Republic of Iraq

Ministry of Higher Education

And Scientific Research

University of Diyala

College of Science

Prediction Of Reservoirs Porosity Based On Resulting Seismic Data Attributes Using Deep Learning Approach

A Thesis

Submitted to the Department of Computer Science\ College of Science\ University of Diyala

In a Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

By

Mohammed Wahab Raheem

Supervised By

Assist.Prof. Dr. Abdulbasit Kadhim Shukur

2022 A.D. 1443 A.H.

بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

﴿ يَرۡفَعِ ٱللَّهُ ٱلَّذِينَ ءَامَنُواْ مِنكُمۡ وَٱلَّذِينَ أُوتُواْ ٱلۡعِلۡمَ دَرَجَٰتٖۚ وَٱللَّهُ بِمَا تَعۡمَلُونَ خَبِيرٞ ١١ ﴾

صدق الله العظيم

سورة المجادلة – الآية (١١)

DEDICATION

To...

- *The souls of my father and mother, my God have mercy on them...*
- *My beloved wife ...*
- *My precious children Abdullah, Gana, and Humam*
- *My beloved brothers and sisters...*
- *My dear friend Dr. Hassan*
- *My friends and classmates... and everyone who want goodness and success for me.*

Mohammed

2022

Linguistic Certification

*This is to certify that the thesis entitled "**Prediction of Reservoirs Porosity Based on Resulting Seismic Data Attributes Using Deep Learning Approach**" was prepared by "**Mohammed W'ahab Raheem**" was prepared under my linguistic supervision. It was amended to meet the style of English language.*

Signature:

Name: **Dr.** Ghazwan Mohammed Jaafar

Date: / /2022

Scientific Amendment

I certify that the thesis entitled ***"Prediction of Reservoirs Porosity Based on Resulting Seismic Data Attributes Using Deep Learning Approach"*** *was prepared by* ***Mohammed W'ahab Raheem*** *has been evaluated scientifically ,therefore, it is suitable for debate by the examining committee.*

Signature:

Name: **Assist. Prof. Dr.** Abbas Abdulazeez Abdulhameed

Date: / / 2022

Signature:

Name: Assist. Prof. Dr. Ruaa Adeeb Abdulmunem

Date: / / 2022

Supervisor Certification

I certify that this research entitled ***"Prediction of Reservoirs Porosity Based on Resulting Seismic Data Attributes Using Deep Learning Approach"*** *was prepared under my supervision at Department of Computer Science\ College of Sciences\ the University of Diyala by* ***"Mohmmed W'ahab Raheem"*** *as a partial fulfillment of the requirements for the degree of* ***Master of Science in Computer Science***

(Supervisor)

Signature:

Name**: Assist. Prof. Dr. Abdulbasit Kadhim Shukur**

Date: / /2022

Approved by Department of Computer Science\ College of Science\ University of Diyala

Signature:

Name: **Assist. Prof. Dr.Bashar Talib Hamed**

Date: / /2022

Head of Department of Computer Science\ College of Science\ University of Diyala

Examination Committee Certification

We certify that we have read the thesis entitled ***"Prediction of Reservoirs Porosity Based on Resulting Seismic Data Attributes Using Deep Learning Approach"*** *and as an examination committee, examined the student* ***"Mohammed W'ahab Raheem"*** *in the thesis content and that in our opinion, it is adequate as fulfill the requirement for the Degree of Master in Computer Science at the Computer Science Department, University of Diyala.*

(Chairman)

Signature:

Name: **Prof. Dr.Taha Mohammed Hasan**

Date: / / 2022

(member)

Signature:

Name: **Assist. Prof. Dr. Maki Mahdi Abdulhasan**

Date: / / 2022

(member)

Signature:

Name: **Assist. Prof. Ali Abdulrahman Mahmood**

Date: / / 2022

(member) and (Supervisor)

Signature:

Name: **Assist. Prof. Dr. Abdulbasit Kadhim Shukur**

Date: / / 2022

Approved by the Dean of College of Science, University of Diyala

(The Dean)

Signature:

Name: **Prof. Dr. Tahseen Hussein Mubarak**

Date: **/ / 2022**

ABSTRACT

The massive development of shale oil formations has changed the rules of the game. On the other hand, Machine Learning (ML) and Deep Learning (DL) play an important role in the rapid development of all industries by automating most of the routine processes. The oil industry also gets equal benefits from ML and DL for reservoir development planning and operational accuracy through a series of automated systems. To develop the field, computational static and dynamic simulation models are generated based on various petrophysical properties collected through various resources that are time-consuming and expensive. This study aims to present a comprehensive model in the field of application of ML and DL to model the petrophysical properties using different methods and algorithms. Finally, the multiple ML and DL techniques that are tested in this study are discussed in detail in order to achieve more accuracy in the petrophysical simulation models. Machine learning models were used to support vector regression (SVR) and nearest neighbor regression (KNR), for further improvement, using deep learning algorithms. Use long-term memory (LSTM) and prepare the output by an artificial neural network (ANN). Also, to improve deep learning by recurrent neural networks (RNN) a hybrid method (LSTM) with a recurrent gates unit (GRU) and an artificial neural network (ANN) is used. The best decisions obtained in forecasting oil reservoirs and reducing uncertainty in exploration and drilling is if the data set is divided as follows, the prediction model using machine learning is 90% training and 10% testing. The best results were MAE = 0.238 and RMSE = 0.255 with SVR, while the KNR algorithms achieved results of MAE = 0.276 and RMSE = 0.301. While in deep learning algorithms when splitting the data into 80% training and 20% testing. The best performing result in LSTM had values of MAE = 0.023 and RMSE = 0.029, meaning the best performance for deep learning.

LIST OF CONTENTS

List of Figures

List of Tables

List of Abbreviations

Abbreviation	*DESCRIPTION*
ACE	*Alternative Conditional Expectation*
AI	*Artificial Intelligence*
ANN	*Artificial Neural Network*
CC	*Correlation Coefficient*
CM	*Committee Machine*
DLNN	*Deep Learning Neural Network*
DT	*Decision Tree*
FN	*Functional Network*
GRU	*Gated Recurrent Units*
IDDM	*Insulin-Dependent Diabetes Mellitus*
KNN	*K-Nearest Neighbor*
LR	*Logistic Regression*
LSTM	*Long-Short Term Memory*
MAA	*Multi-Attribute Analysis*
MAE	*Mean Absolute Error*
MDR	*Multifactor Dimensionality Reduction*
ML	*Machine Learning*
NN	*Neural Network*
OFL	*Optimized Fuzzy Logic*
ONN	*Optimized Neural Network*
PNN	*Probabilistic Neural Network*

RMAE	*Root Mean Absolute Error*
SAA	*Single Attribute Analysis*
SVM	*Support Vector Machine*
SVR	*Support Vector Regression*
TS	*Time Series*
TSA	*Time Series Analysis*

Chapter One

Introduction

Chapter One

Introduction

1.1 Overview

Oil is one of the most important natural resources on which developed countries depend on developing their economies. Its exploration and production stages are among the most important priorities of many countries. Oil prospecting and exploration are among the priorities of the mission that require the use of advanced techniques and methods. These methods and techniques differ from place to place according to the nature of the land and the geophysical formations of the fluid reservoirs. Exploration of oil wells involves obtaining information called the well log, which is a set of basic information on wells [1].

The different nature of the Earth's geophysical data is a problem facing drilling and exploration, which is unreliability and the increased risk of drilling. Therefore, Which is why seismic survey technology was discovered, which produced data called seismic data, It is a set of features by which the porosity and permeability of the well and some of the features of the reservoir formations can be known. With the help of machine learning models, which contributed to reducing drilling risks, increasing reliability, and predicting the porosity and permeability of the well [2].

Porosity is an important factor to determine the capacity of reservoirs of liquids and to give an understanding of the liquid and gaseous formations in them. Therefore, the basic standard requirements in tanks are porosity and permeability [3].

Data acquisition and analysis is very time-consuming and expensive, requires significant human and technical efforts, and the reservoir may not be adequately described. As a result, a less expensive and faster method for porosity quantification is required. Porosity can be estimated using the well log and seismic data, but many of these logs are difficult to obtain accurately [4].

An oil and gas reservoir is a rock formation in which petroleum and natural gas have accumulated. The oil and gas inside the reservoir are held by adjacent and accumulating layers of rock. Using available field and laboratory data, ML can describe different reservoir properties. The process of developing a reservoir, usually between the discovery and management phases of a reservoir, incorporates certain characteristics related to its ability to store and produce petroleum [5].

In recent years, deep learning techniques have been developed to process different types of data. One type of deep learning is a recurrent neural network (RNN), which is used for sequential or time-series data, such as text, audio, and video [6]. One associated technique is Long-Short-Term Memory (LSTM) which has processed time-series for a variety of data, and almost all of the excellent results have been achieved through deep learning [7].

This thesis presents a porosity prediction model using machine learning algorithms based on (SVR) Support Vector Regression and K Neighbors Regression (KNR), LSTM-based deep learning algorithms, and Gated Recurring Units (GRU). In addition, a hybrid algorithm was proposed using LSTM and GRU, and (ANN) was used with deep learning algorithms to adjust the output weights.

1.2 Related Works

The following are some recent studies on the relationship between machine learning and porosity prediction:

❖ **S.R. Na'imi. et al. (2014)** [8]**.** In this study, an SVR approach is represented by ML as a functional regression method in regression problems. Use the principle of structural risk reduction. Where appropriate seismic characteristics are extracted, which mainly depend on the porosity of the tank and the water saturation. Then, a quantitative formula for the relationship between porosity parameters. It is obtained by using a nonlinear vector regression algorithm in water saturation and selected seismic features. In the proposed SVR model, the results showed that it is suitable for implementation to predict porosity in small data and solve complex problems, compared to other methods that require more challenges.

❖ **Amin Gholami. et al (2017)** [9]. In this study. A mixed model is proposed to determine the articulation between porosity and seismic features by machine learning in three steps. In addition, the appropriate seismic features that have a prominent effect on porosity are extracted using the reorientation variable method and used as model input parameters. In addition, when compared to the non-parametric method known as alternative conditional expectation (ACE), the input variables are shifted to larger data space. In the next step, the correlation between the input parameters and porosity is quantitatively transformed through the optimized intelligence model, including optimized neural network (ONN), optimized support vector regression (OSVR), and optimized fuzzy logic (OFL) to achieve the predictive validity. In the final step, through the Committee Machine (CM), the integrated outputs of the optimized models to improve prediction accuracy are embedded in the

modeling intelligence. The Committee Machine (CM) model error distribution is very close to the normal distribution. The CM predictions are very compatible with reality because the errors (0.0068) from samplings show the range degree as to be in ± (0.0067) and (0.0301).

❖ **S. P. Maurya, et al (2018)[10].** The study goal is to discover an effective mix of seismic reflection techniques and geostatistical approaches for predicting porosity and identifying potential areas in 3D seismic data spaces. In this study, three geostatistical methods were used to predict porosity: single-attribute analysis (SAA), multi-attribute analysis (MAA), and the probabilistic neural network (PNN) algorithm. In a time interval of 1060-1075 ms, the result obtains a very high porosity (N 15%). These techniques make use of the seismic features generated by model-based reflection and color reflection techniques. The results demonstrated that all three statistical methods used to predict porosity are effective and reliable, but multi-feature and probabilistic neural network analysis provides more accurate and high-resolution porosity sections.

❖ **Xu Zhou, et al (2019)** [11]**.** This paper shows how to use big data analysis to verify the statistical correlations between seismic attributes parameters from three-dimensional seismic surveys and petro-physical properties from (well logs). Using Deep Learning Neural Network (DLNN) approach. The system used in this study consists of four different states with different types of seismic properties designed. To analyze the effect of each seismic property on approach execution. In addition, predict the porosity estimation of each case special features apply cases with the features applied. The cases approach has higher accuracy in predicting the porosity estimation, and the prediction accuracy may change due to the added features to increase seismic quality.

❖ **Anifowose, et al (2019**) [3]. This study used four types of (ML), which are Artificial Neural Network (ANN), Functional Network (FN), (SVM), and Decision Tree (DT). Demonstrate the effectiveness of these techniques in handling large amounts of seismic data. , which aims to estimate the porosity and predict the permeability of the reservoir. Therefore, from the point of view of the study, comparing the results with implementation criteria such as correlation coefficient (CC), root mean absolute error (RMAE) and mean absolute error (MAE) gives better results, it was discovered that SVM, when applied to seismic data, has high accuracy and depth matching. This leads to a significant difference in the results compared to other technologies, it positively affects the efficiency and quality of exploration and production. The study also showed that ANN has more smoothing power than FN with SVM performance. No heterogeneity was found with FN and DT. Porosity estimation and prediction of reservoir permeability were not very effective because five or more traits were used.

❖ **Qitao Zhang, et al (2019) [12]**. This study presents a method for predicting the spatial distribution of reservoir saturation using machine learning. This study used (LSTM) to predict the water saturation distribution. In addition, using data from actual and simulated monitoring of reservoirs. To get a better prediction of water saturation in rocks, the study compared RNN and (GRU), which are popular machine learning algorithms, with LSTM. The results showed that the LSTM method improved other machine learning methods and the fluid crowding prediction pattern. This study presented an alternative method to predict the water saturation distribution in reservoirs quickly and reliably. The LSTM can deal with questions location prediction problems.

❖ **Wei Liu, et al (2020) [13].** This study uses a numerical simulation method to predict oil production. Three prediction values have been proposed using the empirical ensemble decomposition method EEMD in LSTM, ANN, and SVM. The oil production chain in Chinese oil fields was selected as an experimental study. In base petroleum production, the data set must first be divided into training and testing. Then, the test set data is gradually added to the training set and analyzed by (EEMD) to obtain multiple intrinsic mode functions (IMFs). Then an appropriate number of constants (IMFs) are chosen as predictive variables for machine learning. In two real oil fields, the proposed evaluation and verification model was applied to the three values. The experimental results show that the proposed method can provide near-perfect predictions using LSTM over other algorithms.

❖**A. Ogbamikhumi et al (2021)**[14]. In this study conducted to predict reservoir properties, seismic reflection was combined with an artificial neural network (ANN), to predict fluid saturation and improve porosity. Using neural network techniques (NN) and multilayer feed neural networks (MLFN) and probabilistic neural networks (PNNs) computed from target characteristics where reservoir properties performance for porosity prediction predicted from seismic reflection. The expected attributes of the seismic data are related to the characteristics of the reservoir to test the accuracy of the process. The results gave good correlations for MLFN and PNN per well with a mean (CC) of 0.69 and 0.96, respectively, which indicates the evolution of PNN over MLFN.

1.3 Related Work Analyzing

Through the analysis of related works that there are similarities and differences between the previous studies and the current study as follows:

1. The studies are similar to the current study in terms of the following sides:

    - Dealing with porosity and permeability and subjecting them to experiments using a machine learning approach.
    - The type of data usage is the same in the current study, which are seismic data as well as similar in that they are a reliable source in predicting reservoir porosity.

    - The previous studies dealt with many of the experiences that the researcher benefited from in our current study.

2. The studies differ from the current study in terms of the following sides:

    - Selection of experimental characteristics. Where the studies used the design of experimental characteristics according to the well log, while the researcher in the current study used the characteristics of seismic data according to the sequence of the time signal.
    - Dealing with the geological diversity of the Earth and subjecting it to experiments through the proposed model, which was achieved by the proposed system in the current thesis.

1.4 Problem Statement

The main problem facing oil exploration is how to add big data for prediction, exploration, and production. These studies used ML and DL techniques to determine drilling accuracy and reliability, reduce uncertainty and reduce costs, and this is called the use of smart systems and machine learning algorithms in research development, drilling, and production. This thesis will discuss two issues.

✓ Data is the number one problem: Iraq still has flaws in complex calculations. Description of seismic survey data.

✓ Porosity is the second issue: the porosity of the oil tank is very important. Porosity estimation should be very good in tanks and oil tanks should have high reliability before drilling.

1.5 Aim of Thesis

The main objective of this thesis is to design and implement an efficient and effective approach to reservoir porosity prediction based on temporal sequential data processing, ML, and DL techniques to achieve a high degree of accuracy, as well as to compare these techniques to determine the best among them.

1.6 Contribution

The main contribution of this thesis is the application of the oil reservoir porosity prediction system. However, the new contribution to this thesis uses an intelligent system based on seismic data. Another contribution to this thesis is the use of seismic data from a well under exploration and drilling.

1.7 Outline of the Thesis

In this work, the Thesis **" *Prediction Of Reservoirs Porosity Based On Resulting Seismic Data Attributes Using Deep Learning Approach* "** is structured in five chapters; here is a brief description of their contents is given:

Chapter 2: This chapter provides theoretical backgrounds and an overview of reservoir engineering and seismic data. ML and DL are explained with their respective sections. In addition, how the proposed systems can be used through the ML and DL approaches, with an explanation of all the algorithms used in the proposed approach with all the examples and detailed equations.

Chapter 3: This chapter details the proposed approach introduces the proposed main system and design objectives and covers seismic data features that predict porosity.

Chapter 4: This chapter gives presents the results and tests of the proposed system. and experimental results obtained from the implementation of the proposed system.

Chapter 5: This chapter includes conclusions and future work for the development of use seismic data attributes to predict porosity approaches with lists several suggestions for future studies.

Chapter Two

Theoretical Background

Chapter Two

Theoretical Background

2.1 Introduction

Oil is one of the most important areas of life in most countries of the world, and the economies of many countries are completely dependent on oil. Therefore, oil exploration and drilling should be of a high level of importance. In addition, attention should be paid to the development of new methods and technologies for its extraction and production [15].

This chapter describes the theoretical aspects of seismic data analysis and data preprocessing, data science, correlations, and techniques used to predict porosity. Through machine learning, as supports vector regression (SVR) algorithms, and k-nearest neighbor regression (KNR), as well as using Deep learning Recurrent Neural Network (RNN) and its algorithms, such as LSTM, GRU, and accuracy criteria.

