the comparison of accuracy obtained by all the four models. The SVM shows a lowest accuracy of 76.52%±0.23 to classify the drilling activities. The DT and RF obtain the accuracy of 79.44%±0.29 and 83.67%±0.25 respectively. The above three models are failed to capture the uncertainty of the drilling data, due to which the accuracy is less to the proposed model. The FRB classifier used in the first layer efficiently models the uncertainty of the drilling data and unlike the three models mentioned above, the proposed model uses different combination of drilling parameters to train the classifier present at the different layers. Therefore the proposed model identifies the ten drilling activities with high accuracy. Further, the proposed method can be used to generate daily drilling report (DDR) in terms of percentages of time spent to perform the drilling activities. The existing DDR is prepared and filled by drillers. In the current scenario, the DDR reports record only the activity that has been performed for a longer duration, this loss of information can be recovered by the proposed method. Figures 3.34-3.37 show a comparison of the percentage of time spent to perform the ten drilling activities for the drilling data of the four rig wells. The drilling data of rig C shows that 43.25% of time is spent to perform the DRWR activity, 6.62% of time is spent to complete DRWO activity likewise 1.30%, 5.75%, 0.93%, 3.28%, 3.34%, 4.7%, 0.46% and 30.30% percentages of time are spent to perform the TROWO, TROWR, TRIWR, TRIWO, RONB, ROFB, CRWO and RECI activities. Table 3.9

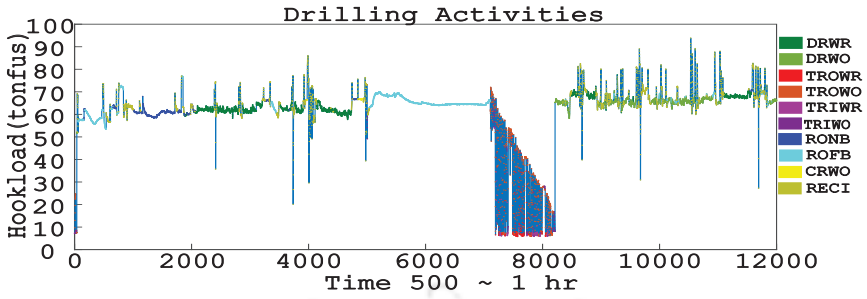


Figure 3.17: *Various drilling activities at Day-1*

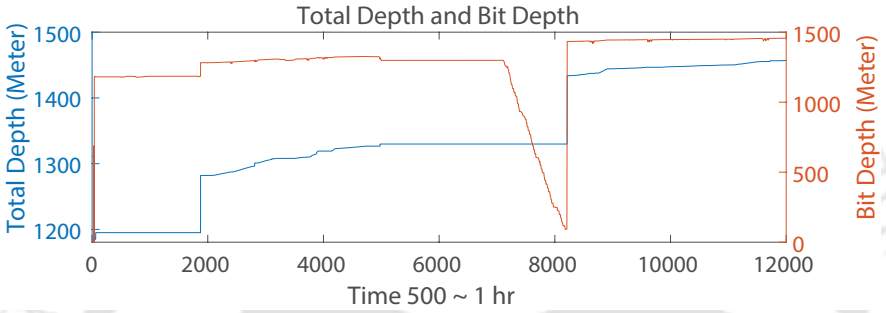


Figure 3.18: *Variation in Total depth and Bit depth at Day-1*

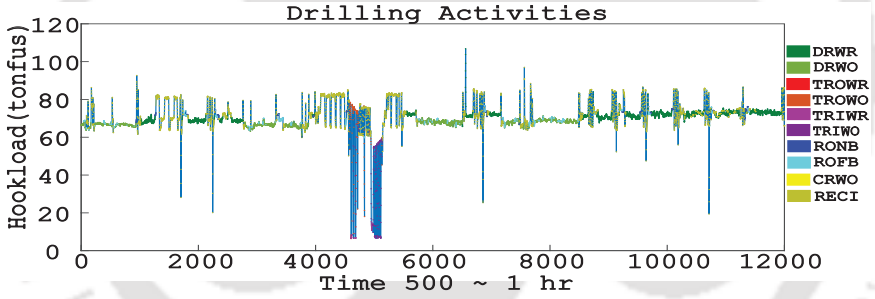


Figure 3.19: *Various drilling activities at Day-2*

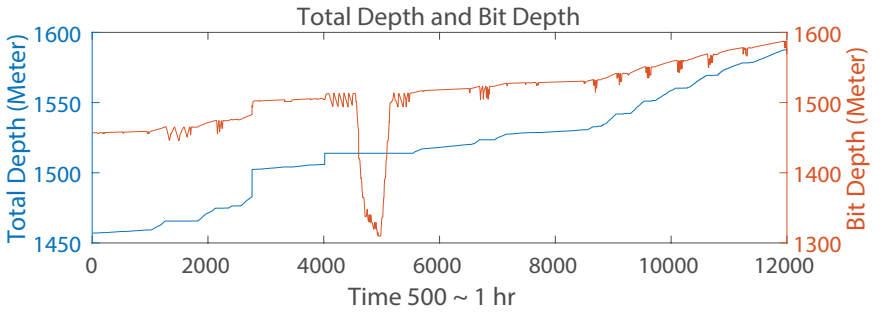


Figure 3.20: *Variation in Total depth and Bit depth at Day-2*

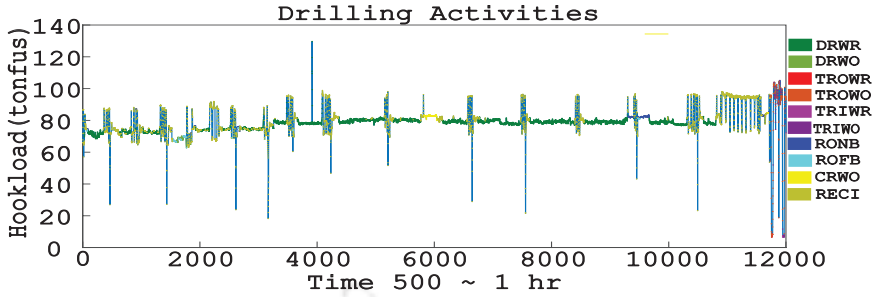


Figure 3.21: *Various drilling activities at Day-3*

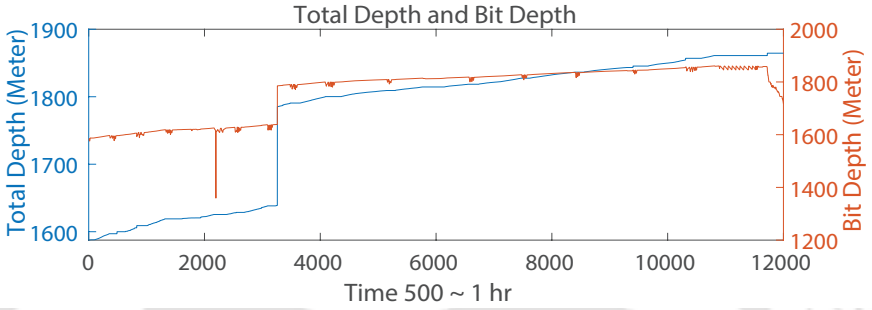


Figure 3.22: *Variation in Total depth and Bit depth at Day-3*

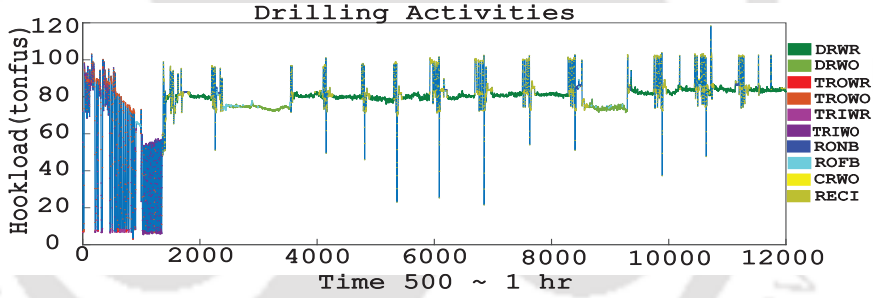


Figure 3.23: *Various drilling activities at Day-4*

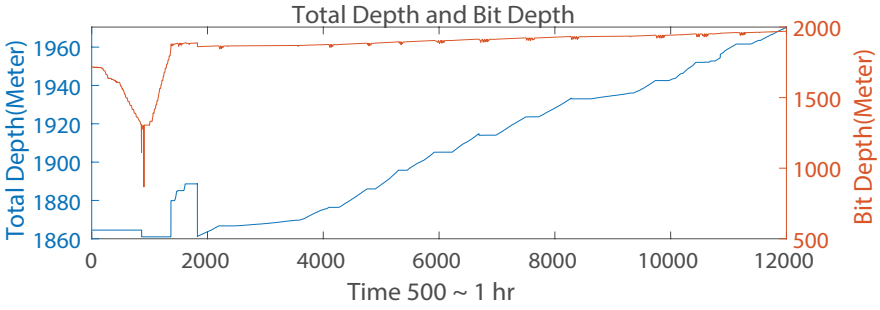


Figure 3.24: *Variation in Total depth and Bit depth at Day-4*

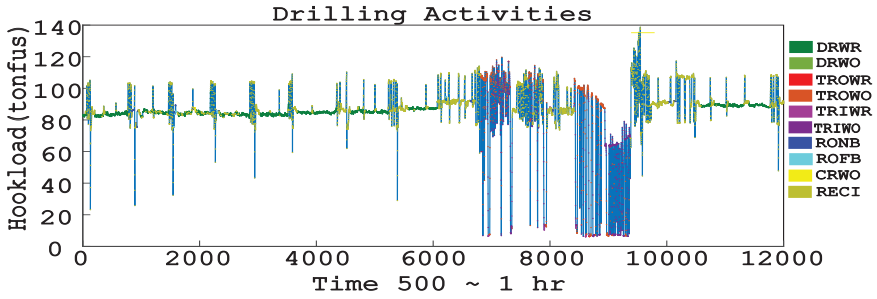


Figure 3.25: *Various drilling activities at Day-5*

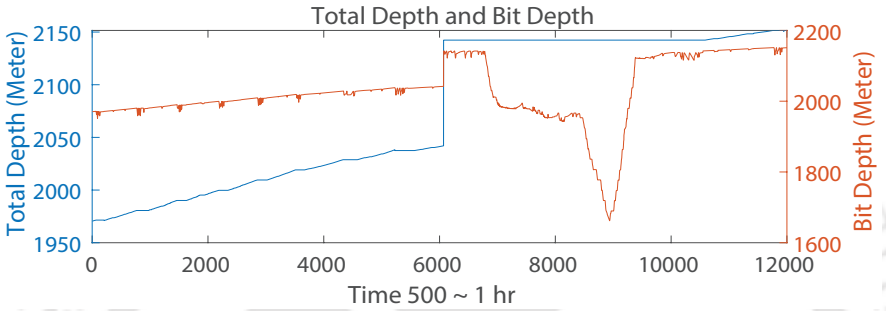


Figure 3.26: *Variation in Total depth and Bit depth at Day-5*

denotes the percentage of time spent to perform the ten drilling activities in the four wells.

Table 3.9: *Time spent to perform various drilling activities in the four different wells*

S.NO	RIG	Drilling Activities									
		DRWR	DRWO	TROWR	TROWO	TRIWR	TRIWO	RONB	ROFB	CRWO	RECI
1	A	38.73%	6.49%	2.20%	7.21%	0.60%	11.29%	6.72%	3.67%	0.09%	22.93%
2	B	15.24%	0.03%	4.83%	15.07%	2.95%	22.30%	9.79%	4.04%	1.02%	24.69%
3	C	43.25%	6.62%	1.30%	5.75%	0.93%	3.28%	3.34%	4.7%	0.46%	30.3%
4	D	2.80%	0%	4.95%	19.05%	1.97%	22.56 %	4.29%	19.37%	3.09%	21.88%