2.2 Basic Operations of Seismic Exploration

Seismic exploration is the use of seismic energy to examine the Earth's interior. To aid in the search for fluids such as petroleum, or other minerals, the exploration seismic method is interested at or depth the Earth's surface to estimate subsurface flexible properties and notice differences in properties data. Rock accumulation differences refer to elastic properties under the surface of the earth usually refer to changes in the porosity of fluids. A seismic survey is a method of obtaining a graphical description of the underground structure by analyzing the reflected seismic waves [16].

2.2.1 Seismic Data Acquisition

The purpose of exploring oil and other fluids is carried out through a wide range of sensors scattered in thousands of geology. The Central Console and Operations that are sent data on the ground are to locate oil and gas springs [17].

These reversible waves will be the same frequency, such as waves sent by the wave source, which is usually within 10-100 Hz, long-enough wavelengths allow deep in sub-layers [18].

2.2.2 Seismic Data Processing

Seismic data processing is a string of mathematical operations that bring out noise attendant seismic data as well as make geometrical improvements such that the final seismic image of the subsurface will show a map to underground [19].

2.3 Data Science

Data science consists of topics covering general terms such as (AI),(ML) and (DL), and time series modeling. In this section, some of the primary research use cases that data scientists have benefited from our data science and analysis [20]. As shows in Figure (2.1).

Data science analysis principles and techniques are widespread and numerous, and perhaps one of them that increases the accuracy of the possibilities is the integration of past data with current data to reveal links in the data. In addition, the interpretation of results about some prior knowledge (theoretical) of properties and phenomena based on these data was analyzed using smart technologies [21].

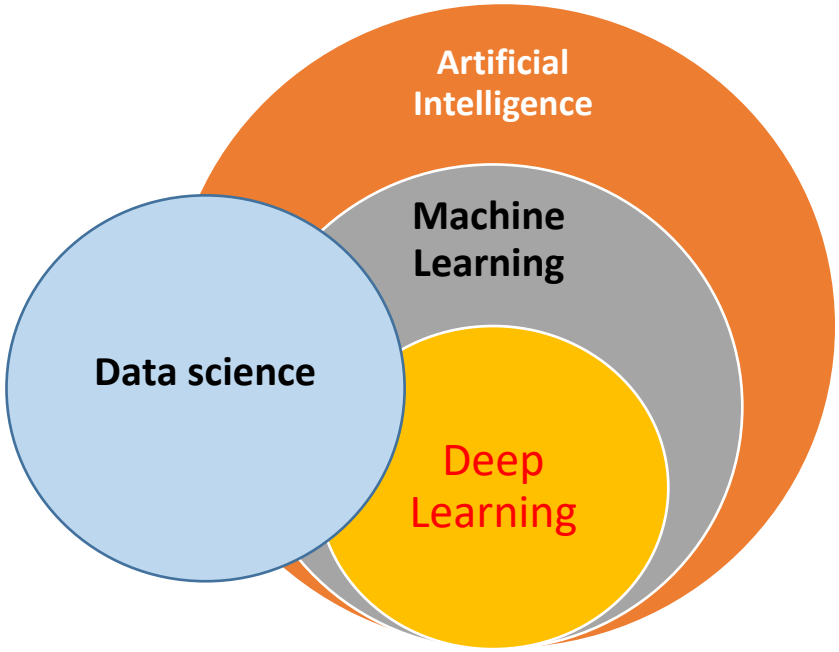


Figure (2.1): Relation Between (AI, ML, DL) and Data Science

2.4 Principal Processing in a Dataset

Data processing technology is necessary. Preprocessing is the concept of changing raw (raw) data into a clean dataset. The dataset needs to check for loss values, distortions, and other inconsistencies before they are implemented in the algorithm. To be suitable for use in a model [21].

Also, data preprocessing is an iterative method for converting raw data into understandable and inferable patterns. Datasets (raw) are more common and are characterized by incompleteness, inconsistency, lack of behavior and trends, as well as errors. So to deal with missing values and handle inconsistencies, preprocessing is required [22].

2.5 Correlation Matrix

The correlation coefficient (CC) is a mensuration that represents the extent of the statistical relationship between two variables of an interval

or ratio level. The correlation coefficient is measured so that it is always between(0 and 1) [28].

When the value of (CC) is close to zero, it means that there is little correlation between the properties, and whenever the value of the distance is greater than 0, it means that the correlation is large and therefore we can cancel out some of the features. The case can be applied, for values in both directions, the positive or negative direction, with increasing and decreasing the relationship between properties. The two variables are often given the symbols X and Y, or the name of the attribute in the form of matrix 2D [29].

Pearson's correlation, the most commonly used correlation for scalar variables, shows how properties are related. Assigns a value between 0 and 1, where 0 has no correlation, and 1 is positive. The correlation value indicates 0.9 as used in the model and matrix (CC) [30].

- A value of 1 indicates a positive linear correlation.
- A value of 0 indicates that there is no linear correlation.

$$CC=r = \frac{\sum(Xi-\bar{X})(Yi-\bar{Y})}{\sqrt{\sum(Xi-\bar{Y})^2((Yi-\bar{Y})^2}} \quad (2.1)$$

CC = r

X_i = x-Variable in a Sample

$\bar{X}$= Mean of the x-variable

Y_i = The Y-Variable in a Sample

$\bar{Y}$= Mean of the Y-Variable

2.6 Time Series (TS)

Time series of data are in the form of a set of repeated variables with the same or several different values, either by a fixed amount or variable depending on the data type. Time series can be written in the form [31]:

$$\{X_1, X_2...X_n\} \text{ , n= number of sample}$$

$$\{x_{t-1}, x_t, x_{t+1}\}, \text{ t = time}$$

Time series analysis is interesting for a variety of reasons. The most significant of these are [32].

- Modeling: We could create an easy mathematical sample that illustrates the experimental design of Y_1, Y_2,....., Y_T. This model may be dependent on unknown variables, which must be estimated.
- Forecasting: With the inputs Y_1, Y_2, ... and Y_T, we might hope for the parameter Y_{T+L} (L1), as well as providing a new representation of A for the prediction delay.
- Control: It is the required intervention in operations generated by Yt values so that future values are changed to produce a new result [32].

In time series, only one variable record is referred to as a univariate. However, if records are taken from more than one variable, they are referred to as multivariates. Time series can be continuous or discrete. Observations are measured in a series-connected time in each time condition, while the observations are measured in a separate time series at separate time points [33].

2.7 Time Series Analysis (TSA)

Time series analysis is a specific method of analyzing a series of data points collected over some time. In time series analysis, analysts record data points at consistent intervals over a specified period rather than simply recording data points sporadically or randomly. Because time series analysis involves many categories or differences in data, analysts sometimes have to make complex models, and models that are too complex or try to do too many things may fail to distinguish random error from true relationships, making the analysis skewed and predictions unrealistic [32].

The time series are usually very long, it enables them to analyze the data and make predictions from subsequent values that fall within the series that can be expected from each other. Figure (2.3) Standard Model for Time Series Prediction Task [34].

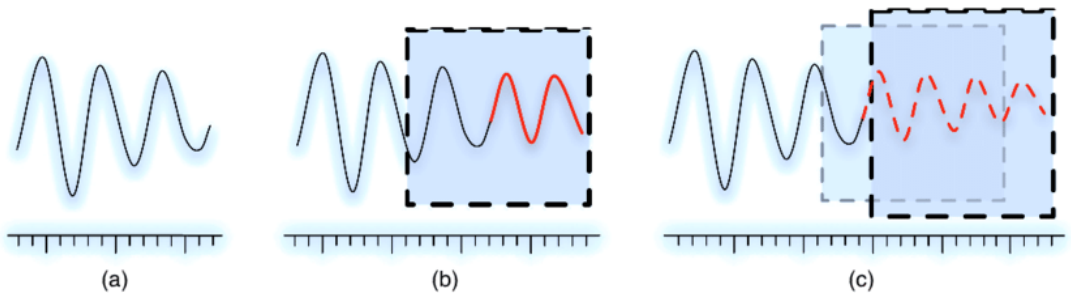


Figure (2.2): The Standard Model of a Time Series

(a) The input time series may show periodicity and thus a predictable structure.

(b) The purpose is to forecast the highest numeral of data points that should go into the forecast window.

(c) The task grows difficult when it arrives at recursive prediction i.e. long-range forecasting of a time series means reusing the previous forecast values as inputs to keep forecasting [35].

2.8 Artificial Intelligence (AI)

Artificial intelligence (AI) is one of the most sought-after technologies of our modern age. Interest in the use of artificial intelligence continues to develop and industrial growth in particular. The emergence of smart technologies increasing the demand for innovations that support artificial intelligence. Data analytics enables the use of AI for short-term decision-making and inference [36].

Artificial intelligence (AI) techniques have developed greatly and the complexities of the problems Machine learning (ML) technology has emerged, and with the passage of time and need, deep learning (DL) techniques have emerged, where all technologies have contributed to solving many complex problems. Especially in the oil field through data from oil wells, the oil and gas industry now relies heavily on various analytical modeling techniques to get the best results [37].

2.9 Machine learning (ML)

Machine learning is a subset of artificial intelligence. To understand the geophysical information of the Earth, various types of data are collected from the surface and subsurface in the oil and gas industries. Sensors must be prepared to collect large amounts of data. This data must be drawn and analyzed using technical analysis and machine learning intervention [38].

In ML, there are many ways that computing devices can solve problems by learning from experience. The objective is to create mathematical models that can be trained to produce useful outputs when entering the data on which the model is trained. Machine learning models are provided training data and some of them are tested and tuned to produce accurate predictions of the training data by optimization,

prediction, and regression algorithm. The main goal of models is to be able to generalize gained experiences and make valid predictions for new data [39].

Four types of machine learning algorithms are commonly used, supervised, unsupervised, semi-supervised, and reinforcement. In supervised learning, what has previously been learned is used to analyze new data, whereas unsupervised algorithms can infer from new datasets as shown in Figure (2.3) [40].

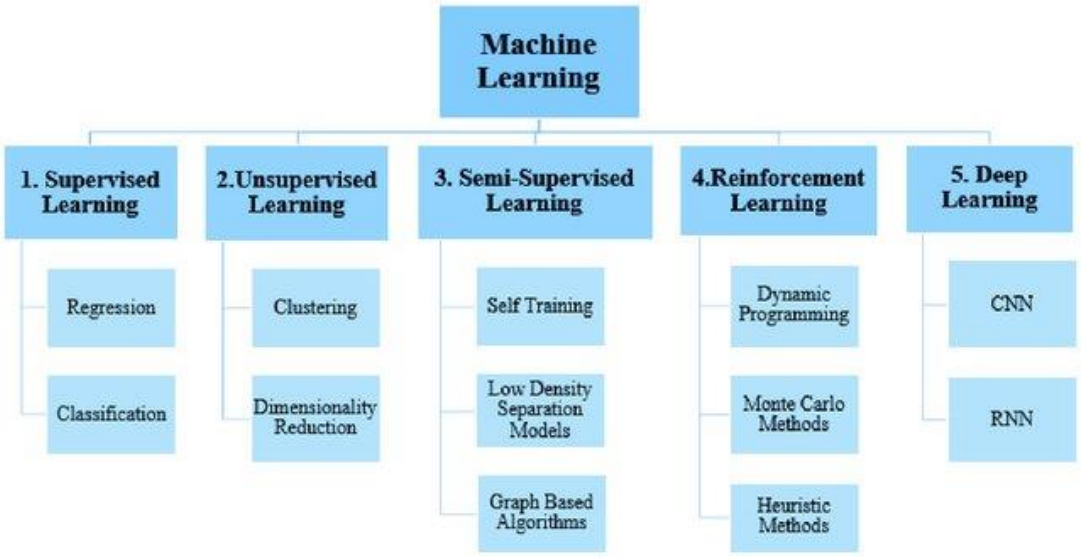


Figure (2.3): Techniques of Machine learning[41].

The proposed method adoption of the very first type of machine learning (supervised learning). As well as some of its algorithms, that were used in detail.

2.9.1 Supervised Learning (SL)

(SL) is a machine-learning model to get the information about a system's input-output relationship established on a given set of paired training samples between the inputs and the outputs. Since the output is

classified as input or supervised data, Input-output training data is also known as labeled training data or supervised data [42].

2.9.2 Unsupervised Learning (UL)

Unsupervised learning (UL) models are a type of machine learning (ML) model that deals with dimensionality reduction, data factors, decoherence, and learning representations between data. UL models are gaining popularity because of their ability to learn without any predefined naming and to reduce noise and redundancy between data samples. However, due to the limited data available to learn, diversity, and complex dimensions, the generalization of UL models for different applications such as image creation, compression, coding, and recognition faces unique challenges [43].

2.9.3 Semi-Supervised Learning (SSL)

Several semi-supervised learning (SSL) algorithms have been developed for learning from both labeled and unlabeled data. Therefore, researchers provide a natural way to represent data in its graphs, these graphs provide a natural way to represent data in a variety of areas. Chart SSL algorithms that combine these two types. These shown to outperform current technology in a variety of applications such as speech processing, computer vision, natural language processing, and other areas of artificial intelligence [44].

2.9.4 Reinforcement Learning (RL)

Reinforcement learning is a branch of machine learning. the mechanism is based on rewarding or punishing skills and behaviors through an agent, in which the agent can react to actions by experiencing all actions in the procedural environment. It can correct skills and punish

skills. It also can carry out actions through feedback and return the state and then move to the next state to gain the required skills [45].

2.10 Support Vector Regression (SVR)

SVR is a supervised learning algorithm used to predict discrete values. SVR uses the same principle as SVM. The basic idea behind SVR is to find the best suitable font. In SVR, the best-fit streak is the super level with the maximum number of points [46]. As in Algorithm (2.1).

Algorithm (2.1): SVR Algorithm

Begin

Phase training dataset:

Step 1: Reading the dataset.

Step 2: Set an initial value of weight

Step 3: Set an initial value of Bias

Step 4:Parameter C, Kernel Type, and Kernel Parameters and calculate the kernel according to Equation (2.4)

Step 5: Set ε to 0.01 and vary C

Step 6: Train the dataset

Step 7: Obtain the best value of C

Step 8: Use the best value of C and ε on the training dataset

Step 9:Update the values of weight and Bias

Testing Phase :

Step 10:Select the testing dataset

Step 11: Implement the test by using the best value of the weight, bais C and ε from the training

Step 12: Implement the prediction

Step 13: Calculate RMSE, MAE using Equations (2.20) and (2.21)

End Algorithm

SVR is a supervised learning algorithm. SVR is the same as SVM but SVR is slightly different from SVM in that the base image in SVR is best suited for regression sequence detection. For example, in additional regression examples that test to reduce the error between the true value and the expected value. SVR tries to provide the best streak for two values, and the best streak is the best [47].

The optimum value that the technologies seek to achieve, is the distance between two or more types of sample points, which is the reduction of the total difference between the sample points and the hyperplane. SVR uses the "kernel" to solve the problem of non-linear regression to obtain a higher regression, as well as to find the optimal level of separation of the sample points [48].

SVR is the most favorable approach to solving regression issues. So used to predict time-series, geology, and other fields this technique includes exhibited effective effects in multiple different applications and areas of research [49][50].

The traditional insensitive ε loss function can be formulated in SVR is as follows:

$$Minimize\ \frac{1}{2}\ \|w\|^2 + C\sum_{i=1}^{n}(x_i + x_i^*) \qquad (2.2)$$

The constraint values are subject to processing for the largest variance values from the expected ones (yi) and the similarity is poor as it is accepted in the following optimization condition [51]:

$$\begin{aligned} subject\ to\ y_i - (w^t\varphi x_i) + b) \ &\leq \varepsilon + x_i \\ w^t\varphi(x_i) + b - y_i \ &\leq \varepsilon + x_i^* \\ x_i x_i^* &\geq 0 \end{aligned} \qquad (2.3)$$

Y_i = Values Features

X_i = Values Label

$\varepsilon = deviation$

b = Constraint Values

By using Lagrange multipliers $y_i\, y_i^*$ and kernel deception, It results in the following double problem:

$$maximum\, w(y, y^*) = -\frac{1}{2}\sum_{i,j=1}^{n}(y_i - y_i^*)k(x_i, x_j) - \varepsilon\sum_{i=1}^{n}(y_i + y_i^*) + \sum_{i=1}^{n} y_i(y_i + y_i^*) \quad (2.4)$$

$$subject\ to \quad \sum_{i=1}^{n} y_i\left(y_i + y_i^*\right) = 0 \quad (2.5)$$

Eventually, the procedure will be:

$$f(x) = \sum_{i=1}^{n}\left(y_i + y_i^*\right) k\left(\mathrm{x}_i, \mathrm{x}_j\right) + \mathrm{b}^* \quad (2.6)$$

2.11 K-Nearest Neighbors (KNN) Regression Algorithm

(kNR) is an exemplar of a supervised learning method. Supervised learning infers a function learner from training data T, which is a set of training examples known as samples [52].

The kNN regression evaluation is dependent on finding the proper number of k nearest neighbors to use in the prediction. A small k with less adjacent data narrows the regression region, resulting in less bias but more variance. A larger k, on the other hand, involves more contiguous data, which allows the regression factors to vary over a larger area,

resulting in greater bias and lower variance Steps to implement the KNN algorithm [53].

Step 1: Initially, K number of neighbors is selected.

Step 2: Depending on the distance law, the point closest to the value of K is taken to be the new point

Step 3: The number of data points, neighbors to K, is calculated in each class.

Step 4: After the result set, the new data point is collected where you can calculate the neighbor values points as shown in algorithm (2.2) [54].

Algorithm (2.2): KNR Prediction

Begin

Phase Training Dataset:

Step 1: Reading The Dataset.

Step 2: Assign the Initial Value of K=1

Step 3: Calculate The Distance (D) Between Each K Neighbor Points

Step 4: Calculate the average of those neighbors to obtain predictor P

Step 5: Determine Nearest Neighbors Based On Minimum Distance To Equation (2. 7)

Step 6: For n =0 :

Step7: Calculate The Distance Between This Predictor And Its Neighbors To Obtain Error ($E_{\_k}$ (n)).

Step 8: n=n+1

Step 9: If $E_{\_k}$ decreases, Increase K and go to **step 3**

Step 10: Else stop and return K=argmin ($E_{\_k}$) [K at the minimum error]

Testing Phase:

Step 11: select the testing dataset

Step 12: Calculate the average of K neighbors

Step 13: Measure the distance between the test data and the average K neighbors

Step 14: implement the prediction

Step 15: Calculate RMSE, MAE using Equations (2.20) and (2.21)

End Algorithm

The KNN classifier uses a distance function to determine how different or similar two examples are. The equation defines the usual Euclidean distance d(x,y) between two instances x and y

$$d(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2} \quad (2.7)$$

Where, x_k represents the k^{th} the feature element of instance **x**, y_k represents the k^{th} the feature element of instance y and n is the total number of features in the dataset.

For many years, pattern recognition and data mining have used and studied KNR regression. To improve the performance of the kNN regression in the regression analysis, a kernel estimator based on some asymptotic properties of KNR was used [55].

2.12 Deep Learning (DL)

Deep learning refers to the acquisition of some type of knowledge Deep learning algorithms are widely used to increase complexity in contrast to standard machine learning algorithms. Deep learning also refers to the use of artificial neural network architectures that have a very large number of processing layers, unlike other methodologies of traditional neural networks. Dealing with complex deep learning methodologies gives the flexibility to analyze large amounts of data and

this gives a distinction to applications that use these methodologies such as industry, agriculture, and other applications to get better results [56].

So that we can simulate the activity of the human brain is by building a neural network. The deep neural network framework consists of three layers of input and outcome and a hidden layer that is usually placed in the input and outcome layer. The concept of deep learning is the total dependence of the deep neural network. It is a subsection of machine learning and uses aspects of artificial intelligence to classify and arrange data as shown in Figure(2.4) [57].

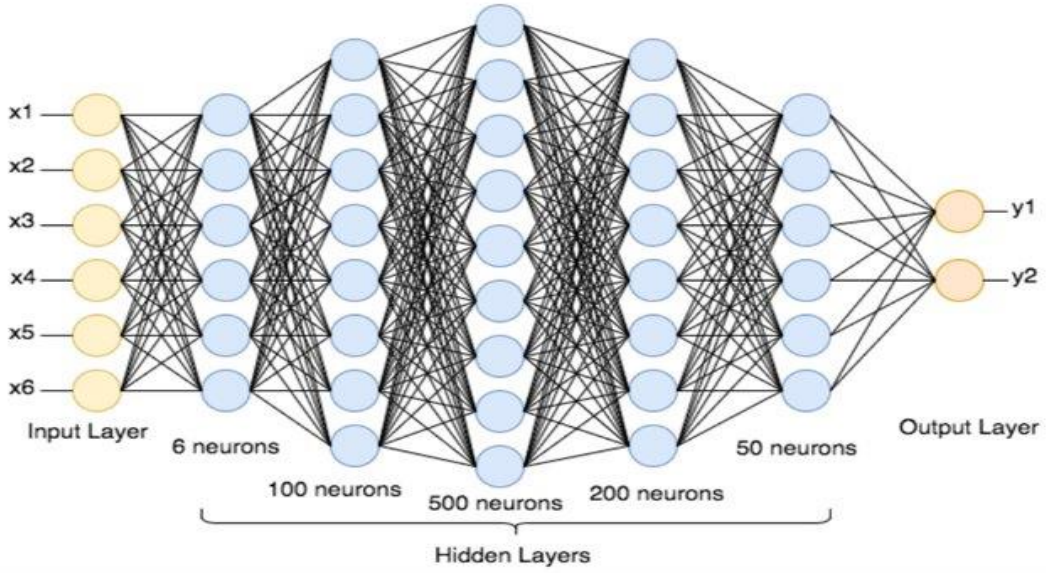


Figure (2.4): General Deep Neural Network Structure

The most popular types of neural networks are Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs)[58].

As there are very too several types of DL models, the theoretical overview will only cover Supervised Learning, more specifically prediction problems with RNNs, which are the basis for the LSTM and GRU and mixed with (ANN).

2.13 Recurrent Neural Networks (RNNs)

RNNs are at the heart of modern approaches to a large number of common tasks [59].

RNNs are useful for modeling sequential and time-series data. However, when using RNNs to inform decision-making, predictions by themselves are not enough, we also need estimates of predictive uncertainty [60].

RNN represents a layer of the neural network. Because there are multiple layers stacked upon each other, an RNN is a deep neural network. They differ from other neural networks in the sense that the hidden layers are linked to time [61].

RNN is used to capture information across time dimensions and store it within the network. Being an effective tool to achieve desirable results in various fields of machine learning, few versions of (RNN) such as (GRU) and (LSTM) have been developed [62].

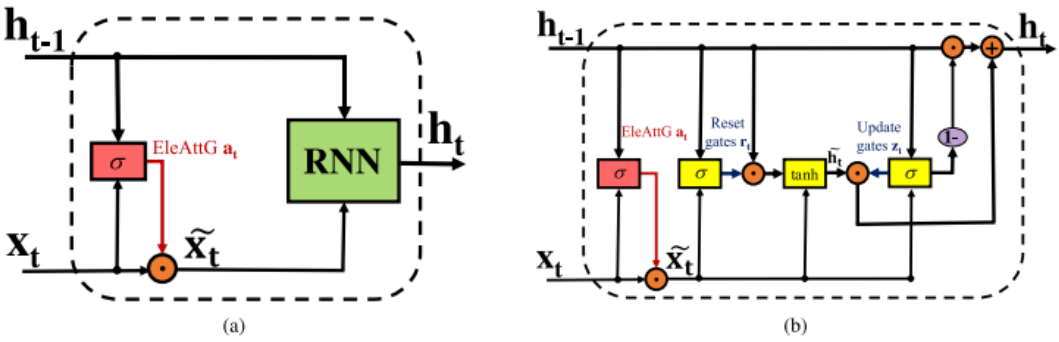
RNN has significant advantages over temporal data processing. Existing RNN models usually train several temporal data that are correlated separately and ignore the correlation between the data [63].

A structured iterative interaction takes place inside the RNN which enables the sequential data processing by carrying the result of the previous time action as input to the current time action, in the time step the RNN neurons perform the same process to include current and prior input information to represent the hidden state of that time step. Often, RNNs have problems in learning long-range dependences and may fail to control long-range neural synchronization in recurrent escalation networks due to spontaneous activity [64].

To handle this issue, there is a lot of suggested by the researcher approaches as gate-dependent, LSTM, GRU in RNN neurons to control

the information flow. Gates optionally provide a means to allow information to pass through or pause quietly, balancing current period information contributions with historical information [65].

The illustration in Figure (2.5) of the elemental attention gate (shown in red) for (a) a general RNN block, where a standard RNN architecture can be an (RNN), (LSTM), or (GRU) and (b) a (GRU) block consisting of an array (N) GRU neurons. Also In the graph, each line contains a vector. Circles indicate element-wise operation, for example, element vector outcome or vector expansion. Yellow squares indicate the original (GRUs) with the output After N, the red square indicates output dimensions D, the same as the x_t input dimension [65].



(a) Generic RNN block (b) Generic GRU block

Figure (2.5): Generic RNN and GRU block [65]

In time-series when data are interdependent, learning performance may deteriorate due to its insufficient ability to model long-term dependencies. By scenario where the current data points are related to the previous data points. Long-term dependence is the basis for time series forecasting. In addition, it feeds front-end neural networks that constrain inputs and targets to be vectors of fixed length, which also makes them unsuitable for sequential learning such as time-series unless they are tuned by ANN [62][66].

RNN is an effective tool for capturing information especially time-series across time dimensions and storing it within the network. Which are developed from versions of neural networks. Acceptance of feedback connections that can produce prior context information is a vital update in RNNs compared to feed-forward neural networks. In other words, RNNs have their internal memory, which gives them an advantage over other neural networks [67].

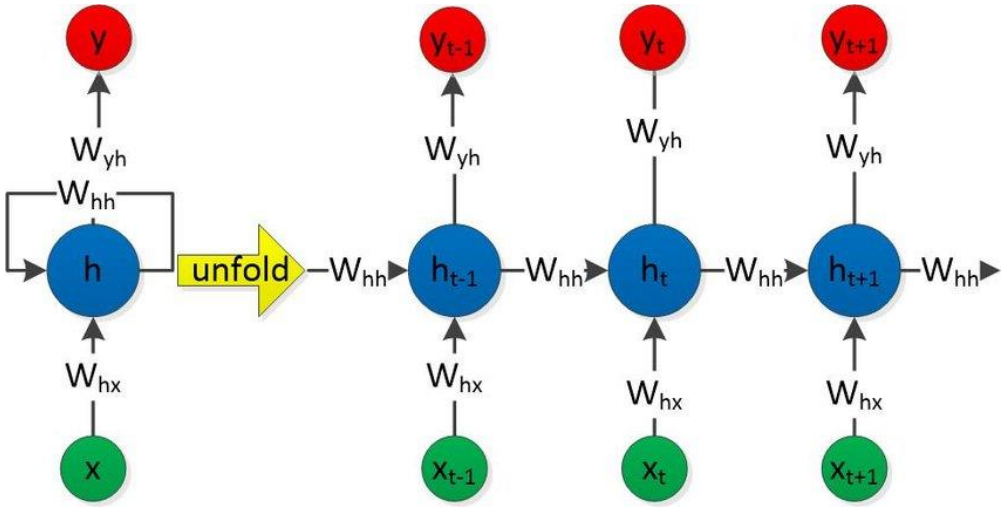


Figure (2.6): The Architecture of RNN [67]

Figure (2.6) Shows the architecture of RNN. Given an input time-series , $x = \{x_1, x_2, \ldots\ldots, x_T\}$. RNN computes the hidden state sequence $h = \{h_1, h_2 \ldots\ldots, h_T\}$ as well as the output sequence $y = \{y_1, y_2, \ldots\ldots, yt\}$ iteratively using the following set of Equation.

$$h_t = f(W_{hx} X_t + W_{hh} h_{t-1} + b_h) \qquad (2.8)$$

And

$$y_t = g(W_{yh} h_t + by) \qquad (2.9)$$

In Equation (2.8), and Equation (2.9)

W_{hx} : Input Hidden Weight

W_{hh}: Weight State Hidden

W_{yh}: Output Hidden Weight

and b_h represents priority the hidden layer and the resulting layer is serialized. In addition, $f_{(0)}$ and $g_{(0)}$ are the hidden layers and the output layer the activation procedures [68].

The RNN uses the hidden state (h_t) in the time step (**t**) to save the network. The hidden state captures all the information included in the previous time steps [67].

2.13.1 Long-Short Term Memory (LSTM)

Used to smooth out the spread and fade of gradients when learning memory dependencies. For this reason, which is the standard for many serial modeling tasks, LSTMs and RNNs are widely relied upon [69].

LSTM models continuous sequence data, as well as depending on whether the sequences are tested uniformly. In this way, all data measurements are subject to duplicate fabric correlation .However, it is difficult for LSTM to model unequal periods in data sequencing. This occurs when measurements in a processing time series are not sampled at a uniform frequency. The problem of non-uniform sampling often occurs in industrial processes, which hinders the application of LSTM [70].

RNNs are made up of a series of repeating neural network modules; this repeating module in standard RNNs would have a very basic structure, such as a single **tanh** layer as illustrated in Figure (2.7) [64].

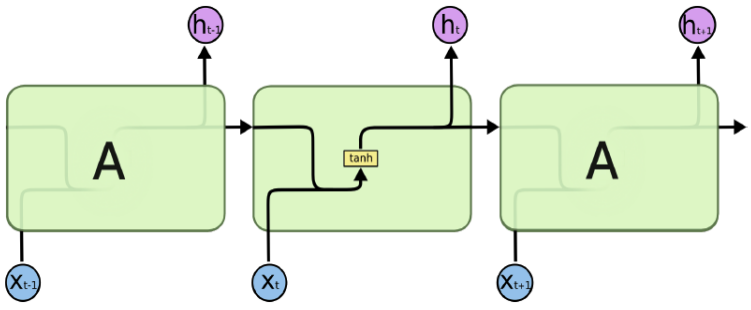


Figure (2.7): Module in a Standard RNN

LSTM is a long-short term memory for recurrent neural networks RNN , which is considered one of the most famous and powerful neural

networks known, and it has evolved significantly in how to deal with continuous temporal data [71].

We also see the distinction of internal and external iteration with deep temporary learning by dependency in LSTM with memory blocks, which is a rather unique feature [64][71].

LSTM layers consist of blocks between memory that are repeatedly connected to a cell or memory unit. These cells consist of gates to determine when the past hidden states of the memory cell and additional update cells were lost. Thus allowing the network to make use of material knowledge as shown in Figure (2.9) [47][45].

To show how to control the input gate, with the input feature, x_t takes the input data x, at time t, the input data flow to the cell. The forget gate determines when the contents of the cell's internal state are forgotten, and the output gate controls the flow to the outlet. The function of the cell in this regard is as follows [72]:

$$\mathrm{i_t} = \sigma\,(u_i x_t + w_i h_{t-1_\mathrm{i}} + \mathrm{b_i}) \tag{2.10}$$

and;

$$f_t = \sigma(u_f x_t + w_f h_{t-1} + b_f) \tag{2.11}$$

And;

$$o_t = \sigma(u_o x_t + w_o h_{t-1} + b_g) \tag{2.12}$$

And;

$$g_t = \sigma(u_g x_t + w_g h_{t-1} + b_g) \tag{2.13}$$

And;

$$c_t = g_t i_t + f_t c_{t-1} \tag{2.14}$$

Finally;

$$h_t = o_t \tanh(c_t) \quad (2.15)$$

The internal redundancy c_t and the current output i_t which is equal to h_t the currently hidden state are both computed at time t using gate parameters U and W (weight matrices) and with b (bias vector) learned in the process.

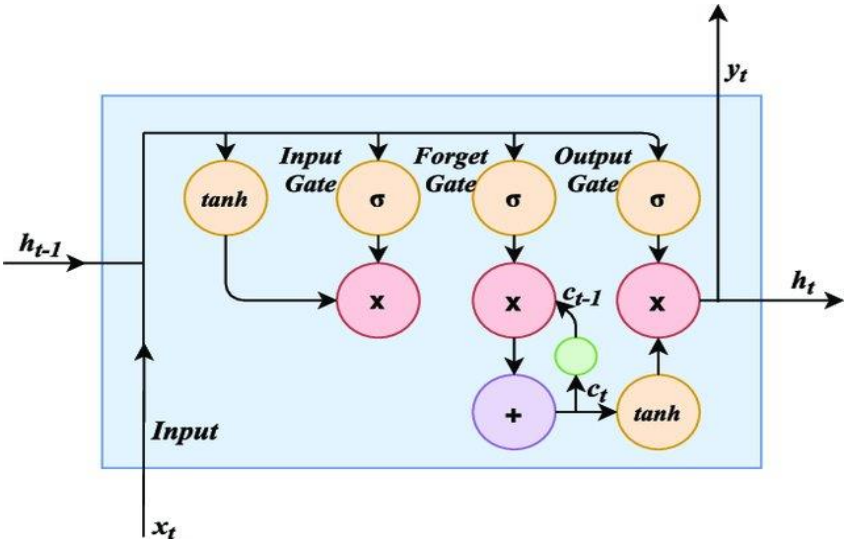


Figure (2.8): Shows Input Feature In LSTM [72]

The LSTM block consists of a cell state and a hidden state. The function of the input gate is to decide which values to update, while the forget gate controls which part of the previous cell state should be forgotten. Based on the input and forget gate, in the output gate the state of the new cell is calculated. In addition, the information that should be delivered to the next node is calculated, that is, it calculates the hidden state of the cell. In this illustration Figure (2.8), three-time steps and a dataset with three sequential data samples ($x_{t-1,+1}$) are visible.

The central LSTM cell is shown in detail to reveal the processes within it. The network is only shown up to the output of the hidden layers (h_{t-1},h_t,h_{t+1}), the output ŷ is not represented here.

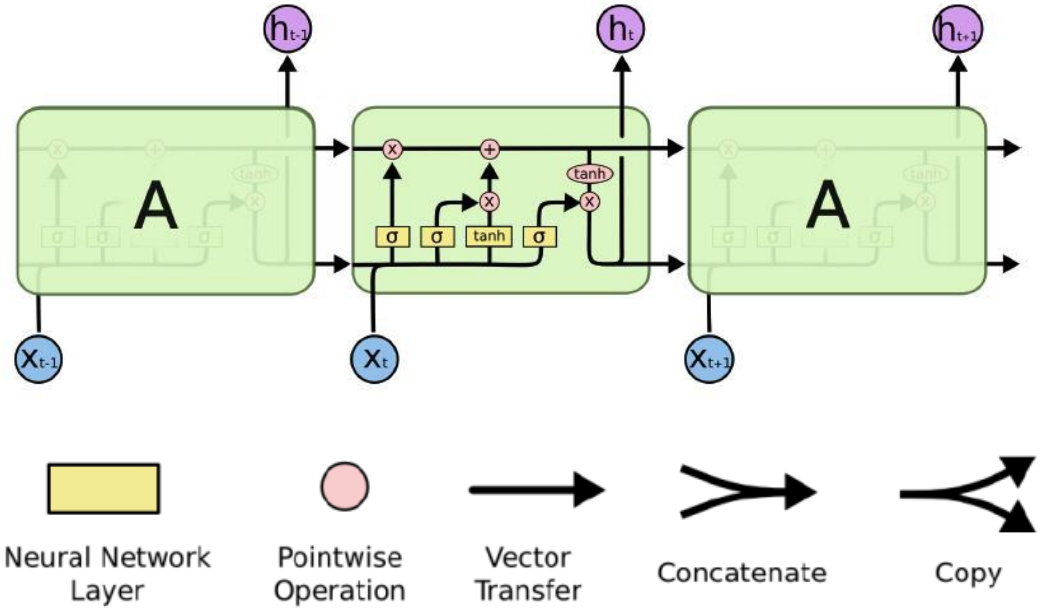


Figure (2.9): The Structure of an LSTM Block [73]

LSTM is a great RNN technology and is the most popular and widely used RNN, with the ability to train long and short-range dependency information as well as the ability to solve color gradient problems, continuous data, temperature, and other sequential applications. The LSTM algorithm is perfect for dealing with problems that have a lot of time-series correlation [74].