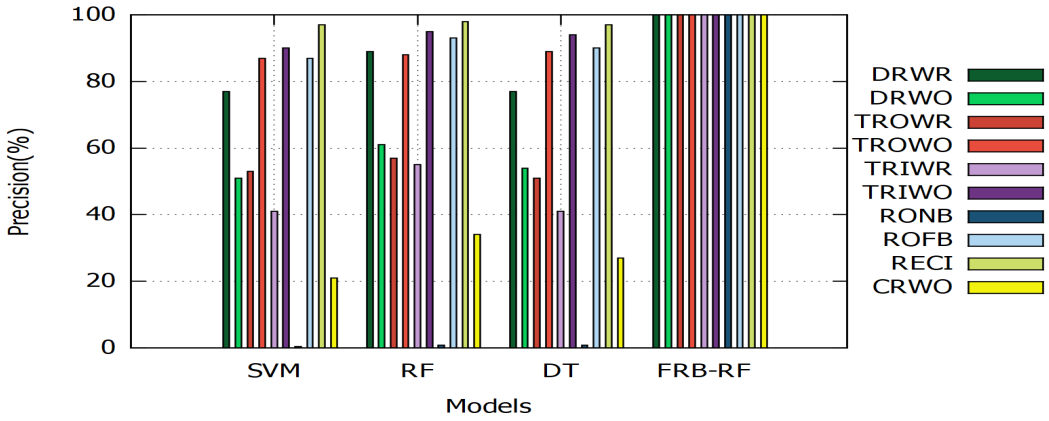


Figure 3.27: *Precision graph of four models*

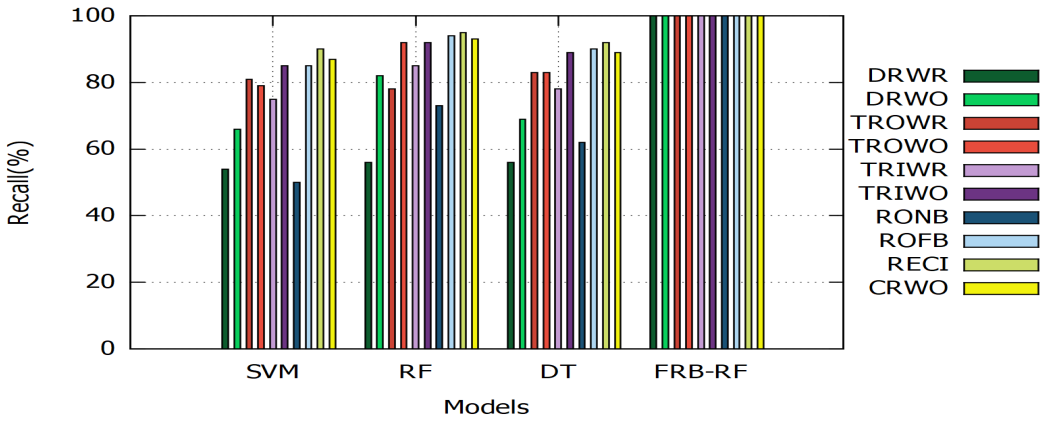


Figure 3.28: *Recall graph of four models*

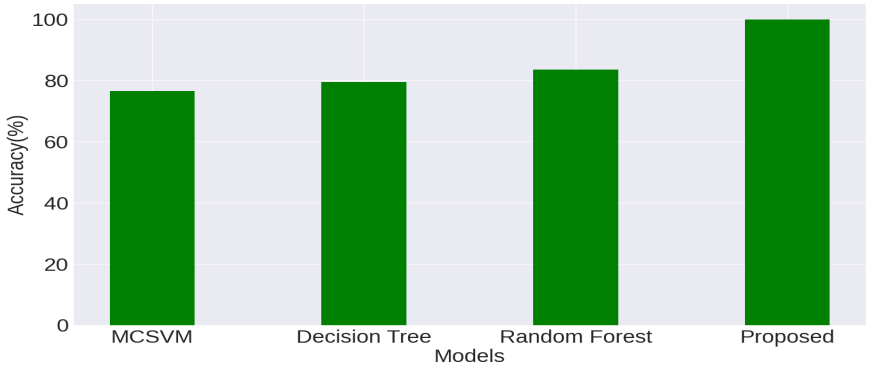


Figure 3.29: *Accuracy of four models*

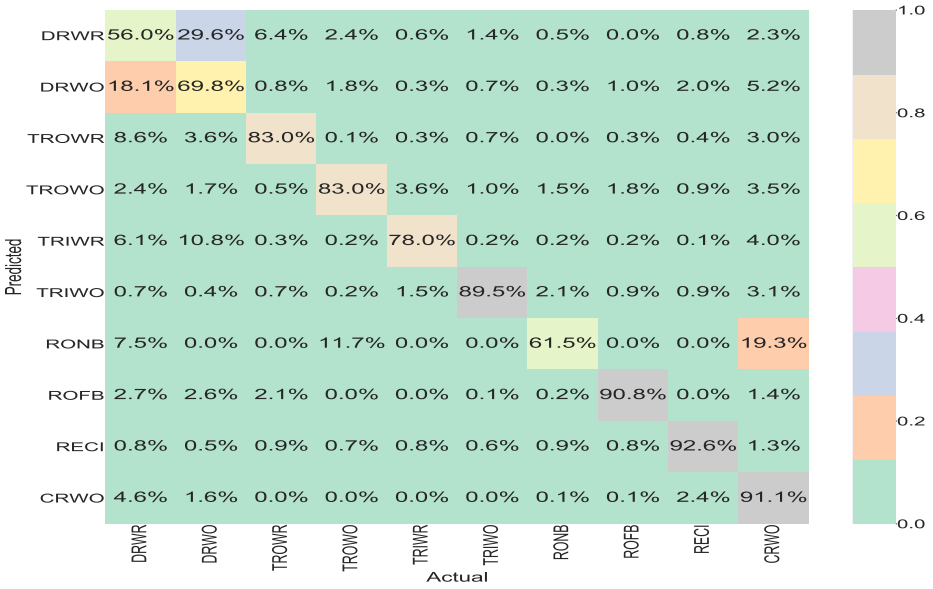


Figure 3.30: *Confusion Matrix For Decision Tree*

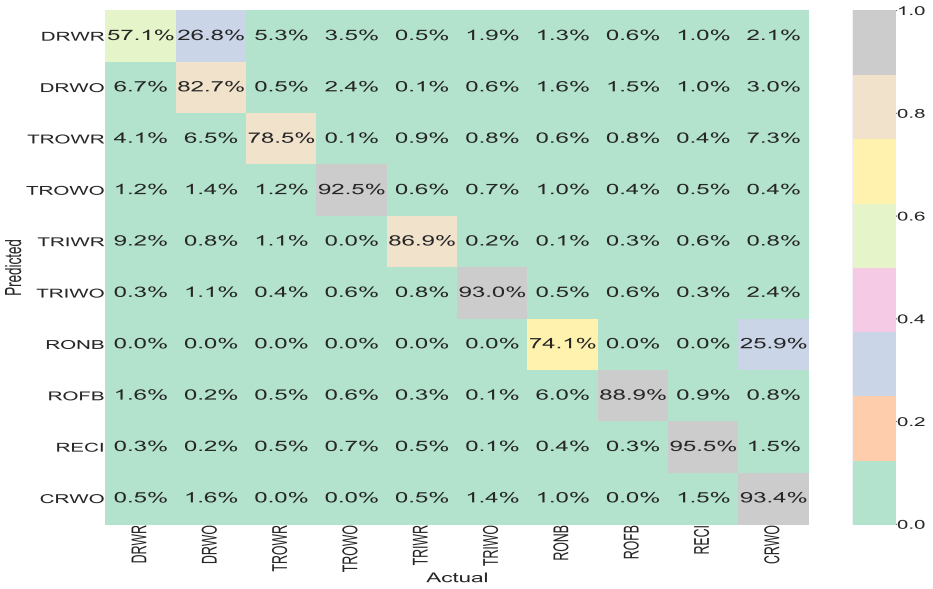


Figure 3.31: *Confusion Matrix For Random Forest*

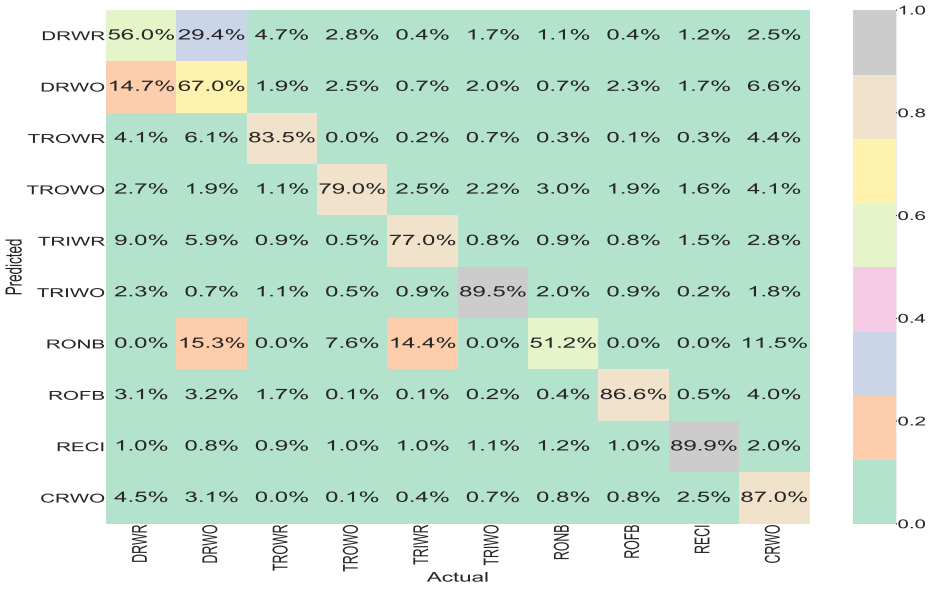


Figure 3.32: *Confusion Matrix For SVM*

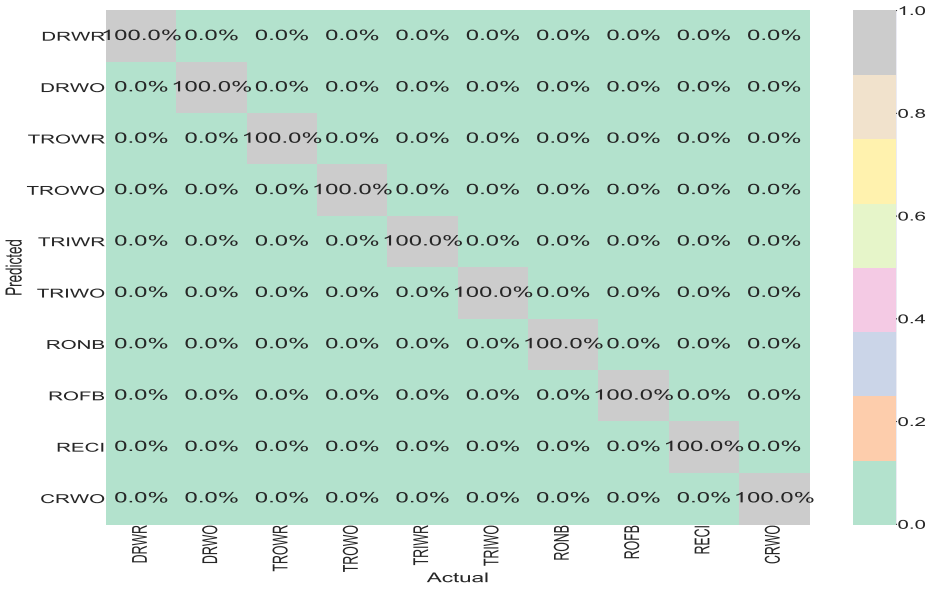


Figure 3.33: *Confusion Matrix For Proposed Method*

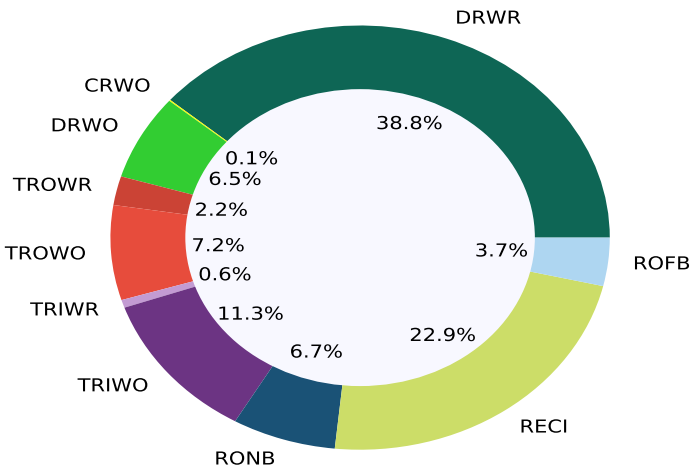


Figure 3.34: *Time spent to perform the various drilling activities in rig A test data*

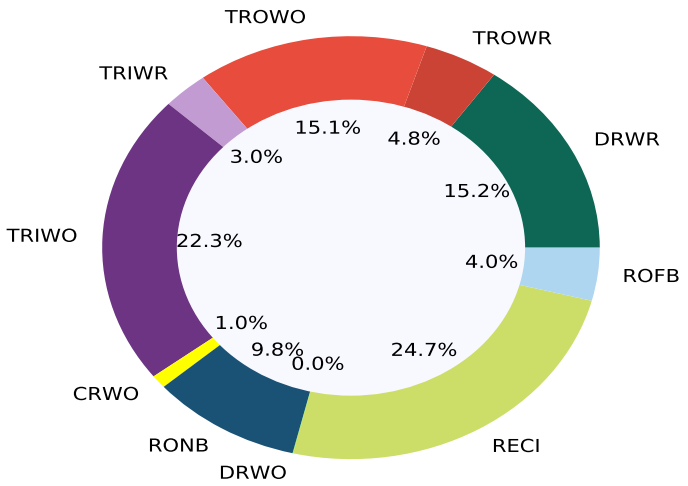


Figure 3.35: *Time spent to perform the various drilling activities in rig B test data*

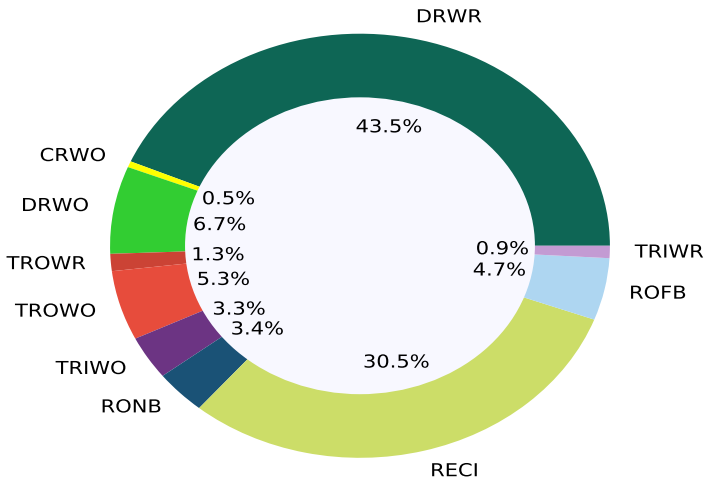


Figure 3.36: *Time spent to perform the various drilling activities in rig C test Data*

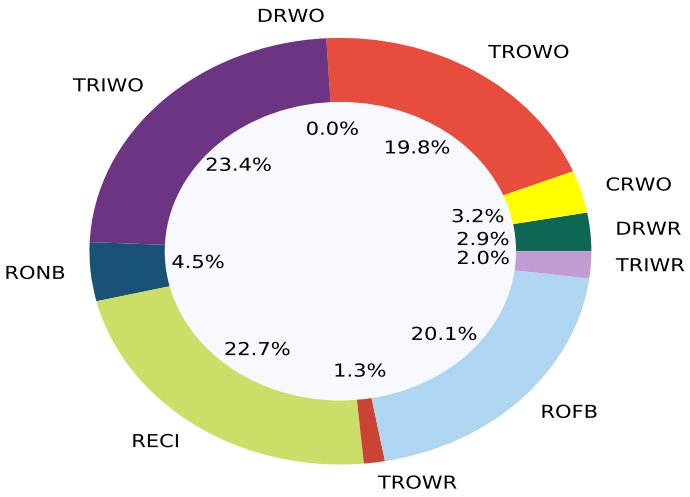


Figure 3.37: *Time spent to perform the various drilling activities in rig D test data*

3.5 Summary

This chapter started with the basic introduction of the various oil well drilling activities. It presented a novel two-level classifier to identify the different drilling activities for the oil well drilling process. The proposed method ensembles the FRB and RF classifiers as stacked layers. The proposed method identified the different oil well drilling activities in the real drilling data with very high accuracy. Further, it generates a detailed report on the amount of time spent to perform each drilling activity in one complete cycle of the drilling process.

Chapter 4

Dynamic Naive Bayesian Classifier and Fuzzy AdaBoost Technique

The previous chapter presented a stacked two-level classifier to identify various oil well drilling activities during the drilling process. The drilling data classified by this classifier is further used to develop the models that can identify the stuck pipe complications during the drilling process. In this chapter a novel anomaly detection classifier is presented that combines the Dynamic Naive Bayesian classifier (DNBC) and Fuzzy AdaBoost technique to detect the stuck pipe anomaly during oil well drilling process.

4.1 Preliminaries

This section presents a brief description of the hidden markov model (HMM) and DNBC.

4.1.1 Hidden Markov Model

In HMM [164], [165] the state of a process can be represented by a single discrete random variable whose values can be the states. The probability of switching between the states S_i to S_j is given by a transition probability. The probability of emission of an observation given a state S_i is the emission probability. These probabilities are maintained as a transition table and emission table respectively. Figure 4.1 shows a HMM with 3 states and 3, observations $O = \{o_1, o_2, o_3, .., o_M\}$. The probabilities associated with HMM are given as

1. Initial probabilities of the states are given as $\pi = \{P(S_1), P(S_2), .., P(S_N)\}$

2. Transition probability between the state S_i to S_j is given as a_{ij}

3. Emission probability b_{ij} is the probability of emission of observation symbol O_j from the state S_i, $P(O_j|S_i)$. $i = 1 \leq i \leq N, j; j = 1 \leq j \leq M$; N =number of hidden states, M= number of observation symbols.

The primary aim of the HMM model is to compute the probability of being in state S_i for the given observation sequence O_j i.e. $P(S_i|O_j)$. Estimation of $P(S_i|O_j)$ requires computation of the of a_{ij} and b_{ij}. Parameters ($\theta = [\pi, a, b]$) of HMM can be computed using Eq.(4.1)-(4.7). At first, Forward and Backward process [165] are applied to compute the likelihood ($P(S_i|O_j, \theta)$) of the given input for the given parameters (θ) of the HMM, then Viterbi algorithm [164] is applied to calculate the sequence of hidden states. Proper training of HMM models correctly reveals the hidden states for the given observation sequences.

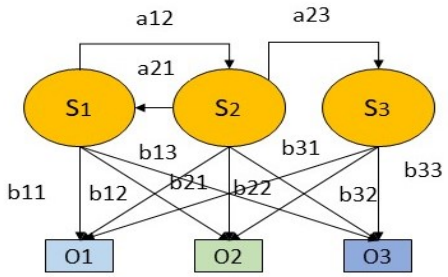


Figure 4.1: *Hidden Markov Model*

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \tag{4.1}$$

$$\alpha_t(i) = (\sum_{j=1}^{N} \alpha_{t-1}(i) a_{ij}) b_i(O_t)) \tag{4.2}$$

Here α and β are the forward and backward probabilities that are further used to calculate the a_{ij} and bij. t is a time stamp and $j = 1 \leq t \leq T$

$$P(O) = \sum_{i=1}^{N} \alpha_T(i) \tag{4.3}$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O)} \tag{4.4}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O)} \tag{4.5}$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)} \tag{4.6}$$

$$b_{im} = \frac{\sum_{t=1,O_t=O_m}^{T}\gamma_t(i)}{\sum_{t=1}^{T-1}\gamma_t(i)} \tag{4.7}$$

Where $\gamma_t(i)$ is probability of being in state i at time t for the given observation O and $\xi_t(i,j)$ is probability of being in state i and state j at time t and $t+1$ for the given observation O.

4.1.2 Dynamic Naive Bayesian Classifier (DNBC)

Traditional HMM computes the parameters of HMM(θ) for the univariate observation sequence. In [29], author proposed the dynamic naive bayesian classifier (DNBC) as shown in the Figure 4.2 to model the multiple observation sequences for gesture recognition. DNBC helps to incorporate the temporal characteristics of the MTS data. It is assumed that all multivariate observation sequences are conditionally independent for the given class label and computation of parameters ($\theta = (a_{ij}, b_{im}, \pi)$) of the DNBC are performed using Eq. (4.3)-(4.4) and Eq. (4.6)-(4.10).

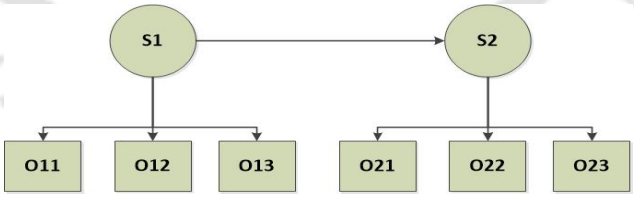


Figure 4.2: *Dynamic Naive Bayesian Network*

$$\alpha_t(i) = (\sum_{j=1}^{N}\alpha_{t-1}(i)a_{ij})\prod_{l=1}^{L}b_i(O_t^l) \tag{4.8}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}(\prod_{l=1}^{L} b_j(O_{t+1}^l)\beta_{t+1}(j)) \tag{4.9}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}(\prod_{l=1}^{L} b_j(O_{t+1}^l)\beta_{t+1}(j))}{P(O)} \tag{4.10}$$

Where, $L=$ is the dimension of multivariate observation sequence.

4.1.3 AdaBoost Technique

AdaBoost [78] is the ensemble technique used to create strong classifier by combining different weak classifiers. In each iteration of the AdaBoost technique, weights are assigned to all the data samples. AdaBoost update these weights to make training algorithm to concentrate more on the data samples that are difficult to classify. After each iteration, a new set of data is created using re-sampling that replicates the number of wrongly classified data samples and further used to train the next weak classifier .

4.1.4 Fuzzy AdaBoost Technique

Misclassification error and updated weights of data samples play an important role during the creation of new dataset using a re-sampling technique for each iteration of the AdaBoost algorithm. Choice of weights is a primary factor that has been extensively researched to design different versions of the AdaBoost method. In [43] author introduced the Fuzzy AdaBoost technique and tried to improve the performance of AdaBoost method by defining the weights as an interval value fuzzy sets (IVFS). In [223] Zadeh first proposed an interval valued fuzzy set where IVFS on x is defined by Eq.(4.11)

$$\overline{F} = [x, \underline{\mu}(x), \overline{\mu}(x)], \underline{\mu}(x) \leq \mu(x) \leq \overline{\mu}(x), \mu\epsilon[0,1] \tag{4.11}$$

Footprint of uncertainty is defined by the upper, $\overline{\mu}(x)$ and lower membership $\underline{\mu}(x)$ functions. FOU [43] based AdaBoost algorithm takes advantage of upper and lower membership of weights to calculate the updated weights of data samples for the next round of boosting algorithm. Calculated updated weights help to create the better distribution of data samples that reduce the misclassification error of the weak classifiers during each iteration.

4.2 Proposed method

This section explains a novel anomaly detection framework by combining fuzzy AdaBoost and ensemble of DNBC classifiers. Table 4.1 shows different notations used in the proposed algorithm.

Table 4.1: *Different notations used in the proposed algorithm*

S.No	Notation	Description
1	w	Weight of each data sample
2	σ	Degree of dispersion of data along dimension L
3	σ^{max}	Maximum value of degree of dispersion
4	σ^{min}	Minimum value of degree of dispersion
5	$R2$	Upper range of FOU
6	$R1$	Lower range of FOU
7	FOU	Footprint of uncertainty
8	$e(k)$	Misclassification Error of k^{th} DNBC
9	$\overline{\mu}_k$	Upper membership FOU of error
10	$\underline{\mu}_k$	Lower membership FOU of error
11	$\overline{\mu}_k^c$	Upper membership of confidence
12	$\underline{\mu}_k^c$	Lower membership of confidence
13	C_k	Confidence of k^{th} DNBC
14	$\overline{\mu}_{k+1}^{w_i}$	Upper membership of $(k+1)^{th}$ DNBC
15	$\underline{\mu}_{k+1}^{w_i}$	Lower membership of $(k+1)^{th}$ DNBC
16	$\overline{Z}_k$	Normalization constant for upper membership
17	$\underline{Z}_k$	Normalization constant for lower membership
18	λ	Defuzzification parameter
19	$y(x_i)$	Actual class label of data x_i
20	$I(x_i)$	Hypothesis for x_i from k^{th} DNBC
21	n	Total number of data samples
22	k	Total number of weak classifiers

4.2.1 Modified Fuzzy AdaBoost algorithm

Our major aim is to create the anomaly detection classifier in the situation when we are unable to collect all possible data samples belonging to an anomalous class. Industrial

monitoring is one of the applications where the aforementioned situation occurs. In [43] author used manually specified footprint of uncertainty (FOU) of error to compute the misclassification errors and updated weights. We modified the existing algorithm for the case of anomaly detection by automatically initializing the FOU of error using the statistical assets of data samples belonging to the normal class. Figure 4.3 and Figure 4.4 are plotted using the two features present in real oil well drilling data set. From Figure 4.3 and Figure 4.4 it is clear that all the data samples from the normal class are less scattered as compared to the data from the anomalous class. Based on the aforementioned concept we design a new generalized interval based function using Eq.(4.13)-(4.18) and new value of FOU is estimated using Eq.(4.18) such that FOU always stay in the sub intervals $[0, R1)$,$[R1, R2]$ and $(R2, 1]$. The following steps are describe to identify the FOU using Eq. (4.13)-(4.18).

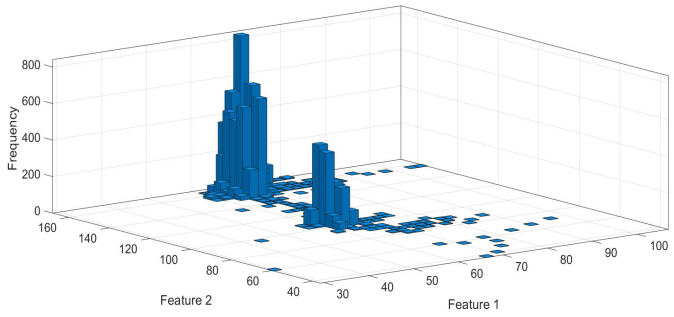


Figure 4.3: *Distribution of normal class data samples*

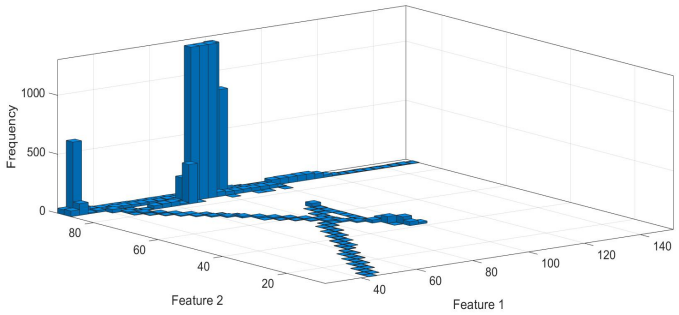


Figure 4.4: *Distribution of negative class data samples*

1. **Step-1:** Initialize the weights of all the data samples $\mathbf{x}_i$ where, n= number of data samples

$$w(\mathbf{x}_i) = \frac{1}{n} \tag{4.12}$$

2. **Step-2:** Compute the mean of data points belonging to the positive class using Eq.(4.13).

$$\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \tag{4.13}$$

3. **Step-3:** Estimate the degree of dispersion(σ) of each positive training data using Eq.(4.14), where L= dimension of the data

$$\sigma(i) = \frac{1}{L}\sum_{l=1}^{L}(\mathbf{x}_{il} - \overline{\mathbf{x}_l})^2 \tag{4.14}$$

$$\sigma_{min} = min(\sigma) \tag{4.15}$$

$$\sigma_{max} = max(\sigma) \tag{4.16}$$

$$R1 = \frac{0.5 * \sigma_{min}}{\sigma_{min} + \sigma_{max}},\ R2 = \frac{0.5 * \sigma_{max}}{\sigma_{min} + \sigma_{max}} \tag{4.17}$$

R_1 and R_2 are multiplied by 0.5 as we want to divide the error ranging from [0, 1] into the sub intervals [0 , $R1$),[$R1$, $R2$] and ($R2$,1].

$$|FOU(k)| = \begin{cases} e(k) * R1 & if\ 0 \leq e(k) < R1 \\ e(k) * (R2 - R1) & if\ R1 \leq e(k) \leq R2 \\ \frac{e(k)-R2}{1-R2} & if\ R2 < e(k) \leq 1 \end{cases} \tag{4.18}$$

After the calculation of FOU Eq. (4.19)-(4.29) are used for estimating updated weights and confidence of the DNBC classifier and Algorithm 2 is used to identify the class label of the data samples.

$$\overline{\mu}_k = e(k) + |FOU(k)|/2 \tag{4.19}$$

$$\underline{\mu}_k = e(k) - |FOU(k)|/2 \tag{4.20}$$

$$\overline{\mu}_k^c = ln\left(\frac{1-\overline{\mu}_k}{\overline{\mu}_k}\right),\ \underline{\mu}_k^c = ln\left(\frac{1-\underline{\mu}_k}{\underline{\mu}_k}\right) \tag{4.21}$$

$$C_k = \lambda * \overline{\mu}_k^c + (1-\lambda) * \underline{\mu}_k^c \tag{4.22}$$

$$I_k(\mathbf{x}_i) = \begin{cases} -1 & if \quad P(x_i|\theta) < thr \\ 1 & else \quad P(x_i|\theta) \geq thr \end{cases} \tag{4.23}$$

Here Eq.(4.23) helps to transform the problem to binary class problem, when the sufficient data for abnormal class is unavailable. So the data samples having probability of being in normal state less than predefined threshold (thr) are assigned a label -1.

$$\overline{\mu}_{k+1}^{w_i} = w_i^k * exp(\overline{\mu}_k^c, I_k(\mathbf{x}_i))/\overline{Z}_k \tag{4.24}$$

$$\underline{\mu}_{k+1}^{w_i} = w_i^k * exp(\underline{\mu}_k^c, I_k(\mathbf{x}_i))/\underline{Z}_k \tag{4.25}$$

$$\overline{Z}_k = \sum_{i=1}^{M} w_i^k * exp(\overline{\mu}_k^c, I_k(\mathbf{x}_i)) \tag{4.26}$$

$$\underline{Z}_k = \sum_{i=1}^{M} w_i^k * exp(\underline{\mu}_k^c, I_k(\mathbf{x}_i)) \tag{4.27}$$

$$w_i^{k+1} = \lambda * \overline{\mu}_{k+1}^{w_i} + (1-\lambda) * \underline{\mu}_{k+1}^{w_i} \tag{4.28}$$

$$Sign(\mathbf{x}_i) = \begin{cases} -1 & if\ \mathbf{x}_i < 0 \\ 1 & if\ \mathbf{x}_i > 0 \end{cases} \tag{4.29}$$

Fundamental steps to identify the anomalies using modified fuzzy AdaBoost and DNBC are shown in the Algorithm 2. Figure 4.5 shows the ensemble of k weak DNBC to build a strong classifier.

ALGORITHM 2: Modified Fuzzy AdaBoost and DNBC for anomaly detection

Input: **x**(MTS data), K (Number of classifiers)
Output: $\hat{y}$

1 Compute the degree of dispersion (σ) of all the data belonging to positive class using Eq.(4.13) and Eq.(4.14).;
2 Initialize the values of R1 and R2 using Eq. (4.15), Eq.(4.16) and Eq.(4.17).;
3 Initialize the weights of all the data samples using Eq.(4.12) ;
4 **forall** $k = 1$ *to* k **do**
5 Calculate the parameter θ_k for k^{th} DNBC using Eq.(4.1)-(4.10);
6 Estimate the misclassification error $e(k)$ of the k^{th} step and if $e(k) \geq 0.5$ or $e(k) = 0$ break the loop and exit,;
7 Calculate the $|FOU(k)|$ using the function defined in Eq.(4.18).;
8 Extract the values of $\overline{\mu}_k$ and $\underline{\mu}_k$ using Eq.(4.19) and Eq.(4.20).;
9 Calculate the values of $\overline{\mu}_k^c$ and $\underline{\mu}_k^c$ using Eq.(4.21).;
10 Confidence weight C_k of k^{th} classifier is calculated using Eq.(4.22).;
11 Indicator function for each data x_i can be calculated using Eq.(4.23).;
12 Compute the values of lower ($\underline{\mu}_{k+1}^{w_i}$) and upper membership ($\overline{\mu}_{k+1}^{w_i}$) values for the updated weights using Eq.(4.24) and Eq.(4.25).;
13 Updated weights of data points for the next step is computed using Eq.(4.28).;
14 Combine k weak DNBC using Eq.(4.30) to detect the state ($\hat{y}$) for **x**.

$$\hat{y}_i = Sign(\sum_{k=1}^{K} C_k I_k(\mathbf{x}_i)) \tag{4.30}$$

Where *Sign* in Eq.(4.30) is a signum function that gives -1 as an output when the input is negative otherwise +1.

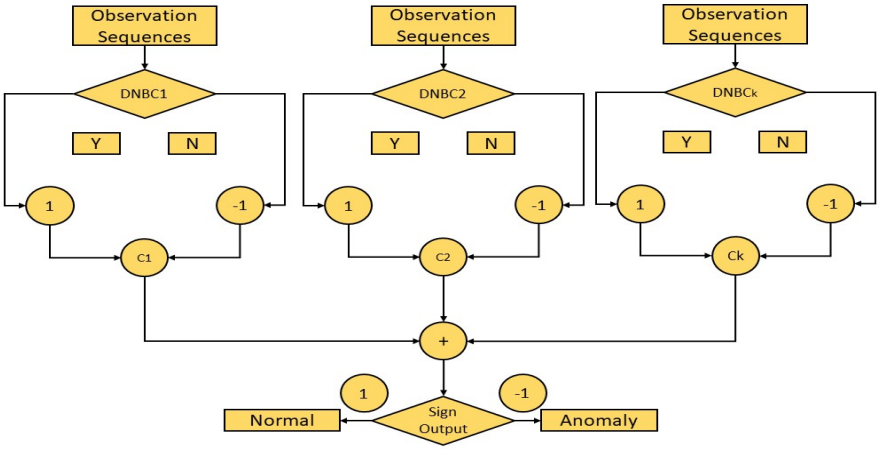


Figure 4.5: *Ensemble of k weak DNBC using fuzzy AdaBoost technique*

4.2.2 Anomaly detection framework

Dynamic Naive Bayesian Classifier and fuzzy AdaBoost algorithm are combined to create a novel anomaly detection framework. Algorithm 2 combines the k weak DNBC classifiers to create the strong DNBC classifier.

4.2.3 Feature extraction

The input to the classifier is given as pairs of trend and value extracted window wise. The <trend value> pairs are extracted in the similar way as described by Bilal et.al [73]. In [73] Piecewise aggregate approximation (PAA) and Least square method was applied to generate the trend value pair feature from the MTS. Instead of using PAA we adapted a fuzzy based method [235] to create the value feature from the MTS.

4.2.3.1 Value features

Feature extraction technique used to create the value feature aims to label the time series data instance with three different tags defined as low(L), medium(M) and high(H) values. Discretization of the time series is performed using k-means clustering method and obtained cluster centers $(V_1), (V_2)$ and (V_3) along with maximum (V_{max}) and minimum (V_{min}) values of the time series are utilized to determine the boundary points

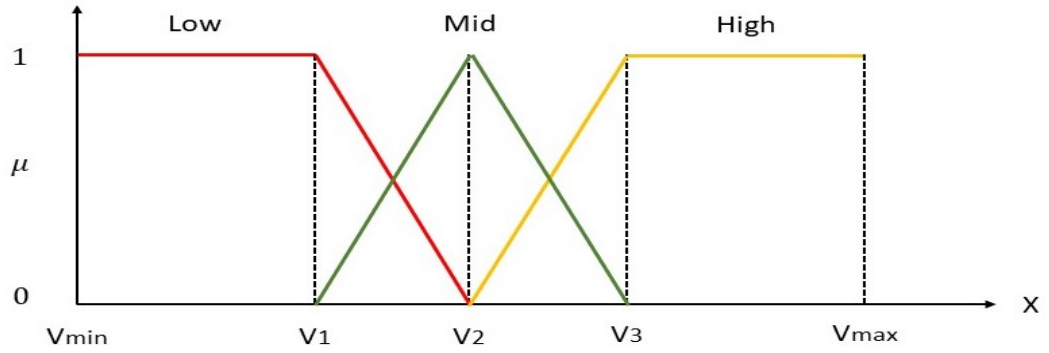


Figure 4.6: *Membership function to calculate the membership values of the fuzzy sets*

[$V_{min}, V_1, V_2, V_3, V_{max}$] of the three fuzzy sets ($F_{Low}, F_{Mid}, F_{High}$). The membership ($\mu$) of the time series data sample to a fuzzy set is computed using Eq. (4.31)-(4.33) and the fuzzy set with the highest membership value is used to label the data instance as low, medium or high. Figure 4.6 shows the membership function required to compute the membership values.

$$\mu_{F_{Low}}(\mathbf{x}) = \begin{cases} 1 & V_{min} \leq \mathbf{x} \leq V_2 \\ \frac{V_2 - \mathbf{x}}{V_2 - V_1} & V_1 \leq \mathbf{x} \leq V_2 \\ 0 & \mathbf{x} \geq V_2 \end{cases} \tag{4.31}$$

$$\mu_{F_{Mid}}(\mathbf{x}) = \begin{cases} 0 & V_{min} \leq \mathbf{x} \leq V_1 \\ \frac{\mathbf{x} - V_1}{V_2 - V_1} & V_1 \leq \mathbf{x} \leq V_2 \\ \frac{V_3 - \mathbf{x}}{V_3 - V_2} & V_2 \leq \mathbf{x} \leq V_3 \\ 0 & V_3 \leq \mathbf{x} \leq V_{max} \end{cases} \tag{4.32}$$

$$\mu_{F_{High}}(\mathbf{x}) = \begin{cases} 0 & \mathbf{x} \leq V_2 \\ \frac{\mathbf{x} - V_2}{V_3 - V_2} & V_2 \leq x \leq V_3 \\ 1 & V_3 \leq \mathbf{x} \leq V_{max} \end{cases} \tag{4.33}$$

Where, $\mathbf{x}$ is a data instance and $\mu_F(\mathbf{x})$ is the membership value of the $\mathbf{x}$ with respect to the fuzzy set F.

4.2.3.2 Trend features

Here we interpret three trends in the time series namely increasing (I), decreasing (D) and constant (C). We used Least square fit [220] method to derive the trend feature from

the time series data by fitting straight line to the given data samples. Extracted trend and value features are combined to create the <trend value> pair feature for the MTS. Table 4.2 shows an example of extracted <trend value> pair from three time series TS_1, TS_2 and TS_3. It is shown in the Table 4.2 how the extracted <trend value> pair of the time series TS_1 changes form $< I, M >$ to $< D, L >$ during the time interval $t1$ to $t5$.

Table 4.2: *<Trend value> pair feature for MTS*

Time Series	TS_1		TS_2		TS_3	
Time Stamp	**Trend**	**Value**	**Trend**	**Value**	**Trend**	**Value**
t_1	I	M	D	H	I	M
t_2	I	H	D	M	D	H
t_3	I	H	I	M	I	H
t_4	I	M	D	L	D	H
t_5	D	L	I	L	I	L

4.3 Experiments and Results

This section presents experiments and results. The efficacy of the proposed method is illustrated through a case study of the detection of stuck pipe anomalies [19] during the oil well drilling process. All experiments are conducted in a computer with Windows 10 operating system, 16 GB RAM and cpu with 3.4 GHz frequency. All scripts are written using MATALB 2018 A version.

4.3.1 Dataset description

To validate the proposed method we inspected the real drilling data of the stuck pipe [19] case occurred in the given oil well rig data provided by the Oil & Natural Gas Corporation Limited (ONGC), India. The training data contains total 101755 data samples recorded at an interval of 7 seconds during the normal drilling operations. Training data is prepared by selecting the normal drilling data for the month of May, July, August and September with the $48909, 32407, 13202$ and 7237 data samples respectively. The data of June month is used to prepare the test data. The test data consists of total 2844 data samples and out of the 2844 data samples, 2519 data instances belong to the normal drilling operation and remaining 325 data samples indicate the stuck pipe complication.

Table 4.3: *Units of different drilling parameters*

S.No.	Parameter	Unit
1	Hookload	$tonfus$
2	Stand Pipe Pressure	$1kgf/cmA$
3	Strokes Per Minute	spm
4	Weight on Bit	$tonfus$
5	Rotation Speed	rpm
6	Flowout	$rel\ \%$
7	Rate of Penetration	$meter/hour$
8	Total Depth	$meter$
9	Bit Depth	$meter$
10	Inlet Density	g/cm^3
11	Outlet Density	g/cm^3

Training data set consists of eleven drilling parameters, each one can be considered as time series. Table 4.3 shows the eleven drilling parameters and their units present in the real drilling data of the given rig. Out of the eleven drilling parameters Weight on bit (WOB), Rate of penetration (ROP), Rotation per minute (RPM) and Strokes per minute (SPM) are selected to prepare the training and test data sets.The selection of these drilling parameter is suggested by a drilling expert.

4.3.2 Initialization of DNBC parameters

DNBC classifiers are trained using the normal drilling data.

$$\begin{bmatrix} a_{nn} & a_{ns} \\ a_{ss} & a_{sn} \end{bmatrix}_{(2,2)} \tag{4.34}$$

During, the initialization of the transition probabilities (a_{ij}) as shown in Eq.(4.34). Transition probabilities from the normal drilling state to normal drilling state (a_{nn}) and the stuck pipe state to the normal drilling state (a_{sn}) are assigned with higher values ((a_{sn})=0.9,(a_{nn})=0.9) and, rest of the transition probabilities (a_{ns}) and (a_{ss}) were initialized with lower ((a_{ns})=0.1,(a_{ss})=0.1)values where, subscripts (s) and (n) denote the stuck pipe and normal drilling states. It is assumed that initially the DNBC model is in normal drilling state. Further, emission probability (b_{ij}) matrices are initialized randomly during the training of the DNBC models. Figure 4.5 shows the schema of

anomaly detection using the proposed method.

4.3.3 Detection of stuck pipe anomalies

Total eight DNBCs were trained using the modified Fuzzy Adaboost algorithm by initializing the parameters (θ) as mentioned in the previous section. Here we trained eight DNBC because after the eighth iteration the misclassification error reached to zero and the Algorithm 2 terminated. The testing is performed on the June month data that contains the stuck pipe anomalies. The experiments are conducted using three windows (W_s) having a size of 50, 100 and 150 data instances respectively. Figure 4.7-4.12 show periods in different drilling parameters when the proposed method successfully identified the occurrence of the stuck pipe anomalies. Red color data samples show the state when the drilling process is in stuck pipe state and blue color instances indicate the normal state of the drilling process. Impact of the stuck pipe complication are straightforward reflected on the drilling parameters as shown in the Figure 4.7-4.12. The best result is found for Fuzzy Adaboost method with the window size of 50 data instances and shown in the Figure 4.7-4.12. Figure 4.8 shows sudden increase in the rate of penetration (ROP) for the data sample range from 2500-2844. For the same range of data samples Figure 4.9 and Figure 4.12 show sudden drop and spikes on the rotation per minute (RPM) and weight on bit (WOB). Further, Figure 4.7, Figure 4.10 and Figure 4.11 shows the drop in hookload (HL), strokes per minute (SPM) and stand pipe pressure (SPP) values for the above mentioned data period. Here, DNBC denotes the individual Dynamic Naive Bayesian Classifier, FAB indicates the classifier created using the ensemble of DNBC classifier using Fuzzy AdaBoost method and AB is the classifier formed using ensemble of DNBC classifier using AdaBoost algorithm. Fuzzy AdaBoost efficiently identified the pattern of stuck pipe and yields satisfactory results as compared to DNBC and AdaBoost DNBC classifiers. Among the three classifiers the proposed method has achieved highest accuracy of 0.97. Performance of all the three classifiers for different window size ($W_s = 50, 100, 150$) are shown in Table 4.4. The model is evaluated for five-fold cross validation setting using nested cross validation technique [35]. DNBC classifier shows lowest accuracy of 0.92. Increase in the window size results degradation of accuracy for

Table 4.4: *Performance of different methods with numerous window size*

Window Size	FAB+DNBC	AB+DNBC	DNBC
50	**0.97 ± 0.27**	0.94 ± 0.17	0.92 ± 0.31
100	0.95 ± 0.16	0.93 ± 0.19	0.90 ± 0.18
150	0.94 ±0.12	0.92 ± 0.15	0.89 ± 0.08

all the three methods. Figure 4.13 shows well progress report of the given well prepared by the drillers during the oil well drilling process. In Figure 4.14 the part surrounded by red box shows date of occurrence of the stuck pipe complication occurred in the given rig which is correctly identified by the proposed method.