As for the input gate, the gate is a layer of sigmoid activation nodes, whose output is multiplied by the output of the *tanh*, the sigmoid of this input gate can stop any element of the input vector not required as this function outputs the values between 0 and 1, according to Equation (2.12). In addition, the forget gate determines whether the previous memory cell is useful for computing the current memory cell. As a result, the forget gate examines the input as well as the previous hidden state. This addition process, instead of multiplication, helps reduce the risk of gradient vanishing. This gate allows the network to understand the state variables that must be remembered or forgotten according to Equation (2.13).

Algorithm (2.3): LSTM Algorithm

Input: X (current input)

H (previous hidden state)

Output :H (current input)

X (previous hidden state)

Begin

Training phase

Step 1: training ={(x_t ,x_{t+1}) t= 1,2,……t_1) and validation set x , validation ={(xt,xt+1), t= t1+1,t1+2,…t2}.

Step 2: initialize w randomly, ± Val

Step 3: adjusting W

Step 4 : for epoch = 1 to n do

Step 4-1: the memory state C_{t-1} is taken by LSTM and it performs a wise multiplication of the elements with the forgot gate (f)

$C_T = C_{T-1} * F_T$, (f gate gives values 0 or 1) ,If f= 0 then past memory state is fully forgotten ,If f= 1 then the past memory state passed to the cell

Step 4-2 : compute new memory state from input state and C'_t layer with present memory state C_t

$$C_t = C_t + (I_t * C'_t)$$

C_t= present memory state at a time step, and it gets pushed through to next time

Step 4-3: apply T and to C_t then do element-wise multiplication with the output gate C, That will be our current hidden state H_t, H_t = Tanh C_T Pass C_t and H_t to next time step, then repeat the process itself.

Step 4-4: compute output error. Xe_{t+1}-x_{t+1} , t= 1, 2 ,…..t1

Step 4-5 : if error ≥ 0.001 then perform backward propagation for all layer n-1 to layer 1

Step 4-6 : update W :W = W +ΔW

Step 4-7: perform forward propagation recurrently to up data the network state

Validation phase

Step 4-8: read validation data

Step 4-9: perform forward propagation recurrently to compute W

Step 4-10: save the current W if epoch ≥ 100 then break

Testing phase

Step 5: read test dataset

Step 6: perform forward propagation recurrently to compute X_e

Step 7: Calculate RMSE, MAE using Equations (2.19) and (2.20)

End Algorithm

2.13.2 Gated Recurrent Units (GRU)

The proposal of varied forms of RNN achieved amazing results for many applications, especially continuous serialization such as the long-term memory module (LSTM) in 1997. But that didn't stop the developers from suggesting algorithms like GRUs which are the simplified performance of LSTM and require less training time with improved network performance based on its continuous data update [75].

proposed a gated recurrent unit (GRU) in 2014 that uses only two gates and can achieve accuracies comparable to the more complicated LSTM for some applications [76].

The Method in GRU is an identical or replica of the parameters of the simple RNN. Only use process update by x corresponding to these

gates are also corrected using random regression through the moment as they seek to reduce the loss/cost function [62][77].

The workflow of a gate recurring unit, for short a GRU, is an RNN iteration but the difference is in the process and gates associated with each GRU. RNN standard often has a problem that needs to be solved, the GRU integrates two mechanisms to operate the gate called the update rate and the reset gate as shown in Figure (2.10) [78].

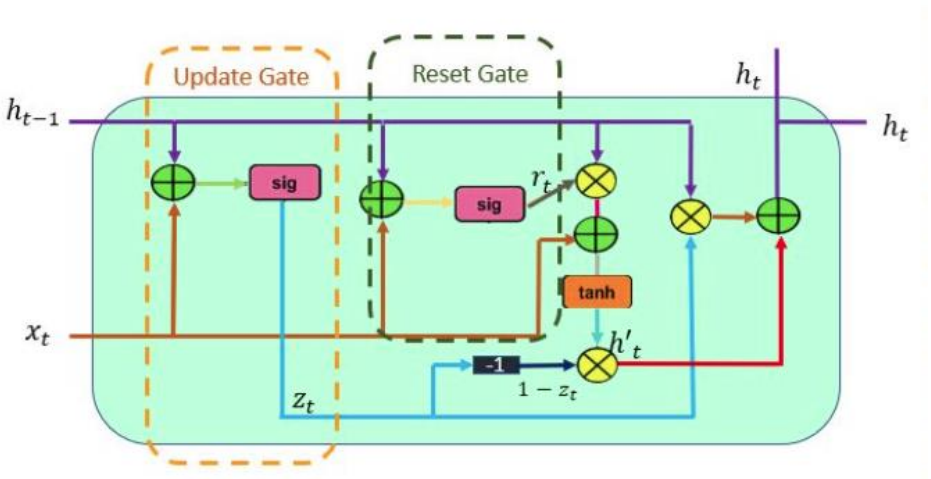


Figure (2.10): Structure GRU

2.13.2.1 Update Gate

Previous data is updated by the update gate to determine its amount to pass through the next state. This is powerful because the model can copy all the information from the past and eliminates the risk of gradient fading [79].

2.13.2.2 Reset Gate

The reset gate of the model is used to determine the value of the cell state for the previous information that needed to be discarded, In short, it decides whether the state of the previous cell is important or not [79].

First, the reset gate comes into play, Saving the data associated with its predecessor (h_{t-1}) to new memory parameters. Previous time

parameters (h_{t-1}) and current time (h(x)) are converted from the correct multiplication using the weight matrix for the following reset gate(h_{t+1}). Then it sends the updated data to the reset gate and multiplies it by the sigmoid function, to get a result rate between [0, 1]. Equation (2.16) is used to describe this operation after summarizing the above steps, the nonlinear activation function is applied and the following sequence is generated [79][62][77].

$$R_t = (W_r * [h_{t-1}] + b_r)$$
$$= (W_{hr}\, h_{t-1} + W_{xr}\, x_t + b_r) \qquad (2.16)$$

where R_t = reset gate, b_r = bias of the reset gate

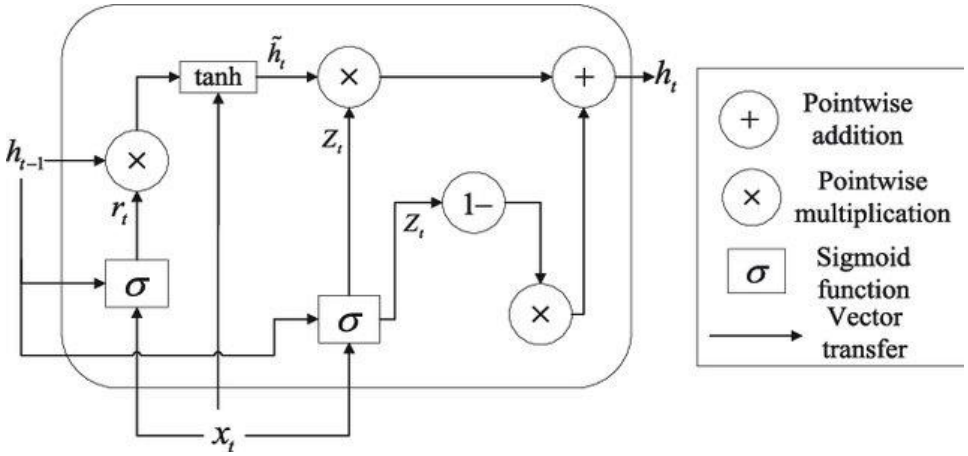


Figure (2.11): Simple Neural Representation [80]

In Figure (2.11), a simple neuronal block model shows how to make the resulting constants ranging from [0 to 1]. The inputs and outputs of the GRU are $x_{(t)}$ and $h_{(t)}$. At $h_{(t+1)}$ represents the future time, and $h_{(t-1)}$ represents states in the previous moment from the current time $h_{(t)}$. Unlike other neural networks, $r_{(t)}$ and $z_{(t)}$ are both basic structures in the GRU, which are called gate reset and gate refresh. The sigmoid function is an activation of the neural network. The resulting filter $x_{(t)}$ is the product of

the value processing by the reset gate [78][81]. In the equations below, the calculation process is illustrated:

$$Z_t = (W_z * [h_{t-1}, x_t] + b_z)$$
$$= (Whz\ h_{t-1} + W_{xz}\ x_t + b_z) \quad (2.17)$$

where z_t = update gate, b_z = bias of the update gate as shown in Equation (2.18). When the reset gate is 0, all memory data is empty. When the reset gate is set to 1, it means that all of the memory data has been transferred to the current GRU [78][81].

$$h_{\tilde{t}} = \mathbf{tanh}(W_{\tilde{h}} * [r_t * h_{t-1}, x_t] + b_{\tilde{h}})$$
$$= (\boldsymbol{tanh}\ (r_t * W_{h\tilde{h}}\ h_{t-1} + W_{x\tilde{h}}\ x_t + b_{\tilde{h}})) \quad (2.18)$$

Where $h\tilde{t}$ = candidate hidden state at time t, $b_{\tilde{h}}$ = bias of candidate hidden state.

GRU can solve the vanishing gradient problem by using a hidden state, it calculates hidden state h_t at time t from the output of the update gate z, previously hidden state h_{t-1} and candidate hidden state $h_{\tilde{t}}$ is calculated in Equation (2.19) [78][81]:

$$h_t = (1 - z_t) * h_{t-1} + z_t * h_{\tilde{t}} \quad (2.19)$$

2.14 Artificial Neural Networks (ANNs)

The biological Neural Network is thought to be the primary natural source principle of the Neural Network, which has approximately 10 billion neurons connected via 100 trillion interconnections in the human brain. Neurons in Neural Networks (NNs) process information and data [82].

The specific neurons communicate with each other via a connection known as synapses, which have a set of variables such as

weights. As a result, Neural Networks have parallel processing distributed systems [83], and Figure (2.12) depicts biological neurons.

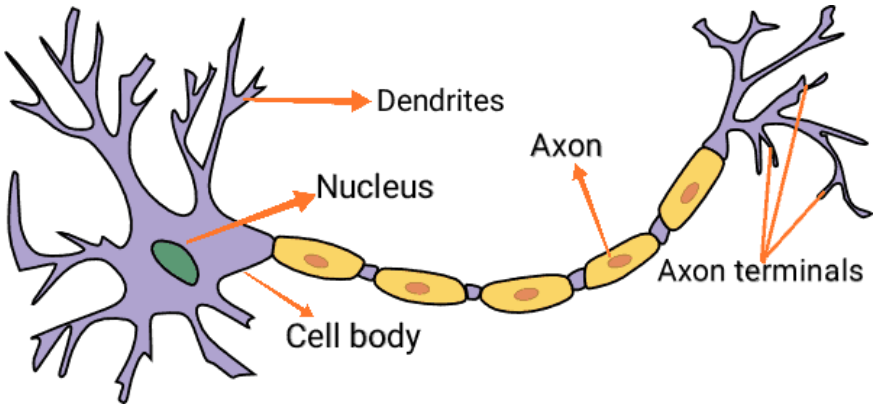


Figure (2.12): Depicts Biological Neurons.

ANN models perform very well in the field of flow prediction. However, there is another area for improving artificial neural network modeling performance; For example, the efficiency of any artificial neural network-based prediction model can be significantly improved by using multiple input parameters chosen from sensitivity analysis and using hybrid models [84].

ANNs are used to predict the behavior of complex systems, and in recent years, the ability of ANN to estimate the viscosity of liquids has convinced researchers to apply this powerful tool in their research[85]. ANNs usually consist of three layers, the external inputs are data, feeders, and densities, and the nodes represent their labels [85][86].

Weighted neural networks make connections between the input and output nodes through the weights of the connected neurons, for which feedback algorithms and regressive functions are used [85].

In the hidden layers, there are neurons present and dependent on them to function the hidden layers, such as the sine function, where the sum of its information comes from input or output nodes through feed-forward or from output nodes through revisions. Where the output value

of the neuron is multiplied by the weighting value after passing the nonlinear function [58] as shown in Figure (2.15).

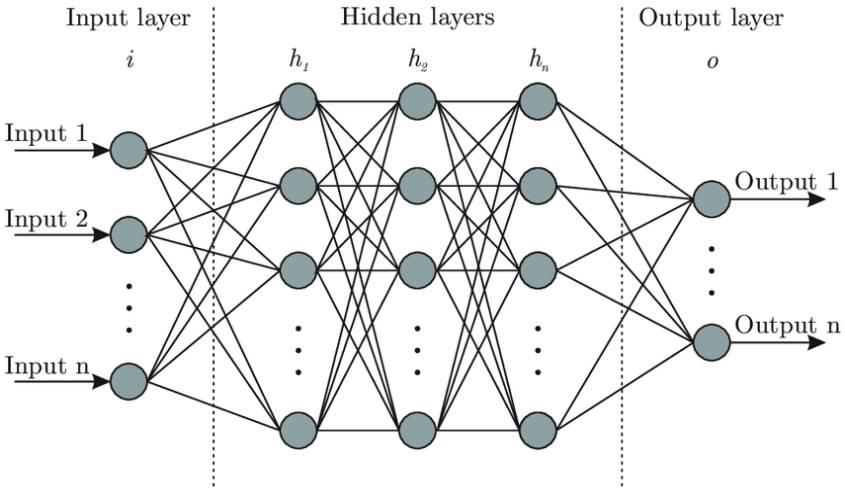


Figure (2.13): Structure of ANN

2.15 Methods of Evaluation

There are many criteria for evaluating the performance of the proposed ML techniques, including the commonly used statistical model, including mean absolute error (MAE) and root mean square error (RMSE).

MAE is the mean absolute error of the expected y-values relative to the actual x-values and is given by[87].

(2.20)

$$MAS = \frac{1}{N}\sum_{i=1}^{n}|X_i - y_i|$$

RMSE is a measure of the level of the spread between the actual x-values about the average of the expected y-values. can be expressed as [88]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n}} \quad (2.21)$$

$x = Predicated\ value\ of\ x$

$y = mean\ value\ of\ y$

$n = the\ number\ of\ obsevation\ in\ the\ data$

Chapter Three

The Proposed System

Chapter Three

The Proposed System

3.1 Introduction

porosity forecasting and evolution through seismic data play an important role in oil reservoir exploration and production of oil. With the advancement of artificial intelligence and machine learning, many achievements have been made in oil reservoirs by predicting the reflected signals for seismic detection of reservoirs. However, due to factors such as economic cost. Type of applications, and tools, it is often difficult to obtain suitable and efficient samples of oil tank seismic data for machine learning. In this case, this will greatly affect the prediction accuracy of the oil reservoir and its production.

This chapter presents a proposed model for a prediction of porosity resulting from seismic data of oil reservoirs by testing algorithms dealing with time series such as SVR and KNR as well as LSTM and hybrid algorithm (LSTM with GRU) and then the proposed model will be produced the best result among them. Both systems share the same Seismic-dataset and structure, but the first model is a prediction based on a machine learning algorithm, and the second model prediction is based on a deep learning algorithm.

This chapter also explains how to create a new Seismic dataset in detail and the application used to collect and process the data, in addition to explaining each stage of the proposed system and the tools and algorithms used.

3.2 A Structure of the Proposed System

The basic idea of the proposed model is that porosity is considered one of the most petrophysical factors, which has an important impact on the determination of oil reservoirs, so determining it with minimum error is of great importance to reduce the cost and time to discover oil locations.

Therefore, the proposed works aim to design and implement the model to predict a porosity resulting from the seismic dataset of oil reservoirs based on machine learning and deep learning algorithms.

In addition, the results of the machine learning algorithms will be compared with the corresponding results of the deep learning algorithms to determine the best performance of the algorithm in determining the porosity resulting from the database.

Figure (3.1) shows the general block diagram of the proposed model, which includes eight stages: Acquisition and load the seismic dataset, preprocessing dataset, selection features using correlation matrix, reshape dataset, splitting the dataset, prediction stage which is divided into two branches:

(i) uses machine learning algorithms, which are Support Vector Regression (SVR) and K Neighbor Regression (KNR).

(ii) uses deep learning algorithms which are long-term memory (LSTM) and Gated Recurrent Units (GRU), and the last stage Performance evaluation using (MAE & RMSE) measurements.

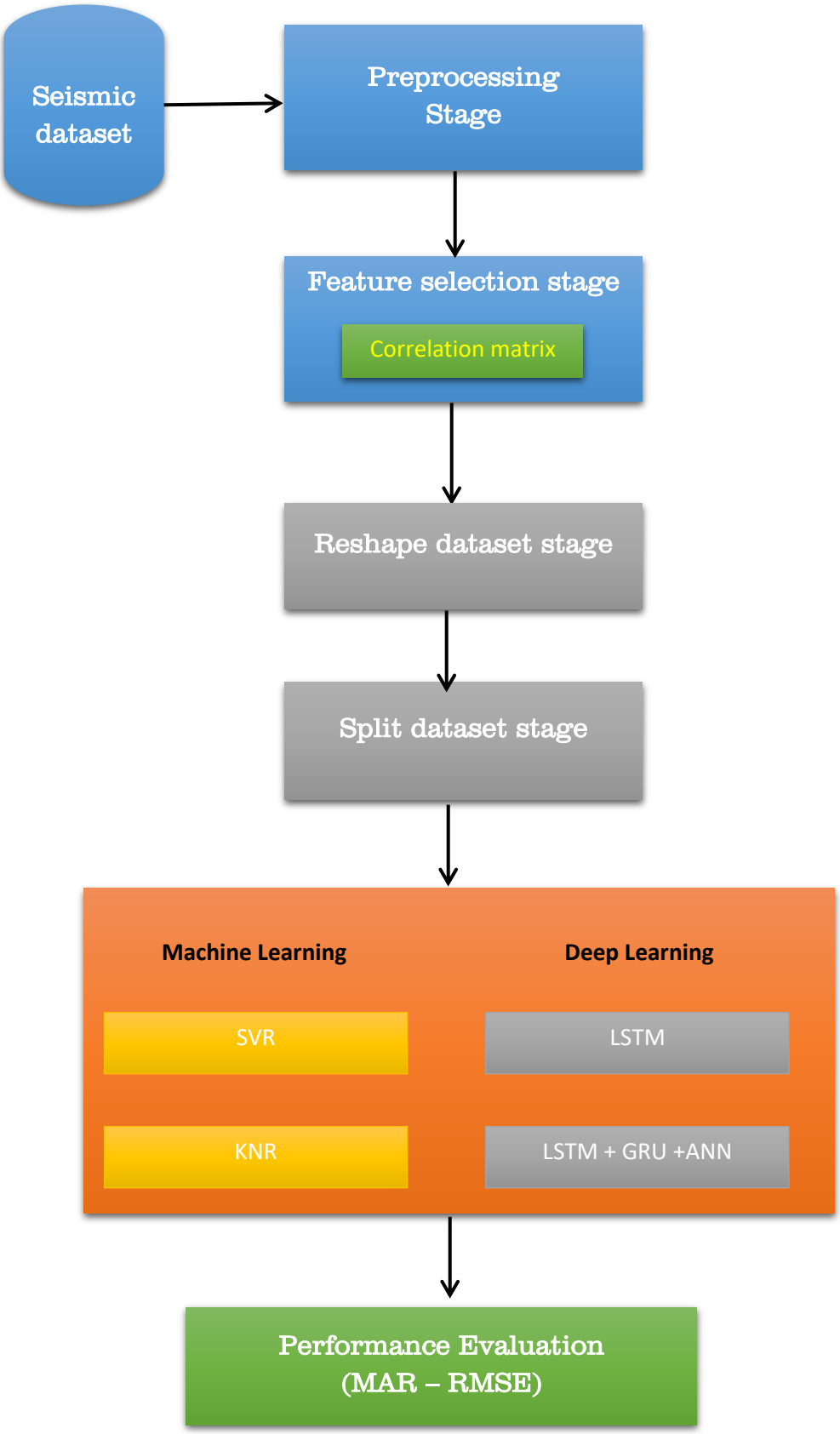


Figure (3.1): A General Block Diagram of the Proposed Model.

Each stage of the proposed model depends on a certain algorithm or techniques that cooperate in order to achieve the objectives of the proposed model. These stages are explained in detail as follows, as shown in Figure (3.1).

3.2.1 Acquisition And Load Seismic Dataset Stage

Data was collected from the Ministry of Oil on one of the oil wells that are still under exploration. One of the advantages of this data is that it is processed by special programs. Seismic data is passed through several stages of processing until it reaches its final form, and then kept in an SEG-Y format. Seismic data is an example of very big data, so an SEG-Y file is very large and can consist of thousands of rows and columns.

This work will choose to analyze not the seismic data itself but the attributes of the seismic data because the features are more useful than raw seismic data in that many of these features will be nonlinear, thus increasing predictability. The second reason is that there is often a benefit in dividing the input data into parts. This process is called feature extraction, and it can often dramatically improve performance by reducing the dimensions of the data before it is used to train the system.

22 seismic features were derived from the seismic data. 26 seismic features represent the maximum number of features in Emerge. In Emerge, the size of the seismic data is indicated by the number of impacts. Seismic features consist of thousands of rows and columns, and each column is called a trace, so each feature contains hundreds or thousands of traces. Dealing with all traces is a very extensive and time-consuming method. For this reason, the algorithm was applied to one trace per feature.

Table (3.1) is illustrated each Feature's name, abbreviations, and description. The seismic dataset deals with porosity prediction and is the objective that attempts to predict the magnitude of this property at all locations from seismic data. The dataset contains 1235 samples of seismic data that are inversions, for which time series have been used rather than depths of seismic inversions as they have the potential to be sensitive to very high prediction cycles.

Table (3.1) Seismic Dataset.

No	Seismic Feature	Abbreviate	Description
1	Amplitude Envelope	AE	Magnitude Of The Complex Trace, Defined By The Trace And Its Hilbert Transform
2	Amplitude Weighted Cosine Phase	AWCF	Product Of The Amplitude Envelope And The Cosine Of The Instantaneous Phase.
3	Amplitude Weighted Frequency	AWF	Product Of The Amplitude Envelope And Instantaneous Frequency.
4	Amplitude Weighted Phase	AEP	Product Of The Amplitude Envelope And Instantaneous phase.
5	Apparent Polarity	AP	Average Of The Amplitude Spectrum Over A Small Window Around The Time Sample
6	Average Frequency	AF	Signed Amplitude Envelope
7	Cosine Instantaneous Phase	CIP	The Cosine Of Instantaneous Phase

8	Derivative	D	Derivative Of The Input Trace Calculated As The Difference Between Adjacent Samples This Tends To Increase The Frequency Content And Changes Peaks To Edges Sharpening The Seismic Image. If You Use A Single Sinusoidal Component, Differentiation Increases The High-End Frequency Content Of The Signal And Rotates Is By A 90 Phase Shift
9	Derivative Instantaneous Amplitude	DIA	Derivative Of The Amplitude Envelope Of The Input Trace.
10	Dominant Frequency	DF	Maximum Of The Amplitude Spectrum Over A Supply The Frequency Small Window Around The Time Sample Parameters.
11	Filter 5/10 – 15/20		5/10 – 15/20 Bandpass Filter Slice Of The Input Trace
12	Filter 15/20 – 25/30		15/20-25/30 Bandpass Filter Slice
13	Filter 25/30 – 35/40		25/30-35/40 Bandpass Filter Slice
14	Filter 45/50 - 55/60		45/50 – 55/60 Bandpass Filter Slice
15	Filter 55/60 - 65/70		55/60 – 65/70 Bandpass Filter Slice
16	Instantaneous Frequency	IF	Not That +180 And -180 Should Be Give The Same Color In The Display.
17	Instantaneous Phase	IP	Phase Of The Complex Trace, Defined By The Trace And Its Hilbert Transform.
18	Integrate	I	The Integration Of The Data Trace

19	Integrated Absolute Amplitude	IAA	Integration Of The Amplitude Envelope Of Input Trace.
20	Quadrature Trace	QT	The Out Of Phase Component Trace, With A -90^0 Phase Rotation Of The Input Trace.
21	Second Derivative	SD	Derivative Operation Applied To The Derivative Of The Input Trace.
22	Second Derivative Instantaneous Amplitude	SDIA	Derivative Operation Applied To The Derivative Of The Amplitude Envelope Of The Input Trace.

3.2.2 Preprocessing Dataset Stage

After the collected seismic dataset, the preprocess includes checking the quality of the seismic data and ultimately improving the types of inputs, chosen steps, and time frames. It directly affects the prediction model's results and their accuracy.

Preprocessing seismic datasets involves three sub-steps compute Mean Absolute Error (MAE), check data types where only numerical data (float64) will be used in prediction modeling, and check for missing data to ensure there are no missing values in the data). The details of the preprocessing steps are present in the algorithm (3.1) .

Algorithm (3.1): Preprocessing Seismic Dataset
Inputs: Load Seismic Dataset N=number of features in a dataset **Output:** Clear Seismic Dataset
Begin Step 1: Read the dataset. Step 2: **For** i=0 to N **do** Step 3: Compute the MAE of the feature using equation (2.20) Step 4: Check the missing data in the feature Step 4-1: Find the Null cell (NaN) then replace the Null cell with feature MAE. Step 5: Go to step 2 and read the next feature until complete all Features are. Step 6: Check the data type in the feature must be =float64 Step 7: **End For**. **End Algorithm**

3.2.3 Feature Selection Stage

Feature selection is the third stage in the proposed system. It is the process of reducing the number of features to improve performance and reduce the computational cost of the proposed model. The seismic dataset consists of 22 features; the correlation matrix method is used to identify the important features as shown in Figure (3.2).

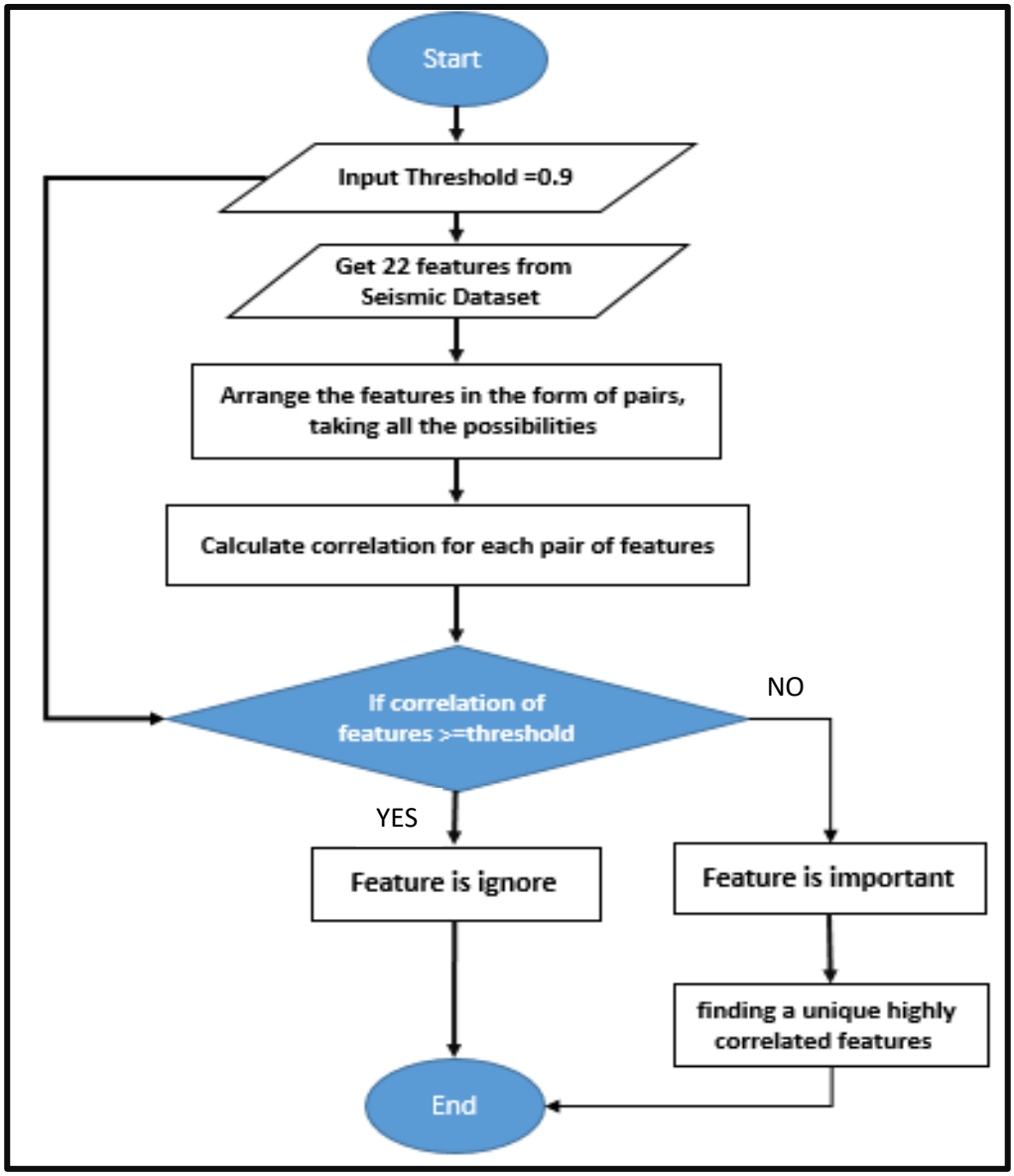


Figure (3.2): A Flowchart of the Correlation Matrix-based Feature Selection Method.

The correlation matrix creates a 2-domination matrix, where the rows and columns represent the 22 Features of the seismic dataset. Each location in the correlation matrix includes the value of the correlation coefficients of two features or pairs of features using Equation (2.1). The values of the correlation matrix are limited in the range [-1,0,1]. When the correlation value approaches 1, the correlation between the features is strong, but if it is 0, it is weak, and between them, the correlation value is

medium. In this method, the absolute is used to of the negative values of the correlation coefficient.

As shown in Figure (3.2), any feature with a correlation value > = Threshold = 0.9 is considered to be an important feature. When the value of the features is less than the threshold, it ignores this feature.

Based on the correlation matrix, if two features are highly correlated, only one of them will be kept. Features that correlate with another feature by threshold or above will be removed. Therefore, 4 highly correlated features are removed which are: [CIP, DF, IP, QT] and will keep 18 Features which are: [AF , AP , AWCP , AWP , D , DIA , F15-36 , F25-40 , F35-50 , F45-60 , F50-65 , F55-70 , I , IAA , IF , SD , SDIA].

3.2.4 Reshape Dataset Stage

This stage aims to reshape the input data into 3 dimensions matrix [samples, timesteps, features] to optimize preprocessing input data into prediction models. For the prediction of time-series data, the model has to estimate features from the previous values. So, if the desired output is $f(t)$, the input feature should be $f(t-1)$.

Time series supervised function is designed to take samples at time $\{t-\text{n}, t-n-1, \ldots, t-1\}$ as input for the machine learning and the output will be sampling at time $\{t, t+1, \ldots, t+n\}$. For the current application, the seismic data are sampled every second, and hence the input is the previous second and the output is the current second, so timesteps are equal to one and the input matrix-shaped in the form [samples, 1, features]. Figure (3.3) shows the time-steps for the data when

the number of features (f) =18 and input data at a time (t-1) to the prediction model and the output data at a time (t).

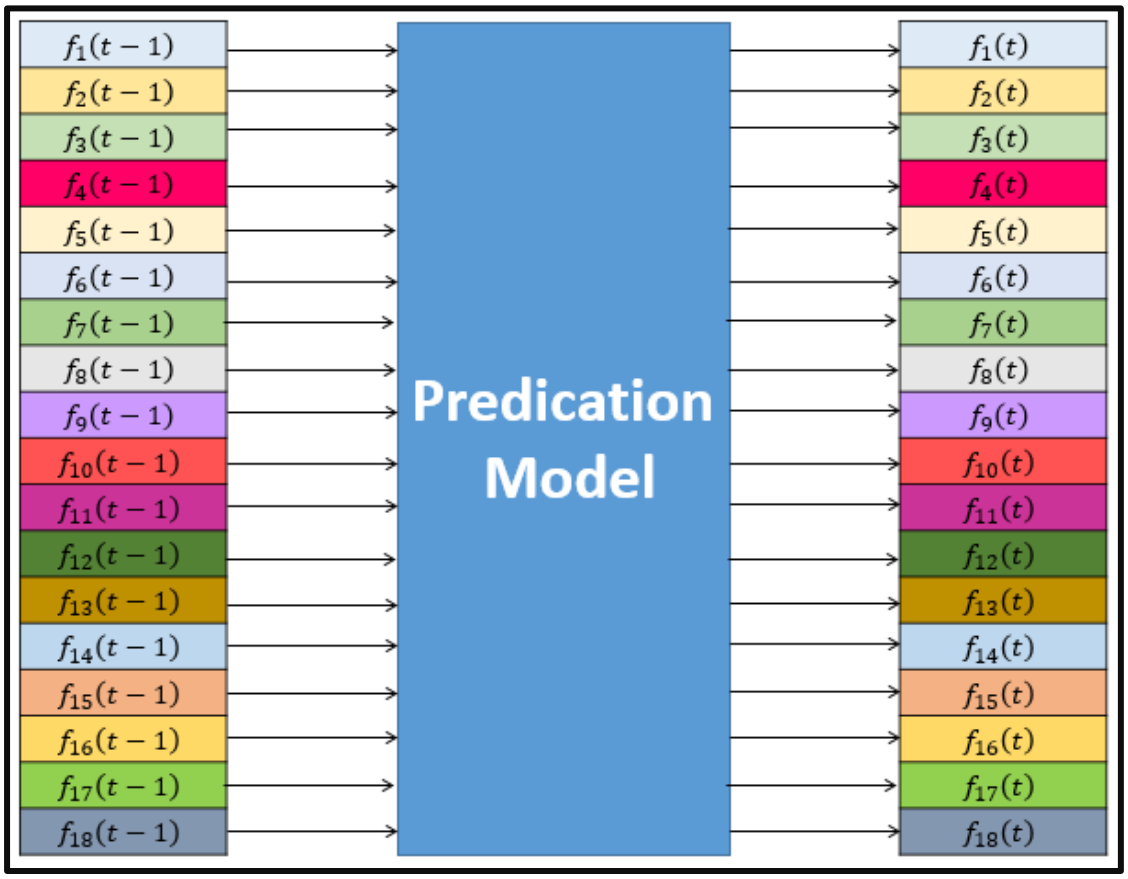


Figure (3.3) Timesteps for Data.

3.2.5 Splitting Seismic Dataset Stage

After selecting 18 features from the input data set using correlation matrix as illustrated in the previous stage, this stage indicated splitting data set into 80% training and 20% testing for machine learning algorithms and splitting the dataset into 80% training and 20% testing and validation for deep learning algorithms.

3.2.6 Prediction Stage

The proposed model in the prediction stage is divided into two branches, which are

1) ML machine learning using SVR & KNR algorithms.

2) Deep learning using LSTM and a proposed hybrid LSTM & GUR algorithm.

3.2.6.1 Prediction Based on Machine Learning

The seismic data is a continuous data type that means a sequence of observations taken sequentially in time, so this work takes models that deal with continuous data to improve prediction. The most common and effective models with continuous data are Support Vector Regression (SVR) and K Neighbor Regression (KNR) models. The details for each model are described as follows:

i. **SVR Model**

SVR allows quantifying the acceptable error in the proposed model and will find a suitable line or super planar with higher dimensions to fit the seismic dataset. The most important SVR parameter is the Kernel type. It can be -Linear - Polynomial - Gaussian SVR. The SVR has to obtain multi-dimensional input and a single output as shown in Figure (3.4).