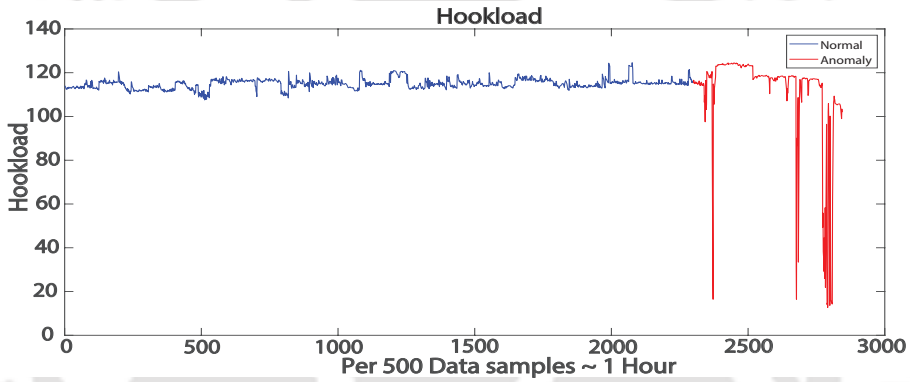


Figure 4.7: *HL parameter during stuck pipe in the given Rig*

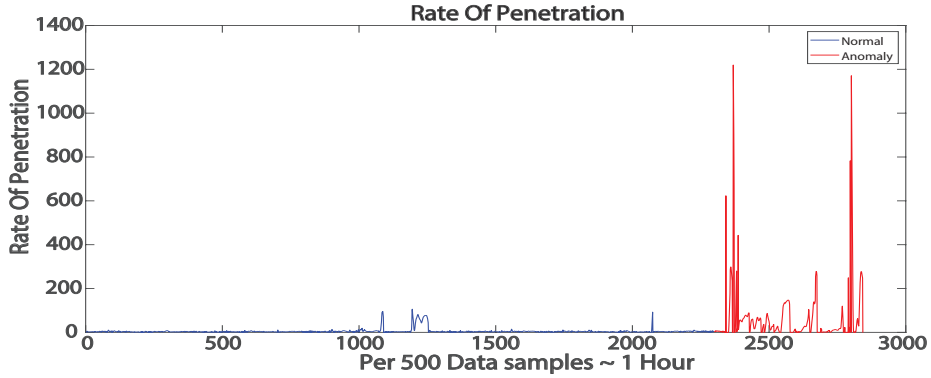


Figure 4.8: *ROP parameter during stuck pipe in the given Rig*

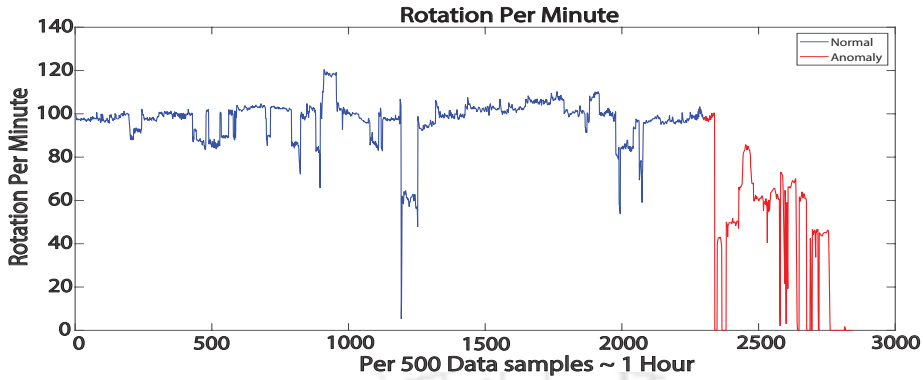


Figure 4.9: *Rotation Per Minute parameter during stuck pipe in the given Rig*

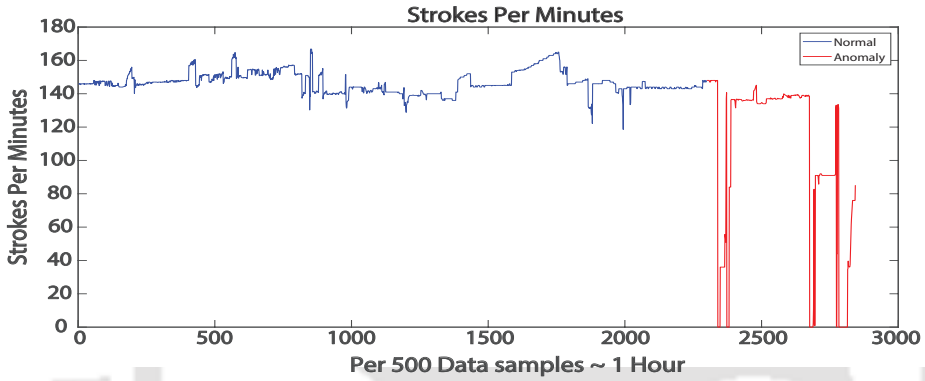


Figure 4.10: *SPM parameter during stuck pipe in the given Rig*

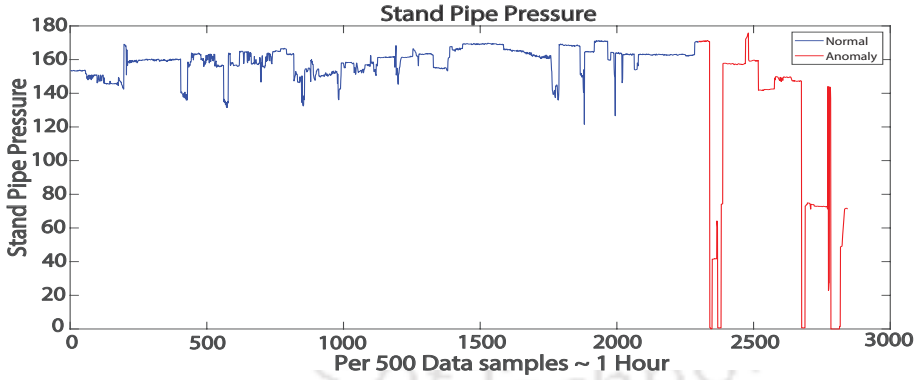


Figure 4.11: *SPP parameter during stuck pipe in the given Rig*

4.3.4 Processing Time

The proposed ensemble model takes 4.5 seconds to identify normal or anomalous state of the data sample in a given window $W_s = 50$ data samples.

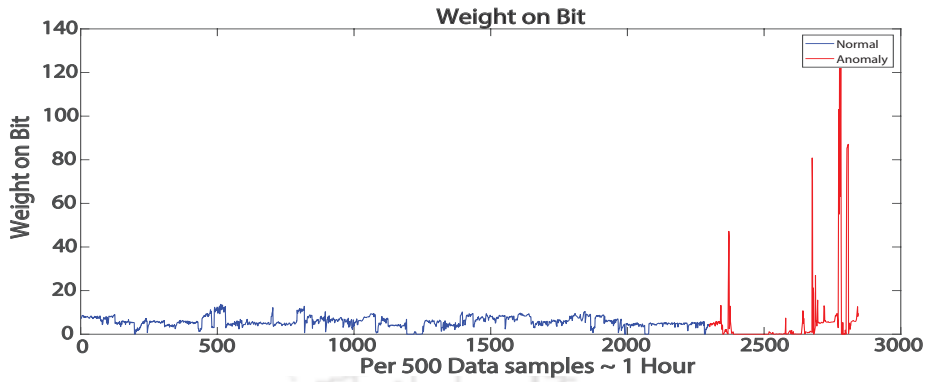
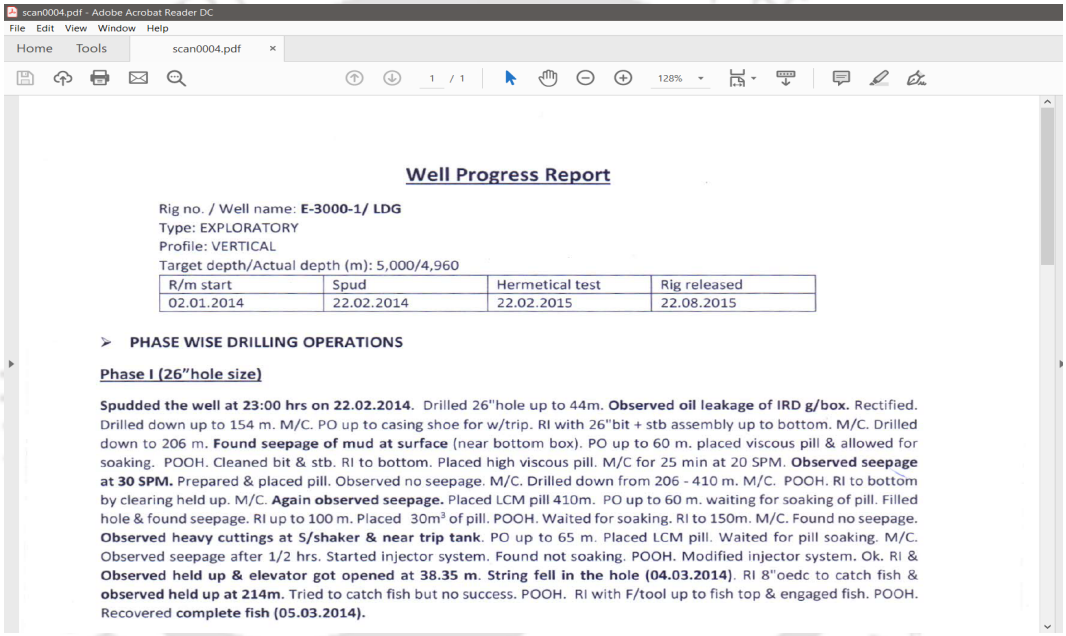


Figure 4.12: *WOB parameter during stuck pipe in the given Rig*



scan0004.pdf - Adobe Acrobat Reader DC

File Edit View Window Help

Home Tools scan0004.pdf

Well Progress Report

Rig no. / Well name: **E-3000-1/ LDG**
Type: EXPLORATORY
Profile: VERTICAL
Target depth/Actual depth (m): 5,000/4,960

R/m start	Spud	Hermetical test	Rig released
02.01.2014	22.02.2014	22.02.2015	22.08.2015

➢ **PHASE WISE DRILLING OPERATIONS**

Phase I (26"hole size)

Spudded the well at 23:00 hrs on 22.02.2014. Drilled 26"hole up to 44m. **Observed oil leakage of IRD g/box.** Rectified. Drilled down up to 154 m. M/C. PO up to casing shoe for w/trip. RI with 26"bit + stb assembly up to bottom. M/C. Drilled down to 206 m. **Found seepage of mud at surface** (near bottom box). PO up to 60 m. placed viscous pill & allowed for soaking. POOH. Cleaned bit & stb. RI to bottom. Placed high viscous pill. M/C for 25 min at 20 SPM. **Observed seepage at 30 SPM.** Prepared & placed pill. Observed no seepage. M/C. Drilled down from 206 - 410 m. M/C. POOH. RI to bottom by clearing held up. M/C. **Again observed seepage.** Placed LCM pill 410m. PO up to 60 m. waiting for soaking of pill. Filled hole & found seepage. RI up to 100 m. Placed 30m³ of pill. POOH. Waited for soaking. RI to 150m. M/C. Found no seepage. **Observed heavy cuttings at S/shaker & near trip tank.** PO up to 65 m. Placed LCM pill. Waited for pill soaking. M/C. Observed seepage after 1/2 hrs. Started injector system. Found not soaking. POOH. Modified injector system. Ok. RI & **Observed held up & elevator got opened at 38.35 m. String fell in the hole (04.03.2014).** RI 8"oedc to catch fish & **observed held up at 214m.** Tried to catch fish but no success. POOH. RI with F/tool up to fish top & engaged fish. POOH. Recovered **complete fish (05.03.2014).**

Figure 4.13: *Well progress report of the given well*

4.4 Summary

This chapter started with the introduction of the HMM and its drawbacks to model the multivariate observation sequences. Later the DNBC classifier was introduced to model the multivariate observation sequences. The fuzzy method based $< trend, value >$ pair features are extracted from the MTS. Initially, the DNBC model was applied to detect the stuck pipe anomalies in real-time drilling data of the given rig. But later, to

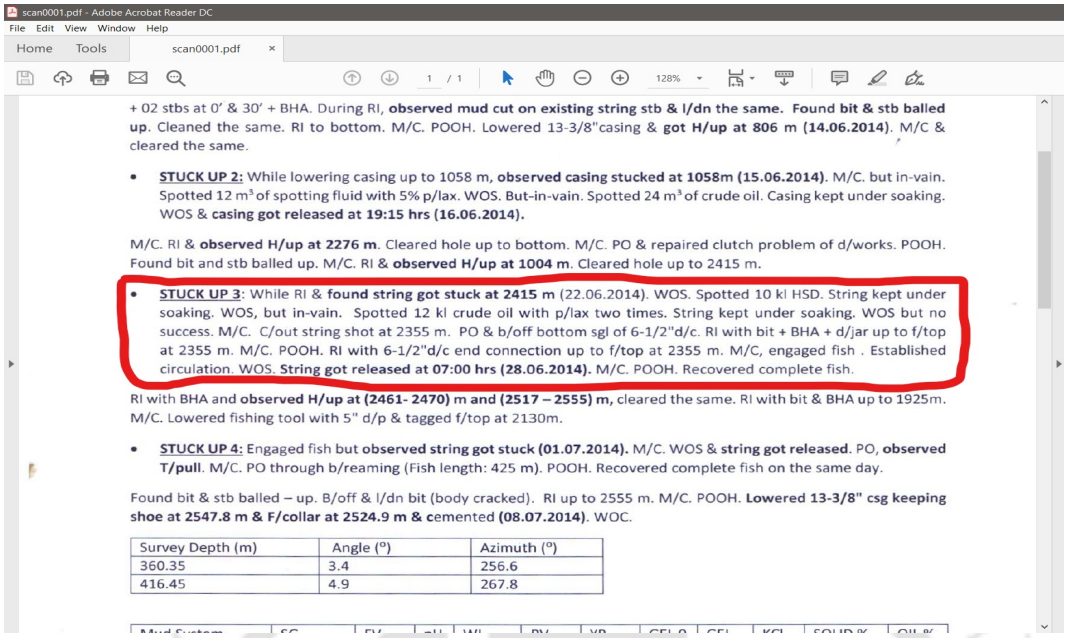


Figure 4.14: *Validation of stuck pipe complication detected by the FAB-DNBC classifier*

enhance the accuracy of the DNBC classifier, the FAB method was used to create the ensemble classifier using the DNBC. The FOU of the FAB method is initialized with the statistical properties of the that data belong to normal drilling operations. The proposed FAB-DNBC classifier successfully identified the stuck pipe anomalies and outperformed the DNBC and AdaBoost DNBC to detect the stuck pipe anomalies. The results are validated using the daily drilling report (DDR) report of the given rig.

Chapter 5

Contextual Anomaly Detection Using Dynamic Bayesian Network

The previous chapter presented the FAB-DNBC classifier to detect the stuck pipe anomaly during the drilling operation. The model developed using the drilling data of the one type of soil may result in the high false alarm rate to detect the stuck pipe complications in the presence of different types of soils. This type of anomaly is also known as contextual anomaly. The first part of this chapter discusses a problem of contextual anomaly during the drilling process and the second part presents a Contextual Dynamic Bayesian Network (CxDBN) that identifies the contextual anomalies on the real time drilling data. The CxDBN is efficient to identify the contextual anomaly during the drilling process and the method outperformed other machine learning models.

5.1 Preliminaries

This section presents the basics of the Contextual anomaly in oil well drilling, Bayesian Network (BN), Dynamic Bayesian Network (DBN). The contextual anomaly is a special kind of anomaly that shows unexpected behavior in the presence of specific context.

5.1.1 Contextual Anomalies in Oil Well Drilling

As mentioned in the previous section, the oil well drilling process requires drilling of deep layers that includes numerous soil formations. The majority of the methods reviewed in the earlier section identified the stuck pipe anomalies without adding the type of soil

information to the models. The behavior of the rig changes concerning the change in soil formation. Figure.5.1 shows a Geo Technical Order (GTO) report prepared to show the change in lithology during the oil well drilling process. The GTO report contains detailed description of the change in the soil layers against the depth of the oil well. The model developed to identify the stuck pipe complication for one type of soil formation may result in high false alarm when applied to detect the stuck pipe condition to a different type of the soil. The data from normal drilling process can be identified as anomalous in the presence of different context Therefore, the context plays a vital role in correctly identifying the anomalies and reduce false alarm rate. Our primary focus is to incorporate the contextual information with the anomaly detection model that is capable of overcoming the drawback of the high false alarm rate caused by the existing stuck pipe identification models.