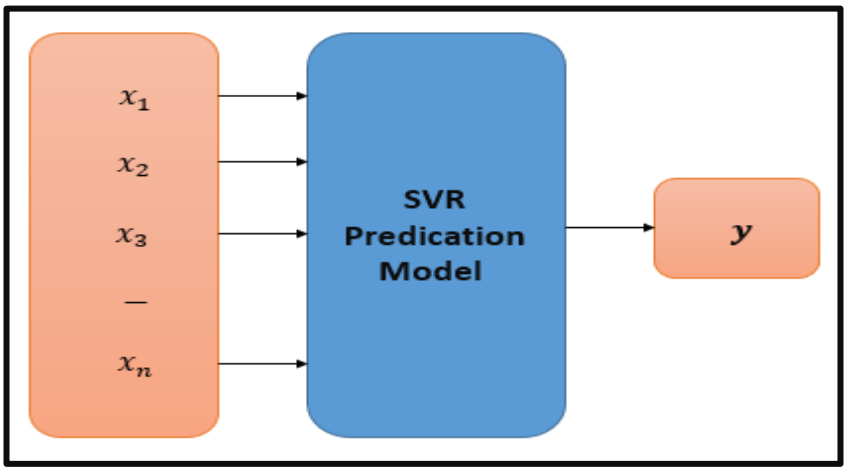


Figure (3.4): General Structure of the SVR.

Therefore, the proposed model has to pick one specific feature to predict using all the features including the one it intended to predict. The details of prediction porosity resulting from the seismic data method of oil reservoirs using SVR .

ii. KNR Model

The KNR algorithm uses to predict the values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set. The details of predication porosity resulting from the seismic data method of oil reservoirs using KNR .

3.2.6.2 Prediction Based on Deep Learning

Prediction time series is the most difficult type of predictive modeling because it adds more complexity through the dependence of the sequence on the input variables. A powerful type of neural network designed to handle sequence dependencies is called an ANN. GRU and LSTM are used in deep learning because they are very large structures that can be successfully trained. The details of the proposed LSTM and GRU are described as follows:

i. LSTM Prediction Model

LSTM can hold information for long periods due to its chain-like structure, where it can solve tasks that are difficult to implement using traditional RNN. LSTM neural networks are structured for sequential data processing. Network status at any time depends both on the present and preceding input of the network. The type of our model architecture is many-to-many. LSTM consists of three main gates:

- Forget gate: there is information that is no longer needed to complete the task, this gate removes it and this improves the performance of the network.
- Input gate: through this portal, information is added to memory cells.

- Output gate: this portal produces the necessary information in hidden layer output.

The LSTM cell is created from the input layer, the previously hidden cell h_{t-1} is entered and the new sequence x_t is entered, where the first step of this combined entry is that it is crushed through the *tanh* layer where *tanh* takes large or small variable numbers and converts them at a specific rate between (-1,1) to generate candidate memory cells $\tilde{C}_t$ according to Equation (2.16).

After completing the above parts, the cell state C_t of LSTM is updated according to Equation (2.14). This equation connects the pre-state C_{t-1} and the present temporary-state c . Through the output gate, LSTM outputs the specified state, based on the cell status, where runs a sigmoid layer to determine the unit state section to be exported according to Equation (2.15). In addition, deals with current output o_t and state C_t with a *tanh* layer to write a new hidden layer state h_t according to Equation (2.15).

In this model, the input is 18 features, two hidden layers each layer consisting of 1235 nodes of LSTM, and 18 outputs, Connected in fully connected layers form as indicated in Figure (3.5).

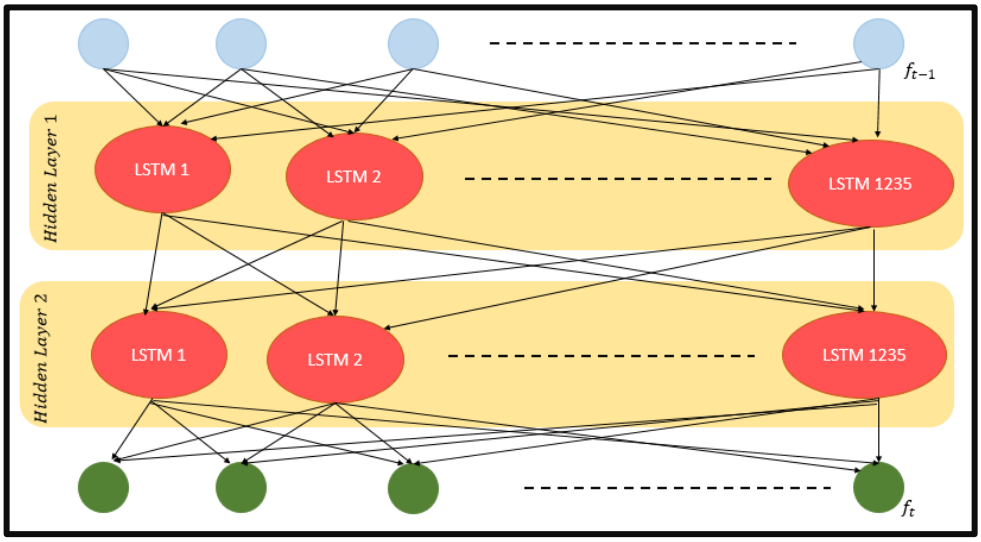


Figure (3.5): Prediction LSTM Model.

ii. A Proposed LSTM, GRU, and ANN Prediction Model

The proposed LSTM, GRU, ANN prediction model aims to improve the results of prediction by using the output of the LSTM and GRU model. In section (i) the LSTM models were presented, while this section will be a hybrid LSTM, GRU, and ANN algorithms.

The proposed model included three hidden layers, the first layer includes 1235 LSTM nodes, the second layer has 1235 GRU nodes and the third layer has 1235 ANN nodes, the output of the first layer represents the input for the second layer and the output of the second layer represents the input for the third layer was proposed as shown in Figure (3.6).

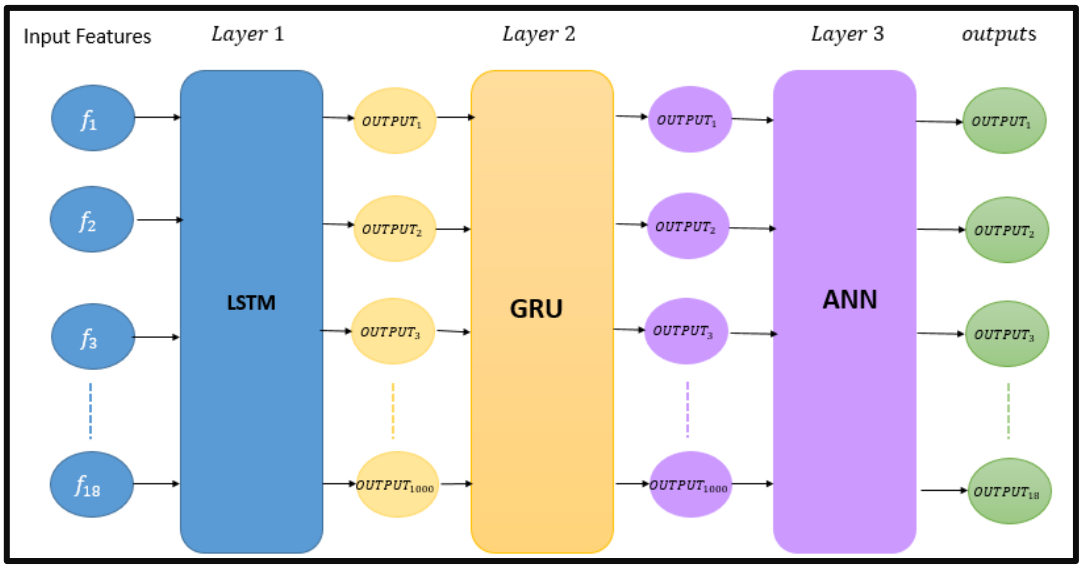


Figure (3.6): A Proposed Hybrid LSTM, GRU, and ANN Regression Model.

As shown in Figure (3.6), the input is 18 features, three hidden layers, the first hidden layer consisting of 1235 nodes of LSTM and the second hidden layer consisting of 1235 nodes of GRU and 1000 outputs as well as the layer consisting of 1235 nodes of GRU and the third" hidden layer consisting of 1235 nodes of ANN and 18 outputs, connected in fully connected layers form.

The details of the proposed Hybrid LSTM, GRU, and ANN Model are illustrated in the algorithm (3.1) and algorithm (3.2).

Algorithm (3.1) Hybrid LSTM, GRU, Prediction
Input: To The LSTM cell **X** (current input) $\mathbf{H_{t-1}}$ (previous hidden state) **Output:** From The GRU cell $\mathbf{H_t}$ (current hidden state)

Begin

Training phase

step 1: Training = {(x_t, x_t+1), t = 1, 2, · · ·, T1} and validation set X

validation = {(x_t, x_t + 1), t = T1 + 1, T1 + 2, · · ·, T2},

step 2: For epoch = 1 to 100 − epoch do

Perform forward propagation recurrently using the Equation from (2.15) to (2.19).

Step 3: GRU cell takes the previous hidden state Ht-1 and generates the candidate hidden state $\widehat{H}_t$ using equation (2.13).

Later passed this entire information to the *tanh* function, the resultant value is the candidate's hidden state.

Step 4: Update hidden state to H_t using the following Equation(2.19) If f value u_t is 0 then new hidden H_t state depends on the candidate state $\widehat{H}_t$ if the f value is 1 then the new hidden H_t the state depends on the previous hidden state H_{t-1} .

step 5: The output of each GRU state H_t is fed to the next one.

Step 6: Compute output error: xet+1 − xt+1, t = 1, 2, · · ·, T1

Step 7: Perform backward propagation

Testing phase

Step 8: Perform forward propagation recurrently to update the network states.

Step 9: Perform forward propagation recurrently to compute Xe = {xet+1, t = T1 + 1, T1 + 2, · · ·, T2}

Step **10**: Save the current Θ.

if epoch ≥ min − epoch then

The LSTM used in the experiment ran for 300 epochs to train to the optimal results.70 batch size was used for the LSTM break.

End Algorithm

Algorithm (3.2) Hybrid GRU, ANN Regression Prediction porosity

Input: features from previous layers

Output: N features for time

Begin

Step 1: Initialize weights, set to small random values, not zeros.

Step 2: Forward- propagation: - each input unit (x_i ,i=1,…,n) receives input signal x_i and broadcasts it to all units in the first hidden layer above (the hidden units).

Step (2.1): Calculation the value of each hidden unit (Z_j, j=1,...,n), then apply active function(tanh) to find the output signal and send this signal to all units in the above-hidden layer.

Step (2.2): Repeat step 2.1 on all hidden layers, the output signal from the last hidden layer sends to all units in the above layer (output units).

Step (2.3): Calculate value each output unit (y_k, k=1,…,n) and apply active function(softmax) to compute the output signals.

Step3: Back-propagation of error: each output unit (y_k, k=1,…,n) receives a desired class corresponding to the input training class, and computes its error.

Step (3.1): Calculate the gradient j(n) for neuron j is an output node and for neuron, j is a hidden node.

Step(3.2): Compute the weight changes $\Delta w_{ji}(n)$ for the weight

connecting neuron i to neuron j.

Step (3.3): Update the weights of the network in layer L.

Step (3.4): Update the bias of the network in layer L.

Step4: Increase iteration and repeat the process until the weights are stable.

End

Chapter Four

Experimental Results and Evaluation

Chapter Four

Experimental Results and Evaluation

4.1 Introduction

This chapter describes the results of prediction models, and the proposed model will be the one that provides the best results among the rest of the results. The following sections describe experimental results from the pre-processing and forecasting phases, which include training and test.

4.2 Implementation Environment

The proposed experiments are performed under a specific system requirement such as Windows-10 operating system, Hardware processor: Core i5- CPU 3320M, 2.60 GHz, and (8GB) RAM. The code was written in python to analyze and learn data then predict

4.3 Results of the Proposed Model

The proposed model aims to improve the quality of prediction by decreasing the error rate in locating oil wells that contain high-quality oil. To achieve this goal, the proposed model followed several stages and it applied on a seismic dataset that has been obtained and collected, in each stage, it uses technique or algorithm to reach the predictive stage using machine learning and deep learning. The results of each step will be presented and discussed in this section.

4.3.1 Clarify Load Dataset

The seismic dataset includes 22 chronological-dependent features on which the models will be trained and tested, these features are (AE, AF, AP, AWCP, AWP, CIP, D, DF, DIA, F15-30, F25-40, F45- 60, F50-60, F55-70, I, IAA, IF, IP, QT, SD, SDIA) Table (4.1) shows a sample of the original values for the input dataset.

Table 4.1: Original Seismic Dataset [5 rows × 22 columns].

Time	A-E	AF	AP	AWCP	AWP	CIP	D	DF	DIA	F15-36	----	SDIA
1888.420044	0.000535	28.055161	0.000535	0.000525	0.005809	0.982095	0.000133	22.867523	0.000022	0.000322	----	-4.190000e-08
1888.520020	0.000535	28.055161	0.000535	0.000525	0.005809	0.982095	0.000133	22.867523	0.000022	0.000322	----	-4.190000e-08
1888.619995	0.000535	28.055161	0.000535	0.000525	0.005809	0.982095	0.000133	22.867523	0.000022	0.000322	----	-4.190000e-08
1888.720093	0.000535	28.055161	0.000535	0.000525	0.005809	0.982095	0.000133	22.867523	0.000022	0.000322	----	-4.190000e-08
1888.820068	0.000535	28.055161	0.000535	0.000525	0.005809	0.982095	0.000133	22.867523	0.000022	0.000322	----	-4.190000e-08

4.3.2 Results of the Preprocessing Dataset

The preprocessing dataset represents statistical operations to preprocess entered data and address the problems it suffers, to improve the performance of the proposed model.

The first step in preprocessing is to check data type. The Check data type is process takes input row in each column and check if the data type is float 64. If the data type is not floating 64, remove the data from the features because it is considered irrelevant data like special symbols and characters. Check type of the dataset is shown in Figure (4.1).

```
In [4]: rawData.dtypes

Out[4]: A-E       float64
        AF        float64
        AP        float64
        AWCP      float64
        AWP       float64
        CIP       float64
        D         float64
        DF        float64
        DIA       float64
        F15-36    float64
        F25-40    float64
        F35-50    float64
        F45-60    float64
        F50-65    float64
        F55-70    float64
        I         float64
        IAA       float64
        IF        float64
        IP        float64
        QT        float64
        SD        float64
        SDIA      float64
        dtype: object
```

Figure 4.1: Check Data Type in the Preprocessing Seismic Dataset.

After checking data type, the check missing data is done as illustrated in Figure (4.2). To determine any null values, each row was examined with data versus each column. In the case of null values, the

Mean value is replaced for all other values of a similar time in the column's data set.

```
<class 'pandas.core.frame.DataFrame'>
Float64Index: 1234 entries, 1888.420044 to 2011.720093
Data columns (total 22 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   A-E     1234 non-null   float64
 1   AF      1234 non-null   float64
 2   AP      1234 non-null   float64
 3   AWCP    1234 non-null   float64
 4   AWP     1234 non-null   float64
 5   CIP     1234 non-null   float64
 6   D       1234 non-null   float64
 7   DF      1234 non-null   float64
 8   DIA     1234 non-null   float64
 9   F15-36  1234 non-null   float64
 10  F25-40  1234 non-null   float64
 11  F35-50  1234 non-null   float64
 12  F45-60  1234 non-null   float64
 13  F50-65  1234 non-null   float64
 14  F55-70  1234 non-null   float64
 15  I       1234 non-null   float64
 16  IAA     1234 non-null   float64
 17  IF      1234 non-null   float64
 18  IP      1234 non-null   float64
 19  QT      1234 non-null   float64
 20  SD      1234 non-null   float64
 21  SDIA    1234 non-null   float64
dtypes: float64(22)
memory usage: 221.7 KB
```

Figure 4.2: Check Missing Data in the Preprocessing Seismic Dataset.

Figure (4.3) presents the sum of the null values for each column, where it shows that the columns contain zero of the null values, this denotes that there are no missing values in the dataset.

```
Out[6]: A-E       0
        AF        0
        AP        0
        AWCP      0
        AWP       0
        CIP       0
        D         0
        DF        0
        DIA       0
        F15-36    0
        F25-40    0
        F35-50    0
        F45-60    0
        F50-65    0
        F55-70    0
        I         0
        IAA       0
        IF        0
        IP        0
        QT        0
        SD        0
        SDIA      0
        dtype: int64
```

Figure 4.3: The Dataset After using Missing Data in the Preprocessing Seismic Dataset.

4.3.3 Results of the Feature Selection using Correlation Matrix Method

Feature selection is the process used to select the most important input features to improve the accuracy prediction model and reduce computational cost. The correlation matrix method is an efficient and faster method used in the feature selection process. Figure (4.4) shows the correlation matrix, the vertical and horizontal lines indicated in 22 attributes in the data set, all the diagonal elements of the correlation matrix must be a value of 1, and the diagonal elements of the correlation matrix are divided into two parts (upper and lower triangle) with the same values which lie in the ranges between [1 to -1].

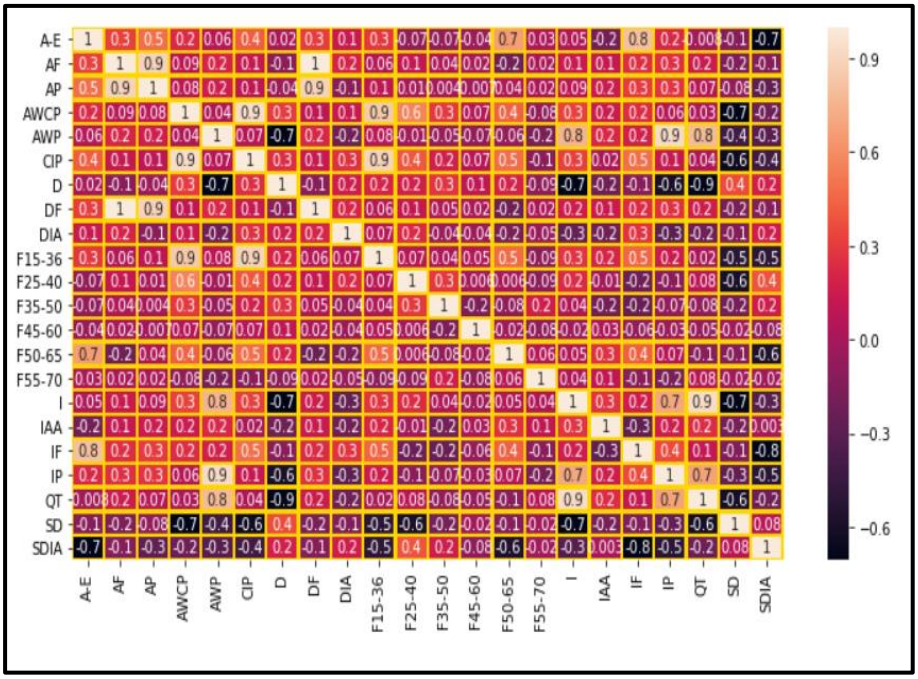


Figure (4.4): The Correlation Matrix for Features in Dataset.

As illustrated in Figure (4.4), the dark colors represent the weak relationship between features such as A-E with QT= -0.0008, A-E and SD = -0.1 , and A-E with SDIA= -0.7.A light colors represent the strong relationship between features such as: AF with DF=1 ,AWP with IP =0.9, I with QT =0.9, AWCP-CIP=0.9,A-F with F50-65 =0.7,and A-F with IF =0.8 and so on.

Based on the above correlation matrix, if two features are highly correlated, only one of them will be kept. Features that correlate with another feature by 'threshold =0.9' or above will be removed. Therefore, will remove four features which are:['CIP', 'DF', 'IP', 'QT'] and will keep 18.

4.3.4 Results of the Feature Selection Using Reshape Dataset Method

Reshape the data so that the prediction of t time steps is calculated based on the given number of past and present time step features. In this step, the conversion from 2D dataset [samples, features] to 3D models, time steps, features]. Table (4.2) shows the results of Reshape technique. Train input Shape:(986, 1, 18) and Train Y shape:(986, 18).Test X shape: (247, 1, 18) Test y Shape:(247,18).

Table 4.2: Seismic Dataset After Applied the Reshape Technique.

Time Step	1	2	3	4	5	6	7	8	9	10
var1(t-1)	0.457713	0.457713	0.457713	0.457713	0.457713	0.457713	0.457713	0.457713	0.457713	0.457713
var2(t-1)	1	1	1	1	1	1	1	1	1	1
var3(t-1)	0.765156	0.765156	0.765156	0.765156	0.765156	0.765156	0.765156	0.765156	0.765156	0.765156
var4(t-1)	0.781791	0.781791	0.781791	0.781791	0.781791	0.781791	0.781791	0.781791	0.781791	0.781791

var15(t-1)	var14(t-1)	var13(t-1)	var12(t-1)	var11(t-1)	var10(t-1)	var9(t-1)	var8(t-1)	var7(t-1)	var6(t-1)	var5(t-1)
0.534517	0.769546	0.255994	0.071566	1	0.330351	0.903048	0.790401	0.99912	0.586478	0.514686
0.534517	0.769546	0.255994	0.071566	1	0.330351	0.903048	0.790401	0.99912	0.586478	0.514686
0.534517	0.769546	0.255994	0.071566	1	0.330351	0.903048	0.790401	0.99912	0.586478	0.514686
0.534517	0.769546	0.255994	0.071566	1	0.330351	0.903048	0.790401	0.99912	0.586478	0.514686
0.534517	0.769546	0.255994	0.071566	1	0.330351	0.903048	0.790401	0.99912	0.586478	0.514686
0.534517	0.769546	0.255994	0.071566	1	0.330351	0.903048	0.790401	0.99912	0.586478	0.514686
0.534517	0.769546	0.255994	0.071566	1	0.330351	0.903048	0.790401	0.99912	0.586478	0.514686
0.534517	0.769546	0.255994	0.071566	1	0.330351	0.903048	0.790401	0.99912	0.586478	0.514686
0.534517	0.769546	0.255994	0.071566	1	0.330351	0.903048	0.790401	0.99912	0.586478	0.514686
0.534517	0.769546	0.255994	0.071566	1	0.330351	0.903048	0.790401	0.99912	0.586478	0.514686

var8(t)	var7(t)	var6(t)	var5(t)	var4(t)	var3(t)	var2(t)	var1(t)	var18(t-1)	var17(t-1)	var16(t-1)
0.790401	0.99912	0.586478	0.514686	0.781791	0.765156	1	0.457713	0.259522	0.401802	1
0.790401	0.99912	0.586478	0.514686	0.781791	0.765156	1	0.457713	0.259522	0.401802	1
0.790401	0.99912	0.586478	0.514686	0.781791	0.765156	1	0.457713	0.259522	0.401802	1
0.790401	0.99912	0.586478	0.514686	0.781791	0.765156	1	0.457713	0.259522	0.401802	1
0.790401	0.99912	0.586478	0.514686	0.781791	0.765156	1	0.457713	0.259522	0.401802	1
0.790401	0.99912	0.586478	0.514686	0.781791	0.765156	1	0.457713	0.259522	0.401802	1
0.790401	0.99912	0.586478	0.514686	0.781791	0.765156	1	0.457713	0.259522	0.401802	1
0.790401	0.99912	0.586478	0.514686	0.781791	0.765156	1	0.457713	0.259522	0.401802	1
0.790401	0.99912	0.586478	0.514686	0.781791	0.765156	1	0.457713	0.259522	0.401802	1
0.790401	0.99912	0.586478	0.514686	0.781791	0.765156	1	0.457713	0.259522	0.401802	1

var9(t)	var10(t)	var11(t)	var12(t)	var13(t)	var14(t)	var15(t)	var16(t)	var17(t)	var18(t)
0.903048	0.330351	1	0.071566	0.255994	0.769546	0.534517	1	0.401802	0.259522
0.903048	0.330351	1	0.071566	0.255994	0.769546	0.534517	1	0.401802	0.259522
0.903048	0.330351	1	0.071566	0.255994	0.769546	0.534517	1	0.401802	0.259522
0.903048	0.330351	1	0.071566	0.255994	0.769546	0.534517	1	0.401802	0.259522
0.903048	0.330351	1	0.071566	0.255994	0.769546	0.534517	1	0.401802	0.259522
0.903048	0.330351	1	0.071566	0.255994	0.769546	0.534517	1	0.401802	0.259522
0.903048	0.330351	1	0.071566	0.255994	0.769546	0.534517	1	0.401802	0.259522
0.903048	0.330351	1	0.071566	0.255994	0.769546	0.534517	1	0.401802	0.259522
0.903048	0.330351	1	0.071566	0.255994	0.769546	0.534517	1	0.401802	0.259522
0.903048	0.330351	1	0.071566	0.255994	0.769546	0.534517	1	0.401802	0.259522

4.4 Results of the Prediction Model

The proposed prediction model includes two branches: Machine Learning branch based on (SVR and KNR) algorithms; Deep learning

branch based on LSTM and a proposed hybrid LSTM, GUR, and ANN algorithm. Results from each branch are presented in this section. In addition, this section reviews the compared results between the two branches to find the best performance.

4.4.1 Results of the Machine Learning Prediction Model

This section introduces the implementation and evaluation of the Support Vector Regression (SVR) and K Neighbor Regression (KNR). To measure the performance efficiency of the algorithms used in this section. will rely on the metrics: absolute error (MAE) using an Equation (2.20) and root mean square error (RMSE) using an Equation (2.21). And can explain the results of the measurements that the lower the error rate, the more accurate the model used.

4.4.1.1 Results of the SVR Prediction Model

The input to the SVR prediction model is 18 features to apply prediction for each feature .

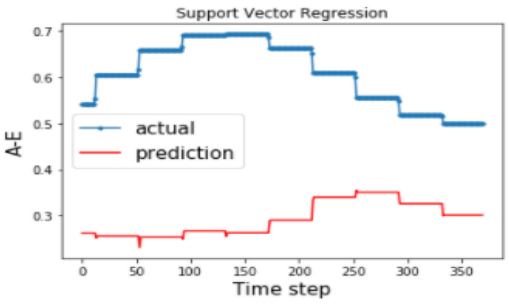
Table (4.3) presents the results of error metrics (MAE)and (RMSE) for the four cases of partition seismic dataset:60%Training and 40% Testing, 70%Training and 30% Testing, 80%Training and 20% Testing, and 90%Training and 10% Testing.

Table (4.3): Results of MAE and RMSE of the Prediction in SVR Model

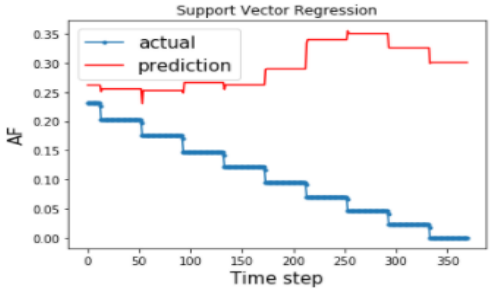
Dataset Splitting Ratios	MAE	RMSE
60% Training &40 % Testing	**0.289**	**0.334**
70% Training &30 % Testing	**0.297**	**0.341**
80% Training &20 % Testing	**0.264**	**0.300**
90% Training &10 % Testing	**0.238**	**0.255**

As shown in the table above the best performance of the SVR with a case of splitting dataset 90% Training and 10% testing, where the MAE=0.238 and RMSE =0.255. Figure (4.6) illustrated the accuracy (prediction and actual) of prediction for all features using the SVR model.

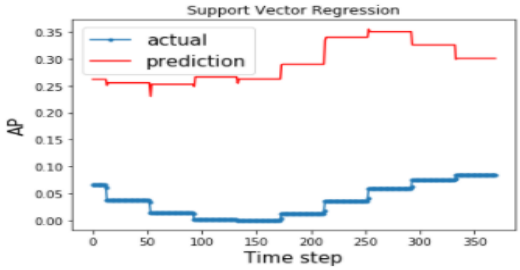
Predict the A-E Feature



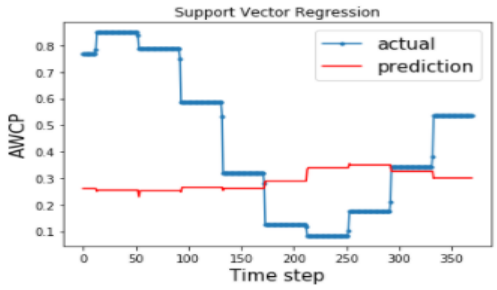
Predict the AF Feature



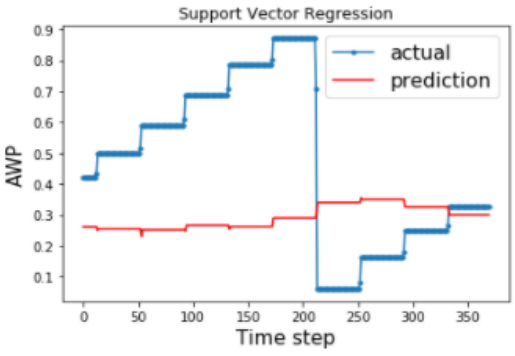
Predict the AP Feature



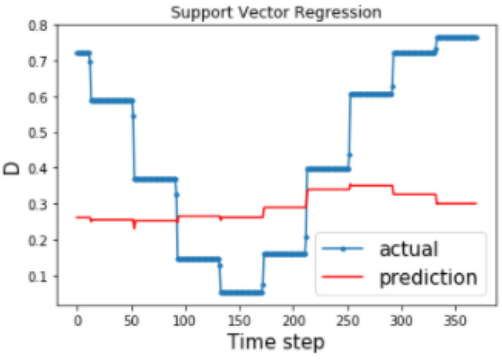
Predict the AWCP Feature



Predict the AWP Feature

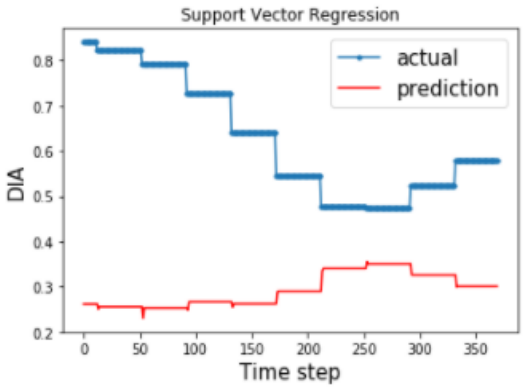


Predict the D Feature

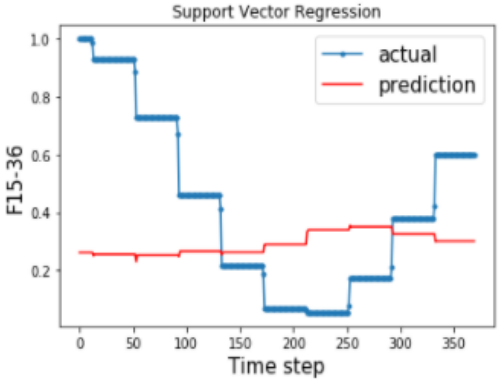


Predict the DIA Feature

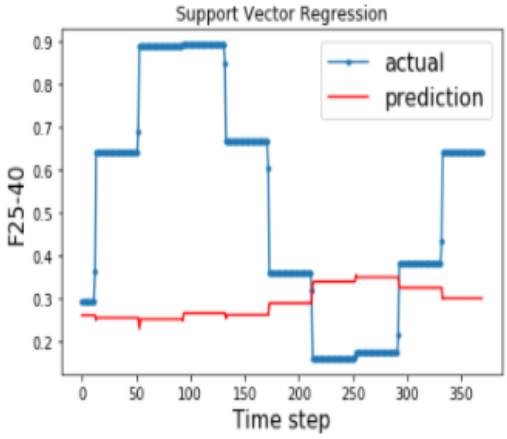
Predict the F15-36 Feature



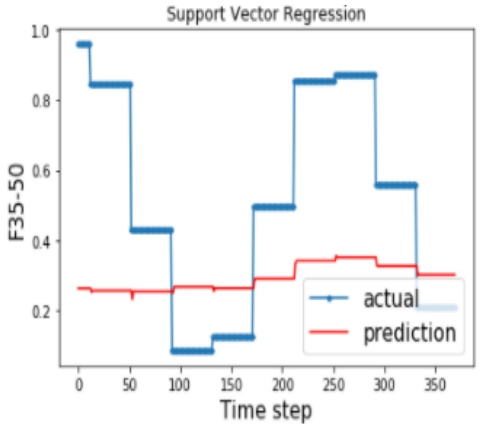
Predict the F25-40 Feature



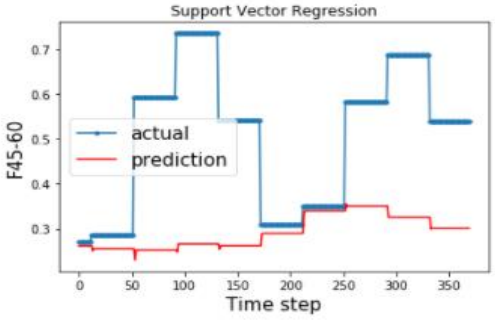
Predict the F35-50 Feature



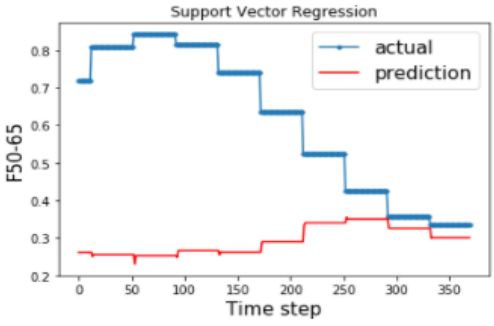
Predict the F45-60Feature



Predict the F50-65 Feature



Predict the F55-70 Feature



Predict the I Feature

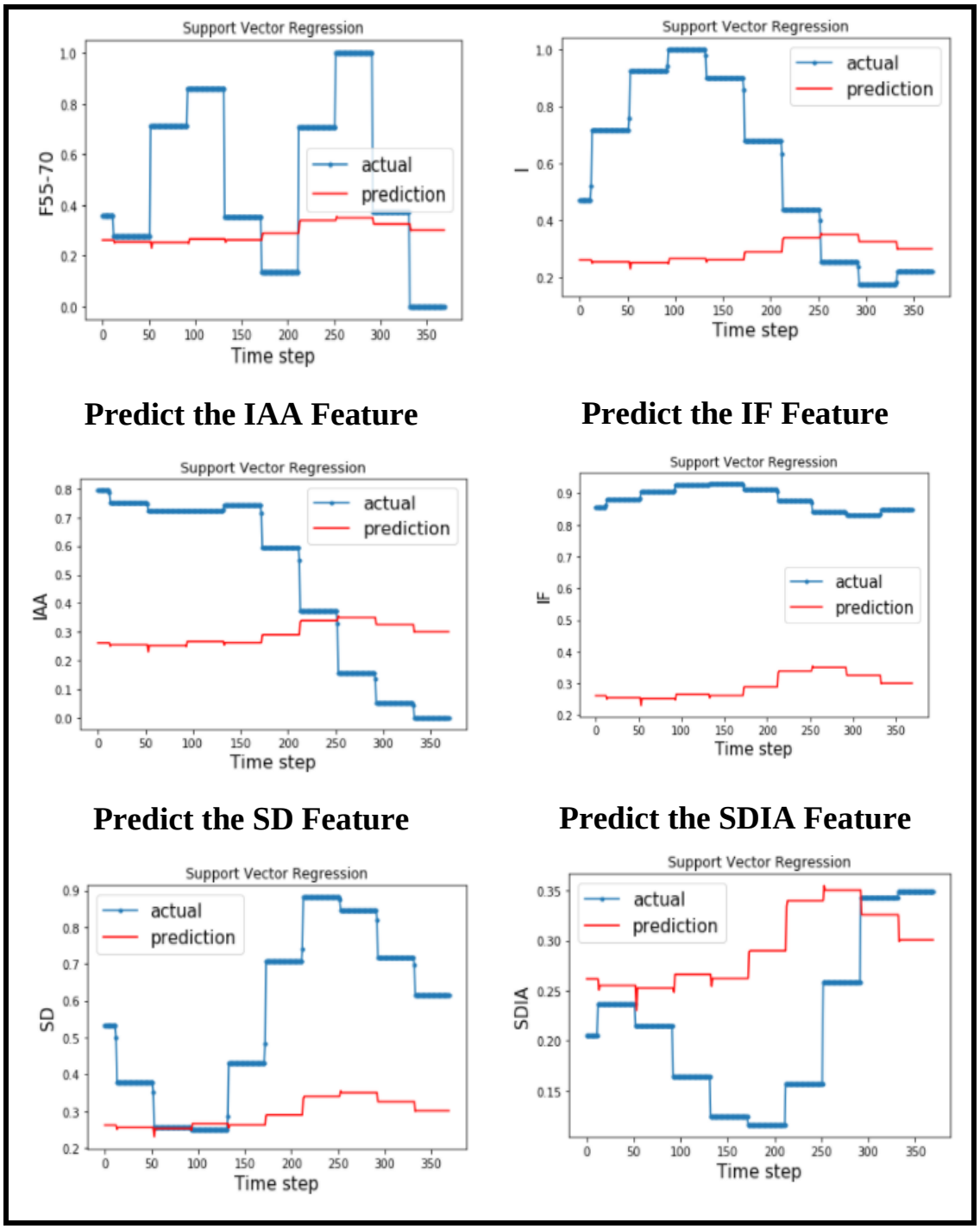


Figure (4.5): Accuracy of Predicted Results in the SVR Model.