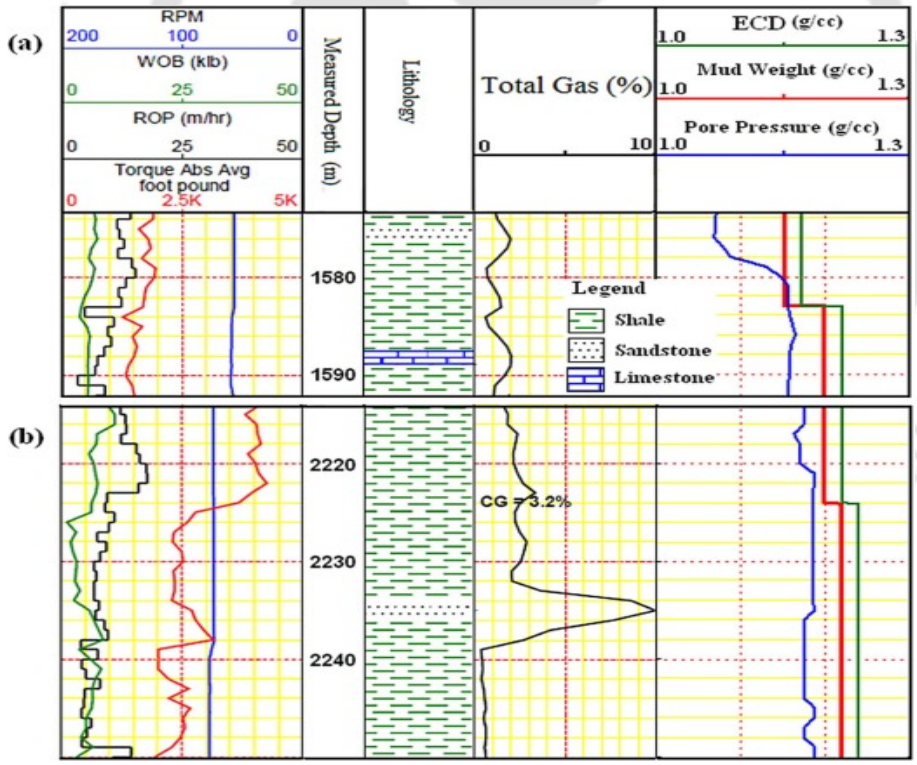


Figure 5.1: *Change in soil formation during oil well drilling [14]*

5.1.2 Bayesian Network (BN)

Bayesian Network [79] is a probabilistic Graphical Model that represents relationship between random variables in terms of DAG. Each node in the directed acyclic graph (DAG) shows the random variable and edges represent relationship between the variables. Figure.5.2 shows BN with random variables A,B,C,D and E. The edges (A → B), (B → (C, D, E)) show dependency of (B) and (C,D,E) over their parents (A) and (B) respectively.

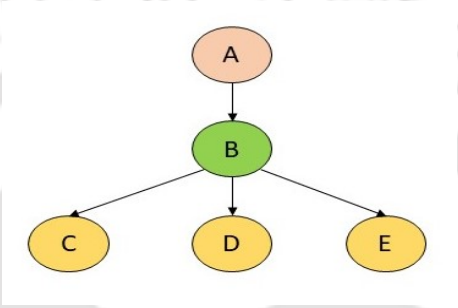


Figure 5.2: *Bayesian Network*

The generalized formula to compute the joint probability can be given as

$$P(X_1, X_2, .., X_R) = \prod_{r=1}^{R} P(X_r | Par(X_r)) \tag{5.1}$$

Where R is the total number of random variables and $Par(X_r)$ is the parent of the attribute X_r. So, the joint probability distribution of Figure 5.2 can be represented as

$$\begin{aligned} P(A, B, C, D, E) = P(A) * P(B|A) * P(C|B) \\ *P(D|B) * P(E|B) \end{aligned} \tag{5.2}$$

5.1.3 Dynamic Bayesian Network (DBN)

Dynamic Bayesian Network [86] or Temporal Bayesian Network is an extension of the Bayesian Network that relates variables of the Bayesian Network over the sequenced time stamps. The DBN is capable of modeling the temporal relationship between the multivariate time series. Figure 5.3 shows a DBN with variable (A,B,C,D,E) that is

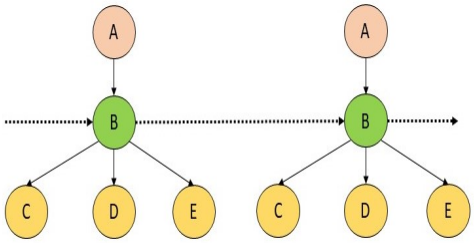


Figure 5.3: *Two time slice Dynamic Bayesian Network*

sliced over two time stamps. The joint probability distribution of the DBN in is given by Eq.(5.3)

$$P(X_1^{1:t}, X_2^{1:t}, .., X_R^{1:t}) = \prod_{t=1}^{T} \prod_{r=1}^{R} P(X_r^t | Par(X_r^t)) \tag{5.3}$$

Parents of X_r^t either come from same time slice or from the previous time slice due to the First Order Markov assumption. The joint distribution of the Figure 5.3 can be expressed by Eq.(5.4).

$$\begin{aligned} P(A^{1:t}, B^{1:t}, C^{1:t}, D^{1:t}, E^{1:t}) = \prod_{t=1}^{T} P(A^{1:t}) * P(B^{1:t}|A^{1:t}) * P(C^{1:t}|B^{1:t}) \\ *P(D^{1:t}|B^{1:t}) * P(E^{1:t}|B^{1:t}) \end{aligned} \tag{5.4}$$

5.2 Methodology

This section presents the methodology used to develop the contextual DBN.

5.2.1 Contextual Dynamic Bayesian Network (CxDBN)

Contextual Bayesian Network is a modified version of the DBN. We configure the causal relationship between the state S and context C. The structure of CxDBN is shown in Figure 5.4. Here S denotes the state, C denotes the context information and E denotes the evidence or observation.

Following assumptions are made to perform the inference with CxDBN:

1. Conditional independence between the context attribute and observation symbols for the given states.

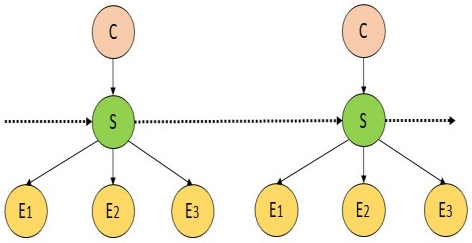


Figure 5.4: *Two time slice Contextual Dynamic Bayesian Network*

2. Mutual independence between the observation symbols.

5.2.2 Inference in CxDBN

Inference in the CxDBN can be done using the following steps:

1. Eq.(5.5) shows the belief of being in some state at time stamp t for the given observations(Obs) and context (C) seen upto time stamp t.

$$Belief(State(t)) = P(State(t)|Obs^{1:t}, C^{1:t}) \tag{5.5}$$

2. Eq.(5.5) can be rewritten as Eq.(5.6)

$$Belief(State(t)) = P(State(t)|Obs^{1:t-1}, Obs^{t}, C^{1:t}) \tag{5.6}$$

3. After further modification Eq.(5.6) is simplified to Eq.(5.7)

$$\begin{aligned} Belief(State(t)) = \alpha * P(Obs^{t}|State(t), Obs^{1:t-1}, C^{1:t}) \\ *P(State(t)|Obs^{1:t-1}, C^{1:t}) \end{aligned} \tag{5.7}$$

   Where α is the normalization constant.

4. If mutual independence exists between the observations then first term in the Eq.(5.7) can be written as

$$P(Obs^{t}|State(t), Obs^{1:t-1}, C^{1:t}) = \prod_{l=1}^{L} P(Obs_l^{t}|State(t)) \tag{5.8}$$

   Where L is the number of observations.

Table 5.1: *Transition probability table for different contexts*

Contexts	State(t-1)	State(t)		
		$State_i$	$State_j$	
C_i	$State_i$	a_{ii}	a_{ij}	T_{C_i}
C_i	$State_j$	a_{ji}	a_{jj}	
C_j	$State_i$	a_{ii}	a_{ij}	T_{C_j}
C_j	$State_j$	a_{ji}	a_{jj}	

5. Second term in the Eq.(5.7) is a simple one step prediction from the state at t-1 time stamp and can be written as

$$P(State(t)|Obs^{1:t}, C^{1:t}) = \sum_{State(t-1)} P(State(t)|State(t-1), C^t) * Belief(State(t-1)) \tag{5.9}$$

6. Finally the belief at time stamp t can be computed by Eq.(5.10). Table 5.1 denotes transition probability table for the different contexts.

$$Belief(State(t)) = \prod_{l=1}^{L} P(Obs_l^t|State(t)) * P(State(t)|Obs^{1:t}, C^{1:t}) \tag{5.10}$$

Where a_{ij} is the transition probability from state i to state j, C_i is the context i, T_{C_i} is the transition probability table for the context C_i.

For example let us consider a dynamic system with two states Normal and Anomalous. The transition probability table for the given three context C1, C2 and C3 can be assumed as given in Table 5.2. Table 5.3 shows assumed emission probability table with two states and three observations Obs_1, Obs_2 and Obs_3 respectively. Assume the initial Belief $Belief(State(t = 0))$ is $[0.65, 0.35]$ i.e. $P(State_0 = Normal) = 0.65$

Table 5.2: *Transition probability table*

Context	State(t-1)	State(t)		
		Normal	**Anomaly**	
C1	**Normal**	0.85	0.15	T_{C_1}
C1	**Anomaly**	0.21	0.79	
C2	**Normal**	0.80	0.20	T_{C_2}
C2	**Anomaly**	0.05	0.95	
C3	**Normal**	0.90	0.10	T_{C_3}
C3	**Anomaly**	0.15	0.85	

Table 5.3: *Emission probability table*

Observation		$State^t$	
	Symbols	**Normal**	**Anomaly**
Obs_1^t	**L**	0.55	0.14
	M	0.15	0.01
	H	0.30	0.85
Obs_2^t	**L**	0.85	0.35
	M	0.14	0.64
	H	0.01	0.01
Obs_3^t	**L**	0.65	0.15
	M	0.34	0.83
	H	0.01	0.02

and $P(State_0 = Anomaly) = 0.35$. If at time stamp $t = 1$ the emission symbols $[Obs_1^1, Obs_2^1, Obs_3^1]$ are $[M, L, M]$ and the context is C1. Then the belief at $t = 1$ using Eq.(5.10) is computed as

$$Belief(State_1 = Normal) = [(0.15 * 0.85 * 0.34) * (0.85 * 0.65 + 0.35 * 0.21)]$$
$$= 0.0271$$
$$Belief(State_1 = Anomaly) = [(0.01 * 0.35 * 0.83) * (0.65 * 0.15 + 0.35 * 0.79)]$$
$$= 0.0108$$
$$\alpha = 1/(0.0108 + 0.0271) = 1/0.0379 = 26.38$$
$$Belief(State_1 = Normal) = 0.0271 * 26.38 = 0.71$$
$$Belief(State_1 = Anomaly) = 0.0108 * 26.38 = 0.29$$

Now at the time stamp $t = 2$ if the emitted observations $[Obs_1^2, Obs_2^2, Obs_3^2]$ are $[H, M, L]$ and the context is C3. Then the belief at $t = 2$ using Eq.(5.10) is computed as follows

$$Belief(State_2 = Normal) = [(0.30 * 0.14 * 0.65) * (0.71 * 0.90 + 0.29 * 0.15)]$$
$$= 0.0186$$
$$Belief(State_2 = Anomaly) = [(0.30 * 0.14 * 0.65) * (0.71 * 0.10 + 0.29 * 0.85)] =$$
$$0.0086$$
$$\alpha = 1/(0.0186 + 0.0086) = 1/0.0272 = 36.76$$
$$Belief(State_2 = Normal) = 0.0186 * 36.76 = 0.68$$
$$Belief(State_2 = Anomaly) = 0.0086 * 36.76 = 0.32$$

But if at $t = 2$ for the same emitted observations the Context is C2 instead of C3 then the Belief would be

$$Belief(State_2 = Normal) = [(0.30 * 0.14 * 0.65) * (0.71 * 0.80 + 0.29 * 0.05)]$$
$$= 0.0156$$
$$Belief(State_2 = Anomaly) = [(0.30 * 0.14 * 0.65) * (0.71 * 0.20 + 0.29 * 0.95)]$$
$$= 0.0114$$
$$\alpha = 1/(0.0156 + 0.0114) = 1/0.027 = 37.03$$
$$Belief(State_2 = Normal) = 0.0156 * 37.03 = 0.58$$
$$Belief(State_2 = Anomaly) = 0.0114 * 37.03 = 0.42$$

From the above example it is clear how the inclusion of context information changes the belief of the dynamic system at the time $t = 2$ from [0.68 0.32] to [0.58 0.42] when the context changes from C3 to C2. In our experiments the drilling parameters are the observations and types of the soil is considered as the contextual attribute. Fig.(5.5) shows two-time slice CxDBN designed using the evidence (drilling parameters) and contextual information(types of soil). The drilling state denotes normal drilling or stuck pipe condition while the drilling operation.

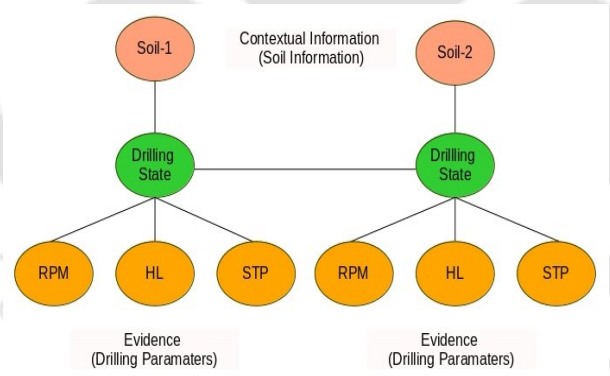


Figure 5.5: *Two time slice Contextual Dynamic Bayesian Network for Drilling Process*

5.3 Experiments and Results

This section describes the experiments and results from real-time drilling data provided by the Oil & Natural Gas Corporation Limited (ONGC), India. All experiments are

conducted in a computer with Windows 10 operating system, 16 GB RAM and cpu with 3.4 GHz frequency. All scripts are written using MATALB 2018 A version.

5.3.1 Data set Description

To test the efficacy of the proposed method, we used the real-time drilling data (RTDD) of the given rig located in the state of Assam, India. The drilling started on 01/01/2016 and ended on 02/02/2018. Training data contains eleven drilling parameters, and each parameter can be considered as the time series. Table 5.4 shows the eleven drilling parameters and their units present in the RTDD of the given rig. Among the eleven drilling parameters Rotation Per Minute (RPM), Stand Pipe Pressure (STP), and Hookload (HL) are considered for preparing the training and test data. The contextual information is provided manually from the GTO report of the given rig.

Table 5.4: *Units of different drilling parameters in given rig*

S.No.	Parameter	Unit
1	Hookload	$tonfus$
2	Stand Pipe Pressure	$1kgf/cmA$
3	Strokes Per Minute	spm
4	Weight on Bit	$tonfus$
5	Rotation Speed	rpm
6	Flowout	$rel\ \%$
7	Rate of Penetration	$meter/hour$
8	Total Depth	$meter$
9	Bit Depth	$meter$
10	Inlet Density	g/cm^3
11	Outlet Density	g/cm^3

The training data contain a total of 2435830 data samples and 525800 test data samples. Out to the 243583 training data samples 1020000, 513490, and 902340 data samples belong to the drilling of the soil-1, soil-2, and soil-3, respectively. In the test data 190000, 51990, and 283810 data samples belong to the drilling of the soil-1, soil-2, and soil-3, respectively. The drilling data of the soil-2 contain the data of the normal drilling process (35990 data samples), and the stuck pipe case (16000 data samples) occurred in the given rig. Table 5.5 shows the distribution of the training and test data

samples for the given rig.

Table 5.5: *Training and Test Data samples for given rig*

Soil Type	Train Data Samples	Test Data Samples	Class
Soil-1	1020000	190000	Normal
Soil-2	513490	51990	(35990) + Stuck Pipe(16000)
Soil-3	902340	283810	Normal
Total	2435830	525800	

5.3.2 Feature Extraction

We used the same procedure as used in the sub-section 4.2.3 of the chapter 4 to extract the <value> pair features from the MTS.

5.3.3 Detection of Contextual Anomaly (Stuck Pipe)

ALGORITHM 3: Identification of contextual anomaly in drilling using CxDBN

Input: SCADA Data,Train Drilling Data (X_{Train}), Test Drilling Data (X_{Test}), Contextual Information, Hookload (HL), Rotation Per Minute (RPM), Stand Pipe Pressure (STP), Drilling Activity

Output: Drilling State ($State$)

1 Select the SCADA data with drilling activity;
2 Select the HL, RPM, STP parameters from the SCADA data to prepare the training data X_{Train};
3 Extract $< value >$ pair features from the training data X_{Train};
4 Train CxDBN using the training features and contextual information from GTO.;
5 For testing select test data X_{Test} from the SCADA data with drilling activity;
6 Select the HL,RPM, STP parameters from the X_{Test} data to extract $< trend, value >$ pair features.;
7 Identify the Drilling State ($State$) of X_{Test} using the trained CxDBN and contextual information from GTO;
8 Return Drilling State ($State$);

Algorithm.3 shows the procedure of detection of the contextual anomaly during the oil well drilling. Initially the SCADA data belong to the drilling activity is selected and HL, RPM and STP parameters are selected to prepare the training (X_{Train}) and test

data (X_{Test}). The contextual information i.e. the type of soil is collected from the GTO report of the given rig. The feature extraction technique described previously was used to train and test the CxDBN. The trained CxDBN is used to detect the drilling state ($State$) of the test data (X_{Test}).

Maximum-likelihood approach [86] is used to learn the parameters (θ) and probabilities of the CxDBN. The major aim of the proposed method is to correctly identify the stuck pipe anomalies and prevent the model from detecting the normal drilling data to be triggered as anomalous in presence of the different context i.e., the soil formation. The experiments are conducted using three windows (W_s) 10, 50, and 100 data samples, respectively. Each data sample is recorded at an interval of seven seconds. Figures 5.6-5.8 shows periods when the proposed model successfully identified the occurrence of the stuck pipe complication for the $W_s = 10$ data samples. The region surrounded by pink, green and orange rectangles in the Figures 5.6-5.8 show the average values of the HL, RPM and STP parameters in a given window during the drilling operation of soil-1, soil-2 and soil-3 formations in the drilling rig site for the given rig. Red color shows the data sample when drilling process enters to the stuck pipe state and blue color data sample indicates the normal state of the drilling process. The impact of the stuck pipe anomalies is clearly reflected on the drilling parameters, as shown in the Figures 5.6-5.8. Figure 5.6 shows a sudden increase in the value of HL for the data sample of range 22600-24200 and for the same data range value of RPM parameter decreases as shown in the Figure 5.7. Figure 5.8 indicates an increase in the value of STP parameter for the same data period mentioned above. The model successfully identified the occurrence of the stuck pipe complication during the drilling of the soil-2 as shown in the Figures 5.6-5.8. The Belief of being in normal state estimated by the CxDBN is shown in Figure 5.9. The values of Belief goes down during the occurrence of the stuck pipe problem for the data range of 22600-24200. Here we put Belief=0.5 as the threshold below which the Belief of being in normal states is considered as the anomaly. A sudden drop in the value of the Belief in normal states in the Figure 5.9 after the data sample 19000 indicates a change in the context. We compare the performance of

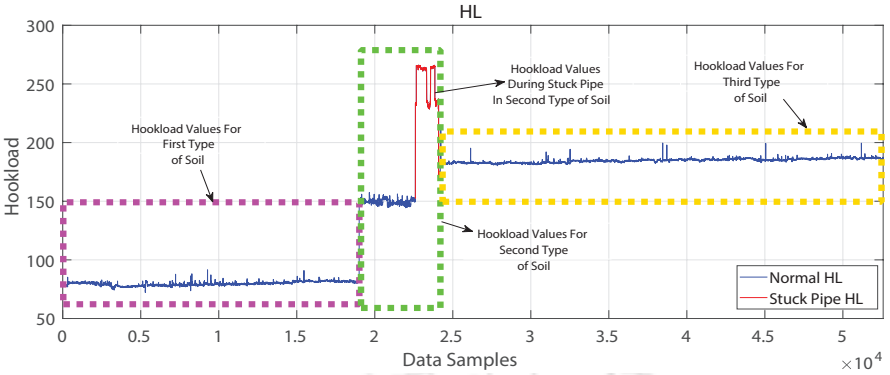


Figure 5.6: *Hookload parameter during drilling of different soils*

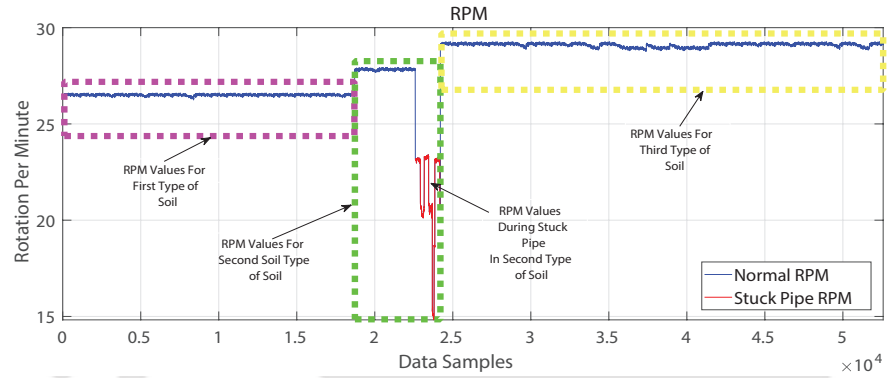


Figure 5.7: *RPM parameter during drilling of different soils*

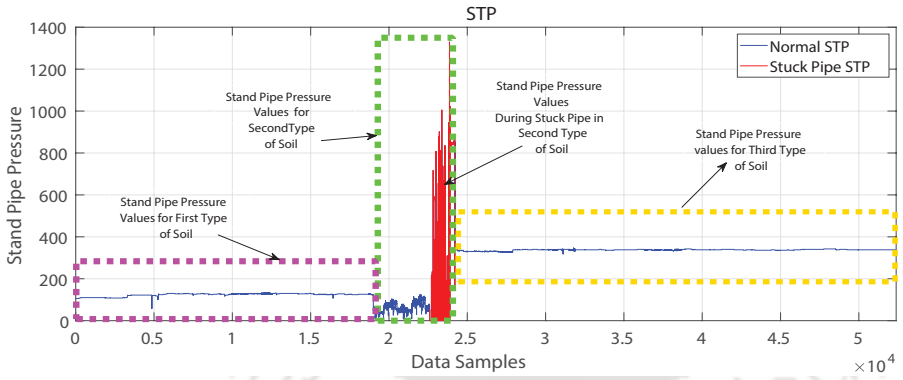


Figure 5.8: *STP parameter during drilling of different soils*

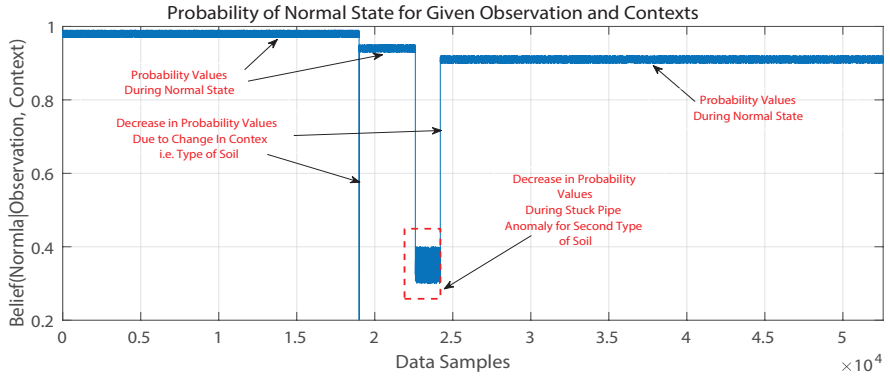


Figure 5.9: *Belief (Normal State) estimated by Contextual DBN*

the four models i.e., One-Class Support Vector Machine (OSVM) [56], Dynamic Naive Bayesian Classifier (DNBC) [142], DBN [86] and the CxDBN to identify the stuck pipe and contextual anomalies for the oil well drilling process. The models are evaluated for five-fold cross validation setting using nested cross validation technique [35]. Table 5.6 shows the accuracy obtained by the aforementioned models for the different size of the window (W_s). The proposed method shows the highest accuracy of 0.96 for W_s=10 data samples. However, the individual DBN classifier trained without adding the contextual information shows the accuracy of 0.76 for the same window size. For the same window size the accuracy of the OCSVM and DNBC is 0.66 and 0.72 respectively. The DBN, OSVM, and DNBC failed to model the change in the context during the drilling process and detects the data sample that belonging to the normal drilling process for the different contexts as the anomalous data. Unlike the OSVM, DBN and DNBC the proposed method correctly identified the change in context and efficiently prevented the high false alarm rate by identifying the normal drilling data in the normal state in the presence of different soil formations. It is clear from Table 5.6 the proposed method outperformed the DBN classifier. The CxDBN is efficient to model the temporal and contextual behavior of the MTS stored in the SCADA system for the oil well drilling process. Further, the CxDBN takes 1.5 seconds to identify normal or anomalous state of the data samples present in a given window ($W_s = 50$ data samples).

Table 5.6: *Performance of different methods with numerous window size*

Models	Window Size	Accuracy(%)
OCSVM	10	0.66±0.17
	50	0.64±0.16
	100	0.62±0.15
DNBC	10	0.72±0.21
	50	0.71±0.18
	100	0.70±0.14
DBN	10	0.76±0.11
	50	0.73±0.13
	100	0.72±0.18
CxDBN	10	**0.96±0.07**
	50	0.95±0.09
	100	0.95±0.08

5.4 Summary

This chapter started with the introduction of the contextual anomaly in oil well drilling process. It presented the CxDBN model that is efficient to identify the contextual anomalies during the oil well drilling. The efficacy of the model is shown through identification of the stuck pipe anomalies in the real drilling data. At last the performance of the model is compared with three other models and the CxDBN outperformed the three models.

Chapter 6

Conclusions and Future Work

The extraction of oil and gas from the deep formation is a complex process that requires continuous supervision to prevent the hazardous situations during the drilling. The various drilling problems such as the stuck pipe complication, washout, and poor hole cleaning causes enormous loss to the oil industries. The supervision process of the oil well drilling involving human operator is tedious and usually results in high false alarm rate. The recent advancement of sensor-equipped technology attracted the oil industry to provide automated supervision of the oil well drilling process. The various drilling parameters that are recorded during the oil well drilling are stored in a SCADA system. The data stored in the SCADA system can be utilized to develop systems to automate supervision of the drilling process by applying ML and AI techniques. With the aim to replace the human-based supervision of the drilling process by the AI-based monitoring, this research has made primarily three contributions to solve the various challenges as discussed in the chapter 1. The first challenge was to identify the various oil well drilling activities associated with the drilling data recorded in the SCADA database. The drilling process involves approximately 20 drilling activities. The identification of the drilling activities are helpful to develop the various models to deal with the various drilling problems. To achieve this aim, the two-level stacked classifier is presented that combines the fuzzy rule-based classifier and random forest classifier in stacked layers to identify the various drilling activities. The FRB classifier efficiently models the uncertainty present in the drilling data. The model is tested on the real drilling data of four wells, and

the model shows very high accuracy to detect the oil well drilling activities. The model also outperformed the decision tree, random forest, and multiclass support vector machine classifiers. In addition, the proposed classifier also provides detailed analysis of the amount of time spend to perform the specific drilling activity in one complete cycle of the drilling process. The created report is helpful for further analysis of establishment of other rigs in the similar rig site.

After the identification of the various oil well drilling activities using the two-level stacked classifier, the second challenge is to develop the machine learning model that can identify the stuck pipe complication in a real-time drilling data. To achieve this objective the FAB-DNBC classifier is developed that successfully identified the stuck pipe complication on the drilling data of the given rig. The FOU parameter of the FAB was initialized using the statistical properties of the data that belong to normal drilling operations. The model outperformed the AdaBoost DNBC, and OSVM classifiers. The result of the model is validated using the DDR report of the tested rig. The proposed model is tested only to identify stuck pipe cases that happen during the drilling of one type of soil. The same model has been found to suffer from high false alarm rate when employed to identify the stuck pipe cases during drilling of the different type of soil. To overcome this situation a novel contextual dynamic bayesian network (CxDBN) is developed and tested to identify the stuck pipe cases from different types of soils.

As mentioned in the chapter 5, the model developed to identify the stuck pipe anomaly may result in high false alarm rate in the presence of different types of soils and this behavior is due to presence of contextual anomalies. To detect the contextual anomalies the CxDBN classifier is developed that integrates the contextual information and DBN to identify the contextual anomaly in the oil well drilling data. The CxDBN model efficiently identified the stuck pipe anomaly on the real-time drilling data of the given rig in the presence of different types of soil. The model outperformed the DNBC, DBN, and OSVM classifiers to identify the contextual anomalies in the given rig.

The evaluation of the proposed models has been performed on the real drilling data. The two-level classifier has been successfully deployed in the oil well drilling site of

Assam in India. The models have shown excellent performance to achieve the desired objectives, and in the future, we would like to deploy the other two models in the drilling sites.

6.1 Future Work

Oil well drilling activity involves nearly twenty different drilling activities. The proposed two-level stacked classifier proposed in the chapter 3 is capable of identifying a total of ten drilling activities. The model can be extended to identify the remaining ten drilling activities that are not included in this thesis. The proposed methods to identify the stuck pipe anomalies in the chapter 5 and chapter 4 require the additional feature pre-processing step i.e., extraction of the $< trend, value >$ pair features which may cause delay during the real-time monitoring of the drilling process. In future, advanced deep learning techniques such as life long learning and knowledge distillation methods can be explored to design adaptive and lightweight deep models that can identify the anomalies in real drilling data. A concept of continual learning can be applied to resolve an issue of contextual anomaly during drilling process. The continual learning-based model are adaptive and easily overcome the problem of catastrophic forgetting caused due to change in context. Few-shot learning and self-supervised techniques for time series can be explored to build the AI-model in presence of less training data which is a very common situation while developing AI-based models for oil well drilling specifically while identifying the stuck pipe cases. Another future work would be to design of a unified framework that combines the proposed models and these lightweight deep models to identify the stuck pipe complications.

Further, the efficacy of the proposed techniques in this thesis is validated using real-time drilling data of Assam region in India. In future, we would like to explore the applicability of the methods to identify the drilling complications in the oil fields located in other regions. The other significant work that can be considered in future is predicting ahead the faults rather than detecting faults.

Publications

Patents

1. Senthilmurugan S, Rashmi Dutta Baruah, Munawar A. Shaik, **Achyut Mani Tripathi**, Senthil Selvaraju, Viswanth Ramba, Bala Kumara Vignesh M, Amol Musale, Gauba SK, Samal KB, "Decision Support System for Oil and Gas Well Drilling". (**Application No. 201911040595**)

Journal Papers

1. Achyut Mani Tripathi, Rashmi Dutta Baruah and Senthilmurugan S "Oil Well Drilling Activities Recognition Using a Hierarchical Classifier", Journal of Petroleum Science and Engineering, Elsevier, Volume 196, Pages 107883, January 2021.

Conference Papers

1. **Achyut Mani Tripathi** and Rashmi Dutta Baruah, "Contextual Anomaly Detection in Time Series Using Bynamic Bayesian Network", Asian Conference on Intelligent Information and Database Systems,Phuket, Thailand, March 26-29, 2020.

2. **Achyut Mani Tripathi**, Rashmi Dutta Baruah, "Anomaly Detection in Multivariate Time Series Using Fuzzy AdaBoost and Dynamic Naive Bayesian Classifier", Accepted in IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2019), Bari, Italy, October 6-9, 2019.

Other Publications

The following publications, co-authored by the author of this thesis, are not included in the thesis:

1. **Achyut Mani Tripathi**,"Enhancing Multivariate Time Series Classification Using Long Short Term Memory and Evidence Feed Forward HMM",International Joint Conference on Neural Networks (IJCNN), July 19-24, 2020, Glasgow,UK

2. **Achyut Mani Tripathi**, Rashmi Dutta Baruah,"Multivariate Time Series Classification With An Attention-Based Multivariate Convolutional Neural Network", IEEE International Joint Conference on Neural Networks (IJCNN), July 19-24, 2020, Glasgow,UK

3. **Achyut Mani Tripathi** and Rashmi Dutta Baruah, "Acoustic Event Detection Using Fuzzy Integral Ensemble and Oriented Fuzzy Local Binary Pattern Encoded CNN", In FUZZ-IEEE 2020 under WCCI, July 19 – 24, 2020, Glasgow, UK, pp. 1-8.

4. **Achyut Mani Tripathi** and Rashmi Dutta Baruah, "Incremental Cauchy Non-negative Matrix Factorization and Fuzzy Rule-Based Classifier for Acoustic Source Separation", Accepted In 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), New Orleans, USA, June 23-26, 2019.

5. **Achyut Mani Tripathi** and Rashmi Dutta Baruah, "Acoustic Event Classification Using Cauchy Non-negative Matrix Factorization and Fuzzy Rule-Based Classifier", in Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), July 9-12, 2017, Naples, Italy, pp.1-6.

6. **Achyut Mani Tripathi**, Rashmi Dutta Baruah, "Anomaly Detection in Data Streams Based on Graph Coloring Density Coefficients", in Proceedings of IEEE Symposium Series on Computational Intelligence (SSCI), Dec 6-9, 2016, Athens, Greece, pp.1-7.

7. **Achyut Mani Tripathi**, Diganta Baruah and Rashmi Dutta Baruah, "Acoustic Event Classification Using Ensemble of One-Class Classifiers for Monitoring Application", in Proceedings of IEEE Symposium Series on Computational Intelligence (SSCI), Dec 7-10, 2015, Cape Town, South Africa, pp.1681-1686.

8. **Achyut Mani Tripathi**, Diganta Baruah and Rashmi Dutta Baruah, "Acoustic Sensor Based Activity Recognition Using Ensemble of One-Class Classifiers", in Proceedings of IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS) , Dec 1-3, 2015, Duoai, France, pp.1-7.

9. Sonia, **Achyut Mani Tripathi**, Rashmi Dutta Baruah, S.B.Nair, "Ultrasonic Sensor-Based Human Detector Using One-Class Classifiers", in Proceedings of IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS) , Dec 1-3 ,2015, Duoai, France, pp.1-6.

References

[1] *Offshore Drilling*. [Online]. Available: https://oilprice.com/Energy/Crude-Oil/Big-Oil-Is-Heading-Offshore.html

[2] *Oil Well Rig*. [Online]. Available: https://en.wikipedia.org/wiki/Drilling_rig

[3] *Onshore Drilling*. [Online]. Available: https://www.wellcontrol.com.au/index.php/onshore-drilling

[4] *Petrowiki*. [Online]. Available: https://petrowiki.org/Drilling_problems

[5] A. K. Abbas, N. A. Al-haideri, and A. A. Bashikh, "Implementing artificial neural networks and support vector machines to predict lost circulation," *Egyptian Journal of Petroleum*, 2019.

[6] A. K. Abbas, A. A. Bashikh, H. Abbas, and H. Q. Mohammed, "Intelligent decisions to stop or mitigate lost circulation based on machine learning," *Energy*, vol. 183, pp. 1104–1113, 2019.

[7] S. A. Agarwal, N. Agarwal, *et al.*, "Auto-release drill collars," in *SPE Indian Oil and Gas Technical Conference and Exhibition*. Society of Petroleum Engineers, 2008.

[8] C. C. Aggarwal, "Outlier ensembles: position paper," *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 49–58, 2013.

[9] C. C. Aggarwal, "Outlier analysis," in *Data mining*. Springer, 2015, pp. 237–263.

[10] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in *Proceedings of the 29th international conference on Very large data bases-Volume 29*. VLDB Endowment, 2003, pp. 81–92.

[11] C. C. Aggarwal and S. Y. Philip, "An effective and efficient algorithm for high-dimensional outlier detection," *The VLDB journal*, vol. 14, no. 2, pp. 211–221, 2005.

[12] C. C. Aggarwal and S. Sathe, *Outlier ensembles: An introduction*. Springer, 2017.

[13] M. M. Ahad, B. Mahour, and K. Shahbazi, "Application of machine learning and fuzzy logic in drilling and estimating rock and fluid properties."

[14] M. A. Ahmed, O. A. Hegab, and A. Sabry, "Early detection enhancement of the kick and near-balance drilling using mud logging warning sign," *Egyptian Journal of Basic and Applied Sciences*, vol. 3, no. 1, pp. 85–93, 2016.

[15] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data mining and knowledge discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[16] S. Al Gharbi, M. Ahmed, S. ElKatatny, *et al.*, "Use metaheuristics to improve the quality of drilling real-time data for advance artificial intelligent and machine learning modeling. case study: Cleanse hook-load real-time data," in *Abu Dhabi International Petroleum Exhibition & Conference*. Society of Petroleum Engineers, 2018.

[17] M. B. Al-Zoubi, "An effective clustering-based approach for outlier detection," *European Journal of Scientific Research*, vol. 28, no. 2, pp. 310–316, 2009.

[18] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2016, pp. 195–200.

[19] A. A. Alshaikh, M. K. Albassam, S. H. Al Gharbi, A. S. Al-Yami, *et al.*, "Detection of stuck pipe early signs and the way toward automation," in *Abu Dhabi International Petroleum Exhibition & Conference*. Society of Petroleum Engineers, 2018.

[20] A. Ambrus, P. Ashok, D. Ramos, A. Chintapalli, A. Susich, T. Thetford, B. Nelson, M. Shahri, J. McNab, M. Behounek, *et al.*, "Self-learning probabilistic detection and alerting of drillstring washout and pump failure incidents during drilling operations," in *IADC/SPE Drilling Conference and Exhibition*. Society of Petroleum Engineers, 2018.

[21] J. T. Andrews, E. J. Morton, and L. D. Griffin, "Detecting anomalous data using auto-encoders," *International Journal of Machine Learning and Computing*, vol. 6, no. 1, p. 21, 2016.

[22] F. Angiulli, S. Basta, and C. Pizzuti, "Distance-based detection and prediction of outliers," *IEEE transactions on knowledge and data engineering*, vol. 18, no. 2, pp. 145–160, 2005.

[23] F. Angiulli and F. Fassetti, "Detecting distance-based outliers in streams of data," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM, 2007, pp. 811–820.

[24] F. Angiulli and F. Fassetti, "Very efficient mining of distance-based outliers," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM, 2007, pp. 791–800.

[25] F. Angiulli and F. Fassetti, "Distance-based outlier queries in data streams: the novel task and algorithms," *Data Mining and Knowledge Discovery*, vol. 20, no. 2, pp. 290–324, 2010.

[26] F. Angiulli and C. Pizzuti, "Fast outlier detection in high dimensional spaces," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2002, pp. 15–27.

[27] A. Arnaout, P. O'Leary, B. Esmael, and G. Thonhauser, "Distributed recognition system for drilling events detection and classification," *International Journal of Hybrid Intelligent Systems*, vol. 11, no. 1, pp. 25–39, 2014.

[28] I. Assent, P. Kranen, C. Baldauf, and T. Seidl, "Anyout: Anytime outlier detection on streaming data," in *International Conference on Database Systems for Advanced Applications*. Springer, 2012, pp. 228–242.

[29] H. H. Avilés-Arriaga and L. E. Sucar, "Dynamic bayesian networks for visual recognition of dynamic gestures," *Journal of Intelligent & Fuzzy Systems*, vol. 12, no. 3, 4, pp. 243–250, 2002.

[30] K. Bach, O. E. Gundersen, C. Knappskog, and P. Öztürk, "Automatic case capturing for problematic drilling situations," in *International Conference on Case-Based Reasoning*. Springer, 2014, pp. 48–62.

[31] M. Bai, X. Wang, J. Xin, and G. Wang, "An efficient algorithm for distributed density-based outlier detection on big data," *Neurocomputing*, vol. 181, pp. 19–28, 2016.

[32] V. Barnett and T. Lewis, *Outliers in statistical data*. Wiley, 1974.

[33] R. D. Baruah and P. Angelov, "Dec: Dynamically evolving clustering and its application to structure identification of evolving fuzzy models," *IEEE transactions on cybernetics*, vol. 44, no. 9, pp. 1619–1631, 2013.

[34] S. D. Bay and M. Schwabacher, "Mining distance-based outliers in near linear time with randomization and a simple pruning rule," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 29–38.

[35] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Information Sciences*, vol. 191, pp. 192–213, 2012.

[36] G. Bhattacharya, K. Ghosh, and A. S. Chowdhury, "Outlier detection using neighborhood rank difference," *Pattern Recognition Letters*, vol. 60, pp. 24–31, 2015.

[37] A. P. Boedihardjo, C.-T. Lu, and F. Chen, "Fast adaptive kernel density estimator for data streams," *Knowledge and Information Systems*, vol. 42, no. 2, pp. 285–317, 2015.

[38] B. Bolker, "R development core team. bbmle: Tools for general maximum likelihood estimation [internet]. 2012."

[39] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini, "Anomaly detection using autoencoders in high performance computing systems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 9428–9433.

[40] W. Bradley, D. Jarman, R. Plott, R. Wood, T. Schofield, R. Auflick, D. Cocking, *et al.*, "A task force approach to reducing stuck pipe costs," in *SPE/IADC Drilling Conference*. Society of Petroleum Engineers, 1991.

[41] T. M. Breuel, "Robust least-square-baseline finding using a branch and bound algorithm," in *Document Recognition and Retrieval IX*, vol. 4670. International Society for Optics and Photonics, 2001, pp. 20–27.

[42] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *ACM sigmod record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.

[43] R. Burduk, "New adaboost algorithm based on interval-valued fuzzy sets," in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2012, pp. 794–801.

[44] R. J. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 1, p. 5, 2015.

[45] G. O. Campos, A. Zimek, and W. Meira, "An unsupervised boosting strategy for outlier detection ensembles," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2018, pp. 564–576.

[46] F. Cao, M. Estert, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in *Proceedings of the 2006 SIAM international conference on data mining*. SIAM, 2006, pp. 328–339.

[47] K. Cao, L. Shi, G. Wang, D. Han, and M. Bai, "Density-based local outlier detection on uncertain data," in *International Conference on Web-Age Information Management*. Springer, 2014, pp. 67–71.

REFERENCES

[48] L. Cao, D. Yang, Q. Wang, Y. Yu, J. Wang, and E. A. Rundensteiner, "Scalable distance-based outlier detection over high-volume data streams," in *2014 IEEE 30th International Conference on Data Engineering*. IEEE, 2014, pp. 76–87.

[49] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[50] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly detection using one-class neural networks," *arXiv preprint arXiv:1802.06360*, 2018.

[51] A. Chamkalani, M. Pordel Shahri, S. Poordad, *et al.*, "Support vector machine model: a new methodology for stuck pipe prediction," in *SPE Unconventional Gas Conference and Exhibition*. Society of Petroleum Engineers, 2013.

[52] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[53] C. Chatar, M. D. Imler, *et al.*, "Overcoming a difficult salt drilling environment in the gulf of mexico: a case study," in *IADC/SPE Drilling Conference and Exhibition*. Society of Petroleum Engineers, 2010.

[54] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 2017, pp. 90–98.

[55] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 133–142.

[56] Y. Chen, X. S. Zhou, and T. S. Huang, "One-class svm for learning in image retrieval." in *ICIP (1)*. Citeseer, 2001, pp. 34–37.

[57] M. Chenaghlou, M. Moshtaghi, C. Leckie, and M. Salehi, "An efficient method for anomaly detection in non-stationary data streams," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[58] M. Chenaghlou, M. Moshtaghi, C. Leckie, and M. Salehi, "Online clustering for evolving data streams with online anomaly detection," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2018, pp. 508–521.

[59] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent & fuzzy systems*, vol. 2, no. 3, pp. 267–278, 1994.

[60] J. Courteille, M. Fabre, C. Hollander, *et al.*, "An advanced solution: The drilling adviser," *Journal of Petroleum Technology*, vol. 38, no. 08, pp. 899–904, 1986.

[61] P. I. Dalatu, A. Fitrianto, and A. Mustapha, "A comparative study of linear and nonlinear regression models for outlier detection," in *International Conference on Soft Computing and Data Mining*. Springer, 2016, pp. 316–326.

[62] T. T. Dang, H. Y. Ngan, and W. Liu, "Distance-based k-nearest neighbors outlier detection method in large-scale traffic data," in *2015 IEEE International Conference on Digital Signal Processing (DSP)*. IEEE, 2015, pp. 507–510.

[63] J. P. DeGeare, *The guide to oilwell fishing operations: tools, techniques, and rules of thumb*. Gulf Professional Publishing, 2014.

[64] H. Du *et al.*, "Robust local outlier detection," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE, 2015, pp. 116–123.

[65] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1285–1298.

[66] J. K. Dutta, B. Banerjee, and C. K. Reddy, "Rods: Rarity based outlier detection in a sparse coding framework," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 483–495, 2015.

[67] F. Y. Edgeworth, "Xli. on discordant observations," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 23, no. 143, pp. 364–375, 1887.

[68] M. Elahi, K. Li, W. Nisar, X. Lv, and H. Wang, "Efficient clustering-based outlier detection algorithm for dynamic data stream," in *2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 5. IEEE, 2008, pp. 298–304.

[69] S. Elkatatny, A. Abdulraheem, M. Mahmoud, A. Z. Ali, I. Mohamed, *et al.*, "Prediction of rate of penetration of deep and tight formation using support vector machine," in *SPE Kingdom of Saudi Arabia Annual Technical Symposium and Exhibition*. Society of Petroleum Engineers, 2018.

[70] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.

[71] B. Esmael, A. Arnaout, R. K. Fruhwirth, and G. Thonhauser, "Automated system for drilling operations classification using statistical features," in *Hybrid Intelligent Systems (HIS), 2011 11th International Conference on*. IEEE, 2011, pp. 196–199.

[72] B. Esmael, A. Arnaout, R. K. Fruhwirth, and G. Thonhauser, "Improving time series classification using hidden markov models," in *Hybrid Intelligent Systems (HIS), 2012 12th International Conference on*. IEEE, 2012, pp. 502–507.

[73] B. Esmael, A. Arnaout, R. K. Fruhwirth, and G. Thonhauser, "Multivariate time series classification by combining trend-based and value-based approximations," in *International Conference on Computational Science and Its Applications*. Springer, 2012, pp. 392–403.

[74] B. Esmael, A. Arnaout, R. K. Fruhwirth, and G. Thonhauser, "A statistical feature-based approach for operations recognition in drilling time series," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 5, pp. 454–461, 2015.

[75] B. Esmael, A. Arnaout, R. K. Fruhwirth, and G. Thonhauser, "A statistical feature-based approach for operations recognition in drilling time series," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 5, pp. 454–461, 2015.

[76] H. Fan, O. R. Zaïane, A. Foss, and J. Wu, "Resolution-based outlier factor: detecting the top-n most outlying data points in engineering data," *Knowledge and Information Systems*, vol. 19, no. 1, pp. 31–51, 2009.

[77] Y. Feng, K. Gray, *et al.*, "Modeling lost circulation through drilling-induced fractures," *Spe Journal*, vol. 23, no. 01, pp. 205–223, 2018.

[78] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.

[79] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine learning*, vol. 29, no. 2-3, pp. 131–163, 1997.

[80] A. A. Garrouch and H. M. Lababidi, "Development of an expert system for underbalanced drilling using fuzzy logic," *Journal of Petroleum Science and Engineering*, vol. 31, no. 1, pp. 23–39, 2001.

[81] A. A. Garrouch, H. Labbabidi, *et al.*, "Using fuzzy logic for ubd candidate selection," in *IADC/SPE Underbalanced Technology Conference and Exhibition*. Society of Petroleum Engineers, 2003.

[82] J. Gebhardt, M. Goldstein, F. Shafait, and A. Dengel, "Document authentication using printing technique features and unsupervised anomaly detection," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 479–483.

[83] G. B. Gebremeskel, C. Yi, Z. He, and D. Haile, "Combined data mining techniques based patient data outlier detection for healthcare safety," *International Journal of Intelligent Computing and Cybernetics*, vol. 9, no. 1, pp. 42–68, 2016.

[84] Z. Geng, H. Wang, M. Fan, Y. Lu, Z. Nie, Y. Ding, and M. Chen, "Predicting seismic-based risk of lost circulation using machine learning," *Journal of Petroleum Science and Engineering*, vol. 176, pp. 679–688, 2019.

[85] K. George, E.-H. Han, and V. Kumar, "Chameleon: a hierarchical clustering algorithm using dynamic modeling," *IEEE computer*, vol. 27, no. 3, pp. 329–341, 1999.

[86] Z. Ghahramani, "Learning dynamic bayesian networks," in *International School on Neural Networks, Initiated by IIASS and EMFCSC*. Springer, 1997, pp. 168–197.

[87] A. Ghoting, S. Parthasarathy, and M. E. Otey, "Fast mining of distance-based outliers in high-dimensional datasets," *Data Mining and Knowledge Discovery*, vol. 16, no. 3, pp. 349–364, 2008.

[88] T. Goebel, R. V. Molina, R. Vilalta, and K. D. Gupta, "Method and system for predicting a drill string stuck pipe event," June 17 2014, uS Patent 8,752,648.

[89] P. Gogoi, D. Bhattacharyya, B. Borah, and J. K. Kalita, "A survey of outlier detection methods in network anomaly identification," *The Computer Journal*, vol. 54, no. 4, pp. 570–588, 2011.

[90] M. Goldstein and A. Dengel, "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm," *KI-2012: Poster and Demo Track*, pp. 59–63, 2012.

[91] O. J. Gonzalez, H. Bernat, P. Moore, *et al.*, "The extraction of mud-stuck tubulars using vibratory resonant techniques," in *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 2007.

[92] S. D. Grigorescu, I. Potarniche, O. M. Ghita, and M. Covrig, "Computer added monitoring of drilling rig systems," in *2011 IEEE International Instrumentation and Measurement Technology Conference*. IEEE, 2011, pp. 1–5.

[93] O. E. Gundersen, R. Kucs, T. Sheehy, M. S. Vocal de Holden, *et al.*, "Transferring knowledge through decision support: A case study in drilling," in *IADC/SPE Drilling Conference and Exhibition*. Society of Petroleum Engineers, 2014.

[94] O. E. Gundersen and F. Sørmo, "An architecture for multi-dimensional temporal abstraction supporting decision making in oil-well drilling," in *Combinations of Intelligent Methods and Applications*. Springer, 2013, pp. 21–40.

[95] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250–2267, 2013.

[96] E. Gurina, N. Klyuchnikov, A. Zaytsev, E. Romanenkova, K. Antipova, I. Simon, V. Makarov, and D. Koroteev, "Application of machine learning to accidents detection at directional drilling," *Journal of Petroleum Science and Engineering*, p. 106519, 2019.

[97] J. Ha, S. Seok, and J.-S. Lee, "A precise ranking method for outlier detection," *Information Sciences*, vol. 324, pp. 88–107, 2015.

[98] A. S. Hadi, A. R. Imon, and M. Werner, "Detection of outliers," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 1, no. 1, pp. 57–70, 2009.

[99] G. Haduch *et al.*, "Solution of common stuck pipe problems through the adaptation of torque drag calculations," in *SPE/IADC Drilling Conference*. Society of Petroleum Engineers, 1994.

[100] W. Halliday, D. Clapper, *et al.*, "Toxicity and performance testing of non-oil spotting fluid for differentially stuck pipe," in *SPE/IADC Drilling Conference*. Society of Petroleum Engineers, 1989.

[101] W. Hempkins, R. Kingsborough, W. Lohec, C. Nini, *et al.*, "Multivariate statistical analysis of stuck drillpipe situations," *SPE Drilling Engineering*, vol. 2, no. 03, pp. 237–244, 1987.

[102] D. Hendrycks, M. Mazeika, and T. G. Dietterich, "Deep anomaly detection with outlier exposure," *arXiv preprint arXiv:1812.04606*, 2018.

[103] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori, "Statistical outlier detection using direct density ratio estimation," *Knowledge and information systems*, vol. 26, no. 2, pp. 309–336, 2011.

[104] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.

[105] J. Howard, S. Glover, *et al.*, "Tracking stuck pipe probability while drilling," in *SPE/IADC Drilling Conference*. Society of Petroleum Engineers, 1994.

[106] H. Huang, K. Mehrotra, and C. K. Mohan, "Rank-based outlier detection," *Journal of Statistical Computation and Simulation*, vol. 83, no. 3, pp. 518–531, 2013.

[107] J. Huang, Q. Zhu, L. Yang, and J. Feng, "A non-parameter outlier detection algorithm based on natural neighbor," *Knowledge-Based Systems*, vol. 92, pp. 71–77, 2016.

[108] B. Hughes, "Drilling engineering workbook," *Baker Huges INTEQ, Houston, TX*, 1995.

[109] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 387–395.

[110] T. Inoue, D. Sugiyama, and T. Shimotomai, "Machine learning approaches to anomaly detection of top drive torque causing drill pipe failure," in *ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering*. American Society of Mechanical Engineers, 2018, pp. V008T11A006–V008T11A006.

[111] R. Jahanbakhshi, R. Keshavarzi, M. Aliyari Shoorehdeli, A. Emamzadeh, *et al.*, "Intelligent prediction of differential pipe sticking by support vector machine compared with conventional artificial neural networks: An example of iranian offshore oil fields," *SPE Drilling & Completion*, vol. 27, no. 04, pp. 586–595, 2012.

[112] Z. X. W. Jie, "A predictive model for oil-drilling parameters based on the hierarchical fuzzy system [j]," *Acta Petrolei Sinica*, vol. 5, 2010.

[113] W. Jin, A. K. Tung, J. Han, and W. Wang, "Ranking outliers using symmetric neighborhood relationship," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2006, pp. 577–593.

[114] I. Kakanakova and S. Stoyanov, "Outlier detection via deep learning architecture," in *Proceedings of the 18th International Conference on Computer Systems and Technologies*. ACM, 2017, pp. 73–79.

[115] J. Kallmeyer, "Contamination control for scientific drilling operations," in *Advances in applied microbiology*. Elsevier, 2017, vol. 98, pp. 61–91.

[116] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009, vol. 344.

[117] K. Kavaklioglu, "Statistical processing methods used in abnormal situation detection," Mar. 19 2013, uS Patent 8,401,819.

[118] F. Keller, E. Muller, and K. Bohm, "Hics: High contrast subspaces for density-based outlier ranking," in *2012 IEEE 28th international conference on data engineering*. IEEE, 2012, pp. 1037–1048.

[119] B. Kiran, D. Thomas, and R. Parakkal, "An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos," *Journal of Imaging*, vol. 4, no. 2, p. 36, 2018.

[120] E. Kirner, E. Schubert, and A. Zimek, "Good and bad neighborhood approximations for outlier detection ensembles," in *International Conference on Similarity Search and Applications*. Springer, 2017, pp. 173–187.

[121] E. M. Knorr and R. T. Ng, "Algorithms for mining distance-based outliers in large datasets," in *VLDB*, vol. 98. Citeseer, 1998, pp. 392–403.

[122] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: algorithms and applications," *The VLDB Journal—The International Journal on Very Large Data Bases*, vol. 8, no. 3-4, pp. 237–253, 2000.

[123] M. Kontaki, A. Gounaris, A. N. Papadopoulos, K. Tsichlas, and Y. Manolopoulos, "Continuous monitoring of distance-based outliers over data streams," in *2011 IEEE 27th International Conference on Data Engineering*. IEEE, 2011, pp. 135–146.

[124] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Loop: local outlier probabilities," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 1649–1652.

[125] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Outlier detection in axis-parallel subspaces of high dimensional data," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2009, pp. 831–838.

[126] H.-P. Kriegel, P. Kroger, E. Schubert, and A. Zimek, "Interpreting and unifying outlier scores," in *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM, 2011, pp. 13–24.

[127] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, pp. 1–13, 2017.

[128] L. J. Latecki, A. Lazarevic, and D. Pokrajac, "Outlier detection with kernel density functions," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2007, pp. 61–75.

[129] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 157–166.

[130] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks," in *International Conference on Artificial Neural Networks*. Springer, 2019, pp. 703–716.

[131] Y. Li, W. Cao, and C. Gan, "A safety assessment method based on online sequential extreme learning machine (os-elm) in deep drilling process," in *2018 37th Chinese Control Conference (CCC)*. IEEE, 2018, pp. 10 228–10 232.

[132] A. Liaw, M. Wiener, *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[133] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[134] J. Liu and H. Deng, "Outlier detection on uncertain data based on local information," *Knowledge-Based Systems*, vol. 51, pp. 60–71, 2013.

[135] T. Love *et al.*, "Stickiness factor-a new way of looking at stuck pipe," in *IADC/SPE Drilling Conference*. Society of Petroleum Engineers, 1983.

[136] E. Lozano and E. Acufia, "Parallel algorithms for distance-based and density-based outliers," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*. IEEE, 2005, pp. 4–pp.

[137] Z. Ma, A. K. Vajargah, H. Lee, H. Darabi, D. Castineira, *et al.*, "Applications of machine learning and data mining in speedwise® drilling analytics: A case study," in *Abu Dhabi International Petroleum Exhibition & Conference*. Society of Petroleum Engineers, 2018.

[138] R. P. Macdonald, V. Krueger, V. Dubinsky, J. D. Macpherson, and D. Dashevskiy, "Method and apparatus for prediction control in drilling dynamics using neural networks," May 4 2004, uS Patent 6,732,052.

[139] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[140] A. Magana-Mora, S. Gharbi, A. Alshaikh, A. Al-Yami, *et al.*, "Accupipepred: A framework for the accurate and early detection of stuck pipe for real-time drilling operations," in *SPE Middle East Oil and Gas Show and Conference*. Society of Petroleum Engineers, 2019.

[141] L. F. Maimó, Á. L. P. Gómez, F. J. G. Clemente, M. G. Pérez, and G. M. Pérez, "A self-adaptive deep learning-based system for anomaly detection in 5g networks," *IEEE Access*, vol. 6, pp. 7700–7712, 2018.

[142] M. Martınez and L. E. Sucar, "Learning dynamic naive bayesian classifiers," in *Proceedings of the Twenty-first International Florida Artificial Intelligence Research Symposium Conference*, 2008, pp. 655–659.

[143] R. Momtaz, N. Mohssen, and M. A. Gowayyed, "Dwof: a robust density-based outlier detection approach," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2013, pp. 517–525.

[144] J. K. Montgomery, S. R. Keller, N. Krahel, M. V. Smith, *et al.*, "Improved method for use of chelation to free stuck pipe and enhance treatment of lost returns," in *SPE/IADC drilling conference*. Society of Petroleum Engineers, 2007.

[145] H. Moradi Koupaie, S. Ibrahim, and J. Hosseinkhani, "Outlier detection in stream data by clustering method," *International Journal of Advanced Computer Science and Information Technology (IJACSIT) Vol*, vol. 2, pp. 25–34, 2014.

[146] M. Moshtaghi, J. C. Bezdek, T. C. Havens, C. Leckie, S. Karunasekera, S. Rajasegarar, and M. Palaniswami, "Streaming analysis in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 14, no. 9, pp. 905–921, 2014.

[147] M. Moshtaghi, J. C. Bezdek, C. Leckie, S. Karunasekera, and M. Palaniswami, "Evolving fuzzy rules for anomaly detection in data streams," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 3, pp. 688–700, 2014.

[148] E. Muller, I. Assent, U. Steinhausen, and T. Seidl, "Outrank: ranking outliers in high dimensional data," in *2008 IEEE 24th international conference on data engineering workshop*. IEEE, 2008, pp. 600–603.

[149] E. Müller, M. Schiffer, and T. Seidl, "Statistical selection of relevant subspace projections for outlier ranking," in *2011 IEEE 27th international conference on data engineering*. IEEE, 2011, pp. 434–445.

[150] M. A. Muqeem, A. E. Weekse, A. A. Al-Hajji, *et al.*, "Stuck pipe best practices-a challenging approach to reducing stuck pipe costs," in *SPE Saudi Arabia Section Technical Symposium and Exhibition*. Society of Petroleum Engineers, 2012.

[151] R. Ng and J. Han, ":" efficient and effective clustering methods for spatial data mining", proc. 20th int. conf. on very large data bases, santiago, chile, morgan kaufmann publishers," 1994.

[152] J. Ning, L. Chen, and J. Chen, "Relative density-based outlier detection algorithm," in *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*. ACM, 2018, pp. 227–231.

[153] R. Nybø, "Efficient drilling problem detection: Early fault detection by the combination of physical models and artificial intelligence," 2009.

[154] J. Orban and M. Iakimov, "Method of determination of a stuck point in drill pipes by measuring the magnetic permeability of pipes," Oct. 9 2012, uS Patent 8,284,074.

[155] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "Loci: Fast outlier detection using the local correlation integral," in *Proceedings 19th International Conference on Data Engineering (Cat. No. 03CH37405)*. IEEE, 2003, pp. 315–326.

[156] C. M. Park and J. Jeon, "Regression-based outlier detection of sensor measurements using independent variable synthesis," in *International Conference on Data Science*. Springer, 2015, pp. 78–86.

[157] J. R. Pasillas-Díaz and S. Ratté, "Bagged subspaces for unsupervised outlier detection," *Computational Intelligence*, vol. 33, no. 3, pp. 507–523, 2017.

[158] E. L. Paula, M. Ladeira, R. N. Carvalho, and T. Marzagao, "Deep learning anomaly detection as support fraud investigation in brazilian exports and anti-money laundering," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2016, pp. 954–960.

[159] M. Pavlidou and G. Zioutas, "Kernel density outlier detector," in *Topics in Nonparametric Statistics*. Springer, 2014, pp. 241–250.

[160] D. Pokrajac, A. Lazarevic, and L. J. Latecki, "Incremental local outlier detection for data streams," in *2007 IEEE symposium on computational intelligence and data mining*. IEEE, 2007, pp. 504–515.

[161] U. Porwal and S. Mukund, "Credit card fraud detection in e-commerce: An outlier detection approach," *arXiv preprint arXiv:1811.02196*, 2018.

[162] S. Priyadarshy, "Visualization of quantitative drilling operations data related to a stuck pipe event," Feb. 15 2018, uS Patent App. 15/558,146.

[163] X. Qin, L. Cao, E. A. Rundensteiner, and S. Madden, "Scalable kernel density estimation-based local outlier detection over large data streams." in *EDBT*, 2019, pp. 421–432.

[164] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[165] L. R. Rabiner and B.-H. Juang, "An introduction to hidden markov models," *ieee assp magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[166] M. Radovanović, A. Nanopoulos, and M. Ivanović, "Reverse nearest neighbors in unsupervised distance-based outlier detection," *IEEE transactions on knowledge and data engineering*, vol. 27, no. 5, pp. 1369–1382, 2014.

[167] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *ACM Sigmod Record*, vol. 29, no. 2. ACM, 2000, pp. 427–438.

[168] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova, "Anomaly detection in dynamic networks: a survey," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7, no. 3, pp. 223–247, 2015.

[169] S. Rayana and L. Akoglu, "Less is more: Building selective anomaly ensembles," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 4, p. 42, 2016.

[170] S. Rayana, W. Zhong, and L. Akoglu, "Sequential ensemble learning for outlier detection: A bias-variance perspective," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 1167–1172.

[171] W. A. Reddie, E. R. Werlein, and H. A. Simon, "Freeing stuck pipe," Nov. 16 1965, uS Patent 3,217,802.

[172] D. Ren, I. Rahal, W. Perrizo, and K. Scott, "A vertical distance-based outlier detection method with local pruning," in *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. ACM, 2004, pp. 279–284.

[173] D. Ren, B. Wang, and W. Perrizo, "Rdf: A density-based outlier detection method using vertical data representation," in *Fourth IEEE International Conference on Data Mining (ICDM'04)*. IEEE, 2004, pp. 503–506.

[174] J. Ren and R. Ma, "Density-based data streams clustering over sliding windows," in *2009 Sixth international conference on fuzzy systems and knowledge discovery*, vol. 5. IEEE, 2009, pp. 248–252.

[175] H. Rizk, S. Elgokhy, and A. Sarhan, "A hybrid outlier detection algorithm based on partitioning clustering and density measures," in *2015 Tenth International Conference on Computer Engineering & Systems (ICCES)*. IEEE, 2015, pp. 175–181.

[176] H. Rostami and A. Khaksar Manshad, "A new support vector machine and artificial neural networks for prediction of stuck pipe in drilling of oil fields," *Journal of Energy Resources Technology*, vol. 136, no. 2, 2014.

[177] P. J. Rousseeuw and M. Hubert, "Robust statistics for outlier detection," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 73–79, 2011.

[178] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John wiley & sons, 2005, vol. 589.

[179] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International Conference on Machine Learning*, 2018, pp. 4393–4402.

[180] B. N. Saha, N. Ray, and H. Zhang, "Snake validation: A pca-based outlier detection method," *IEEE signal processing letters*, vol. 16, no. 6, pp. 549–552, 2009.

[181] M. Salehi, C. A. Leckie, M. Moshtaghi, and T. Vaithianathan, "A relevance weighted ensemble model for anomaly detection in switching data streams," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2014, pp. 461–473.

[182] K. Salminen, C. Cheatham, M. Smith, K. Valiullin, *et al.*, "Stuck-pipe prediction by use of automated real-time modeling and data analysis," *SPE Drilling & Completion*, vol. 32, no. 03, pp. 184–193, 2017.

[183] K. P. Salminen, C. A. Cheatham, and M. A. Smith, "Real-time stuck pipe warning system for downhole operations," Dec. 22 2016, uS Patent App. 15/008,161.

[184] V. K. Samparthi and H. K. Verma, "Outlier detection of data in wireless sensor networks using kernel density estimation," *International Journal of Computer Applications*, vol. 5, no. 7, pp. 28–32, 2010.

[185] H. Santos *et al.*, "Differentially stuck pipe: early diagnostic and solution," in *IADC/SPE Drilling Conference*. Society of Petroleum Engineers, 2000.

[186] M. H. Satman, "A new algorithm for detecting outliers in linear regression," *International Journal of Statistics and Probability*, vol. 2, no. 3, p. 101, 2013.

[187] E. Schubert, R. Wojdanowski, A. Zimek, and H.-P. Kriegel, "On evaluation of outlier rankings and outlier scores," in *Proceedings of the 2012 SIAM International Conference on Data Mining*. SIAM, 2012, pp. 1047–1058.

[188] E. Schubert, A. Zimek, and H.-P. Kriegel, "Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection," *Data Mining and Knowledge Discovery*, vol. 28, no. 1, pp. 190–237, 2014.

[189] A. B. Serapião and J. R. P. Mendes, "Classification of petroleum well drilling operations with a hybrid particle swarm/ant colony algorithm," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 2009, pp. 301–310.

[190] A. B. Serapião, J. R. Mendes, and K. Miura, "Artificial immune systems for classification of petroleum well drilling operations," in *Artificial Immune Systems*. Springer, 2007, pp. 47–58.

[191] A. B. Serapiao, R. M. Tavares, J. R. P. Mendes, and I. R. Guilherme, "Classification of petroleum well drilling operations using support vector machine (svm)," in *Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*. IEEE, 2006, pp. 145–145.

[192] R. Shivers, R. Domangue, *et al.*, "Operational decision making for stuck pipe incidents in the gulf of mexico: A risk economics approach," *SPE Drilling & Completion*, vol. 8, no. 02, pp. 125–130, 1993.

[193] M. Shukla, Y. Kosta, and P. Chauhan, "Analysis and evaluation of outlier detection algorithms in data streams," in *2015 International Conference on Computer, Communication and Control (IC4)*. IEEE, 2015, pp. 1–8.

[194] P. Skalle, J. Sveen, and A. Aamodt, "Improved efficiency of oil well drilling through case based reasoning," in *Pacific Rim International Conference on Artificial Intelligence*. Springer, 2000, pp. 712–722.

[195] P. Skalle, A. Aamodt, O. E. Gundersen, *et al.*, "Detection of symptoms for revealing causes leading to drilling failures," *SPE Drilling & Completion*, vol. 28, no. 02, pp. 182–193, 2013.

[196] G. Smrithy, S. Munirathinam, and R. Balakrishnan, "Online anomaly detection using non-parametric technique for big data streams in cloud collaborative environment," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 1950–1955.

[197] S. G. Stroud, "System for determining the free point of pipe stuck in a borehole," Nov. 24 1987, uS Patent 4,708,204.

[198] S. Su, L. Xiao, L. Ruan, F. Gu, S. Li, Z. Wang, and R. Xu, "An efficient density-based local outlier detection approach for scattered data," *IEEE Access*, vol. 7, pp. 1006–1020, 2018.

[199] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE transactions on systems, man, and cybernetics*, no. 1, pp. 116–132, 1985.

[200] J. Tamboli and M. Shukla, "A survey of outlier detection algorithms for data streams," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 2016, pp. 3535–3540.

[201] B. Tang and H. He, "A local density-based approach for outlier detection," *Neurocomputing*, vol. 241, pp. 171–180, 2017.

[202] J. Tang and Z. Chen, "chee fu, aw, and w. cheung, d. 2002. enhancing effectiveness of outlier detections for low density patterns," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 535–548.

[203] X. Tang, R. Yuan, and J. Chen, "Outlier detection in energy disaggregation using subspace learning and gaussian mixture model," *Int. J. Control Autom*, vol. 8, pp. 161–170, 2015.

[204] R. M. Tavares, J. R. P. Mendes, C. K. Morooka, and J. C. R. Plácido, "Automated classification system for petroleum well drilling using mud-logging data," in *Proc. of 18th International Congress of Mechanical Engineer, Offshore & Petroleum and Engineering, Ouro Preto, Brazil*, 2005.

[205] D. Toshniwal *et al.*, "A framework for outlier detection in evolving data streams by weighting attributes in clustering," *Procedia Technology*, vol. 6, pp. 214–222, 2012.

[206] M. S. Uddin, A. Kuh, Y. Weng, and M. Ilić, "Online bad data detection using kernel density estimation," in *2015 IEEE Power & Energy Society General Meeting*. IEEE, 2015, pp. 1–5.

[207] I. Å. Valås, "A data-intensive approach to prediction of unwanted events during oil and gas well drilling," Master's thesis, Institutt for datateknikk og informasjonsvitenskap, 2005.

[208] B. van Stein, M. van Leeuwen, and T. Bäck, "Local subspace-based outlier detection using global neighbourhoods," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 1136–1142.

[209] F. I. Vázquez, T. Zseby, and A. Zimek, "Outlier detection based on low density models," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2018, pp. 970–979.

[210] N. H. Vu and V. Gopalkrishnan, "Efficient pruning schemes for distance-based outlier detection," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2009, pp. 160–175.

[211] S. Walfish, "A review of statistical outlier methods," *Pharmaceutical technology*, vol. 30, no. 11, p. 82, 2006.

[212] C. Wang, H. Gao, Z. Liu, and Y. Fu, "A new outlier detection model using random walk on local information graph," *IEEE Access*, vol. 6, pp. 75 531–75 544, 2018.

[213] C. Wang, H. Gao, Z. Liu, and Y. Fu, "Outlier detection using diverse neighborhood graphs," in *2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. IEEE, 2018, pp. 58–62.

[214] X. Wang, X. L. Wang, Y. Ma, and D. M. Wilkes, "A fast mst-inspired knn-based outlier detection method," *Information Systems*, vol. 48, pp. 89–112, 2015.

[215] A. Willersrud, M. Blanke, L. Imsland, and A. Pavlov, "Drillstring washout diagnosis using friction estimation and statistical change detection," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 5, pp. 1886–1900, 2015.

[216] K. Wu, K. Zhang, W. Fan, A. Edwards, and S. Y. Philip, "Rs-forest: A rapid density estimator for streaming anomaly detection," in *2014 IEEE International Conference on Data Mining*. IEEE, 2014, pp. 600–609.

[217] W. Yan, Y. Peng, and H. Wu, "The research and development of automatic vertical drilling tool," in *Mechatronics and Automation (ICMA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1226–1231.

[218] D. Yang, E. A. Rundensteiner, and M. O. Ward, "Neighbor-based pattern detection for windows over streaming data," in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. ACM, 2009, pp. 529–540.

[219] X. Yang, L. J. Latecki, and D. Pokrajac, "Outlier detection with globally optimal exemplar-based gmm," in *Proceedings of the 2009 SIAM International Conference on Data Mining*. SIAM, 2009, pp. 145–154.

[220] G. Yarim, R. J. Uchytil, R. B. May, A. Trejo, *et al.*, "Stuck pipe prevention–a proactive solution to an old problem," in *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 2007.

[221] H. Yavari, M. Sabah, R. Khosravanian, D. Wood, *et al.*, "Application of an adaptive neuro-fuzzy inference system and mathematical rate of penetration models to predicting drilling rate," *Iranian Journal of Oil & Gas Science and Technology*, vol. 7, no. 3, pp. 73–100, 2018.

[222] J. Yin and J. Wang, "A model-based approach for text clustering with outlier detection," in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 2016, pp. 625–636.

[223] L. A. Zadeh, "Fuzzy sets, information and control," *vol*, vol. 8, pp. 338–353, 1965.

[224] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Transactions on computers*, vol. 100, no. 1, pp. 68–86, 1971.

[225] F. Zhang, S. Miska, M. Yu, E. Ozbayoglu, N. Takach, *et al.*, "A fast graphic approach to estimate hole cleaning for directional drilling," *SPE Drilling & Completion*, vol. 32, no. 01, pp. 51–58, 2017.

[226] J. Zhang, "Advancements of outlier detection: A survey," *ICST Transactions on Scalable Information Systems*, vol. 13, no. 1, pp. 1–26, 2013.

[227] J. Zhang, Y. Jiang, K. H. Chang, S. Zhang, J. Cai, and L. Hu, "A concept lattice based outlier mining method in low-dimensional subspaces," *Pattern Recognition Letters*, vol. 30, no. 15, pp. 1434–1439, 2009.

[228] K. Zhang, M. Hutter, and H. Jin, "A new local distance-based outlier detection approach for scattered real-world data," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2009, pp. 813–822.

[229] L. Zhang, J. Lin, and R. Karim, "Adaptive kernel density-based anomaly detection for nonlinear systems," *Knowledge-Based Systems*, vol. 139, pp. 50–63, 2018.

[230] J. Zhao, Y. Shen, W. Chen, Z. Zhang, S. Johnston, *et al.*, "Machine learning–based trigger detection of drilling events based on drilling data," in *SPE Eastern Regional Meeting*. Society of Petroleum Engineers, 2017.

[231] Y. Zhao and M. K. Hryniewicki, "Xgbod: improving supervised outlier detection with unsupervised representation learning," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[232] Y. Zhao and M. K. Hryniewicki, "Dcso: dynamic combination of detector scores for outlier ensembles," *arXiv preprint arXiv:1911.10418*, 2019.

[233] Y. Zhao, Z. Nasrullah, M. K. Hryniewicki, and Z. Li, "Lscp: Locally selective combination in parallel outlier ensembles," in *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 2019, pp. 585–593.

[234] Z. Zheng, H.-Y. Jeong, T. Huang, and J. Shu, "Kde based outlier detection on distributed data streams in multimedia network," *Multimedia Tools and Applications*, vol. 76, no. 17, pp. 18 027–18 045, 2017.

[235] P.-Y. Zhou and K. C. Chan, "Fuzzy feature extraction for multichannel eeg classification," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 2, pp. 267–279, 2018.

[236] A. Zimek, M. Gaudet, R. J. Campello, and J. Sander, "Subsampling for efficient and effective unsupervised outlier detection ensembles," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 428–436.

[237] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, no. 5, pp. 363–387, 2012.

[238] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," 2018.