4.4.1.2 Results of the KNR Prediction Model

The KNR algorithm uses to predict the values of any new data points. This means that the new point is assigned a value based on how

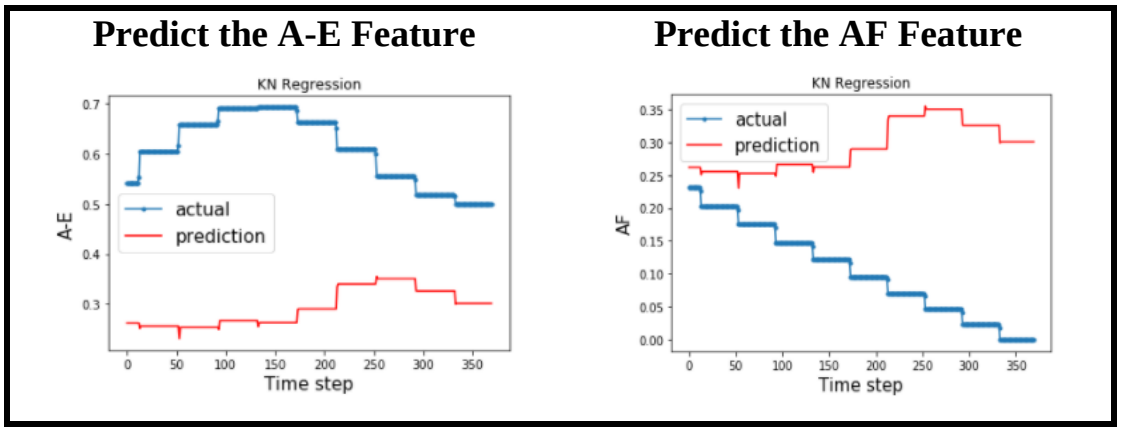
closely it resembles the points in the training set. The input to the KNR prediction model is 18 features to apply prediction for each feature .

Table (4.4) shows the results of error metrics (MAE) and (RMSE) for the four cases of partition seismic dataset:60% Training and 40% Testing, 70%Training and 30% Testing, 80%Training and 20% Testing, and 90% Training and 10% Testing.

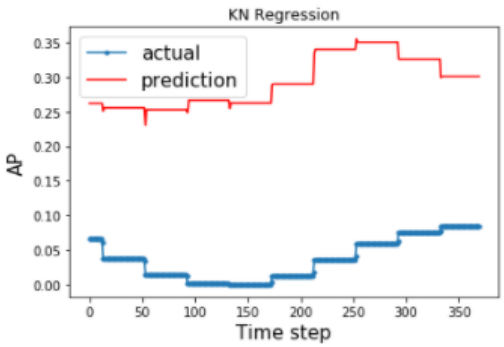
Table (4.4): Results of MAE and RMSE of the Prediction in KNR Model

Dataset Splitting Ratios	MAE	RMSE
60% Training &40 % Testing	0.332	0.392
70% Training &30 % Testing	0.332	0.386
80% Training &20 % Testing	0.296	0.344
90% Training &10 % Testing	0.276	0.301

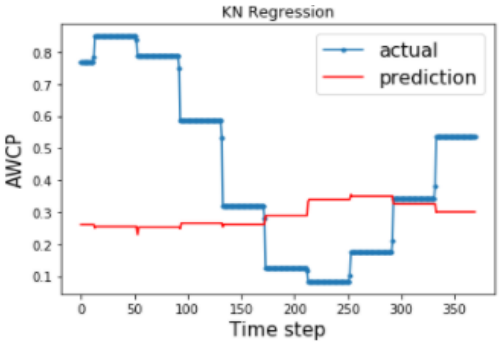
As shown in the table above the best performance only of the KNR with a case of splitting dataset 90% Training and 10% Testing, where the MAE=0.276 and RMSE =0.301. Figure (4.6) is illustrated the accuracy (prediction and actual) of prediction for all features using the KNR model.



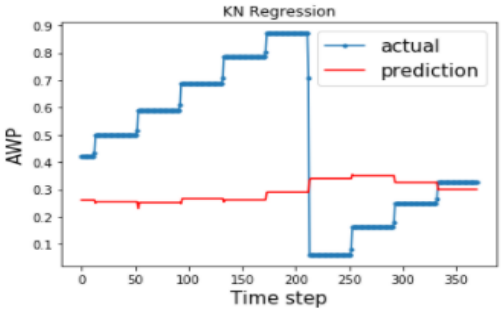
Predict the AP Feature



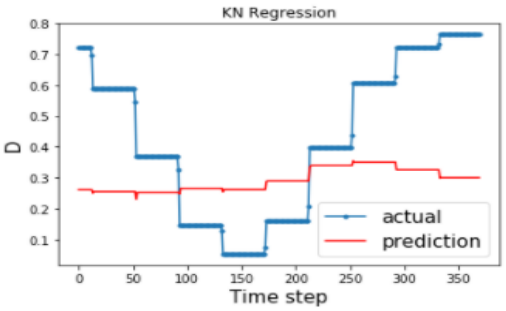
Predict the AWCP Feature



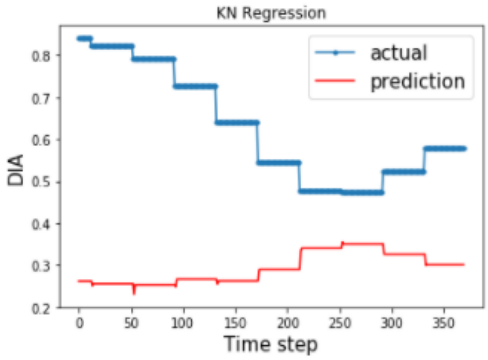
Predict the AWP Feature



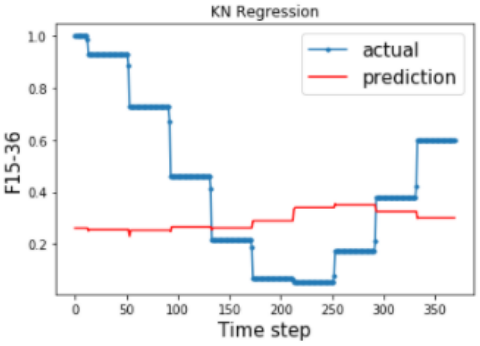
Predict the D Feature



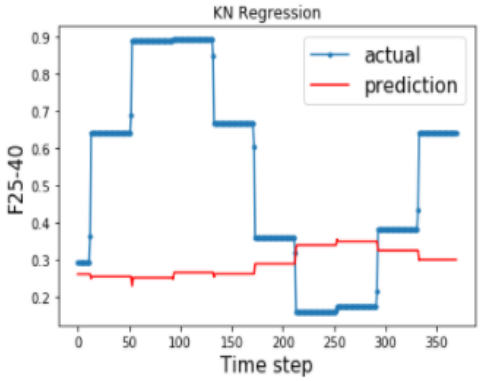
Predict the DIA Feature



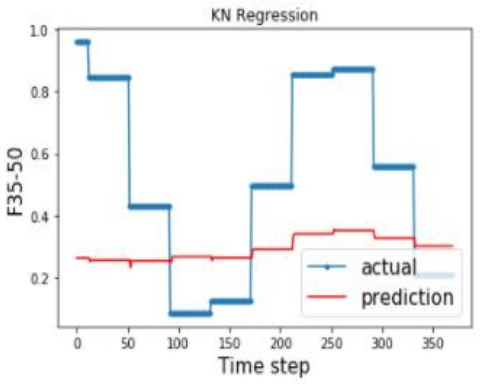
Predict the F15-36 Feature



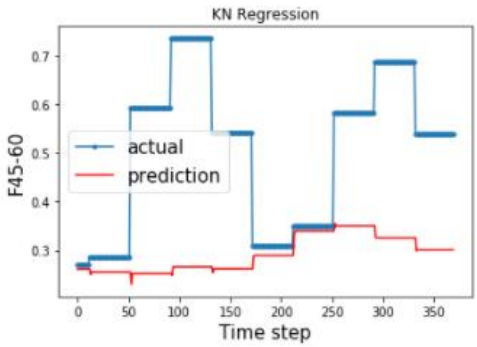
Predict the F25-40 Feature



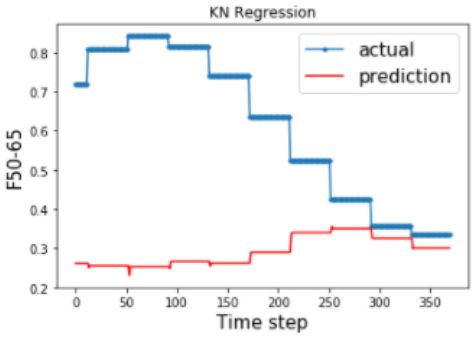
Predict the F35-50 Feature



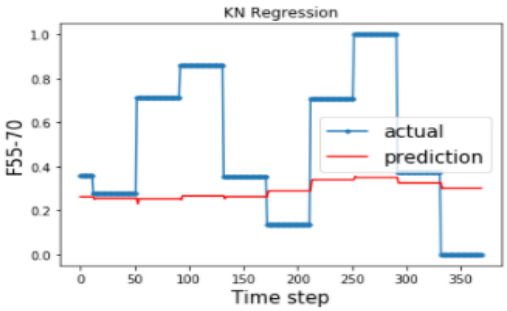
Predict the F45-60Feature



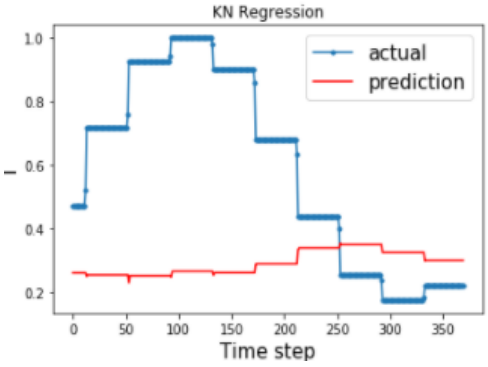
Predict the F50-65 Feature



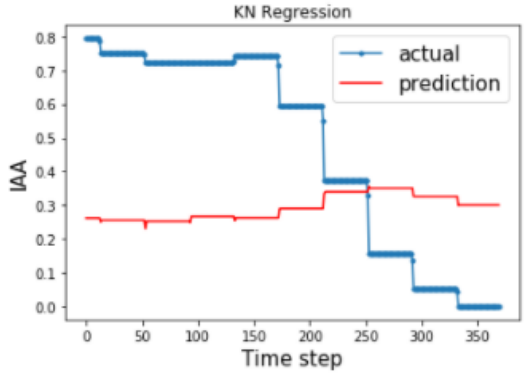
Predict the F55-70 Feature



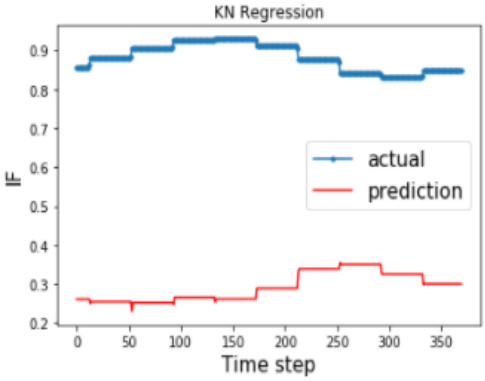
Predict the I Feature



Predict the IAA Feature



Predict the IF Feature



Predict the SD Feature

Predict the SDIA Feature

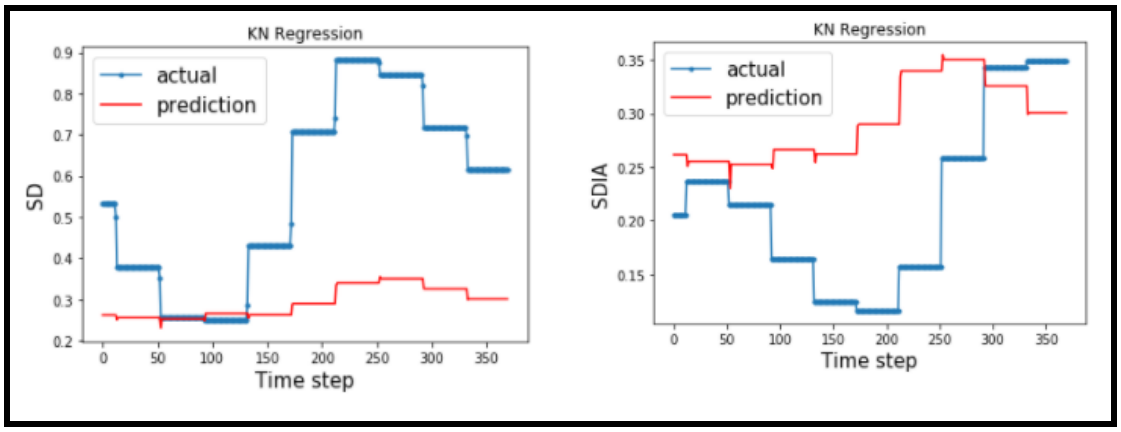


Figure (4.6): Accuracy of Predicted Results in the KNR Model.

When the performance comparison between SVR and KNR is based on the results of the error measures (MAE and RMSE) as shown in Tables (4.3 and 4.4), it is found that SVR has the best over the results from the KNR prediction models.

4.4.2 Results of the Deep Learning Prediction Model

This section will display the results obtained from the implementation of the prediction models (LSTM) and Hybrid (LSTM, GRU, and ANN).

Seismic dataset is divided into two sets: Training Set: It is the set of data that is used to train and make the model learn the hidden features/patterns in the data. Testing Set: The test set is a separate set of data used to test the model after completing the training. It provides an unbiased final model performance metric in terms of accuracy metrics mean absolute error (MAE) using Equation (2.20) and root means square error (RMSE) using Equation (2.21). The lower the error ratio represents the greater the accuracy of the model.

4.4.2.1 Results of the LSTM Prediction Model

The input dataset to the LSTM prediction model is reshaping data in 3d form [Samples, Time steps, Features]. As long as the model is trained to take one millisecond of inputs to predict the porosity resulting from seismic data of oil reservoirs for the millisecond, the time step equal to one, it will be the form of the input for training and testing data as [samples, 1, features].

The LSTM model includes two hidden layers, each of which comprises 1000 LSTM nodes. LSTM utilized 100 epochs for training and a batch size of 64 for the training dataset to avoid over fitting.

LSTM model has two layers that are fully connected, the first layer is a hidden layer one which has 1000 nodes of LSTM. The inputs are 3D format : [18,1,18] but the output in 2D format: [18,1000] .The second layer is a dense layer which is the last layer from which the output was obtained, and has 18 nodes. The input into the dense layer is in 2D format [18,1000] because it's coming from the output of the first layer and the output is in 2D format [18,18].

The summary of the LSTM model and the total number of parameters in the network and the number of parameters that have been trained are shown in Table (4.5).

Table 4.5: Results of MAE and RMSE of the Prediction in LSTM Model

Dataset Splitting Ratios	MAE	RMSE
60% Training &40 % Testing	**0.11**	**0.13**
70% Training &30 % Testing	**0.145**	**0.158**
80% Training &20 % Testing	**0.023**	**0.029**
90% Training &10 % Testing	**0.038**	**0.043**

As shown in the table above, the performance of the predictive model was studied and its performance was evaluated based on the results of error measures in four cases of database partitioning. The study shows the best performance of the LSTM model with 80% Training & 20% Testing case, where the MAE= 0.023 and RMSE= 0.029.While in case 90% Training &10% Testing the MAE = 0.038 and RMSE = 0.043 ,in case of the 60% Training &40% Testing the MAE= 0.11 and RMSE= 0.13.The Lowest performance of the LSTM with 70% Training and 30% Testing, where MAE= 0.145 and RMSE= 0.158.

Figure (4.10) shows the loss of the LSTM model, the loss for training is approximately =0.25 and for testing is approximately = 0.07 in 100 epochs.

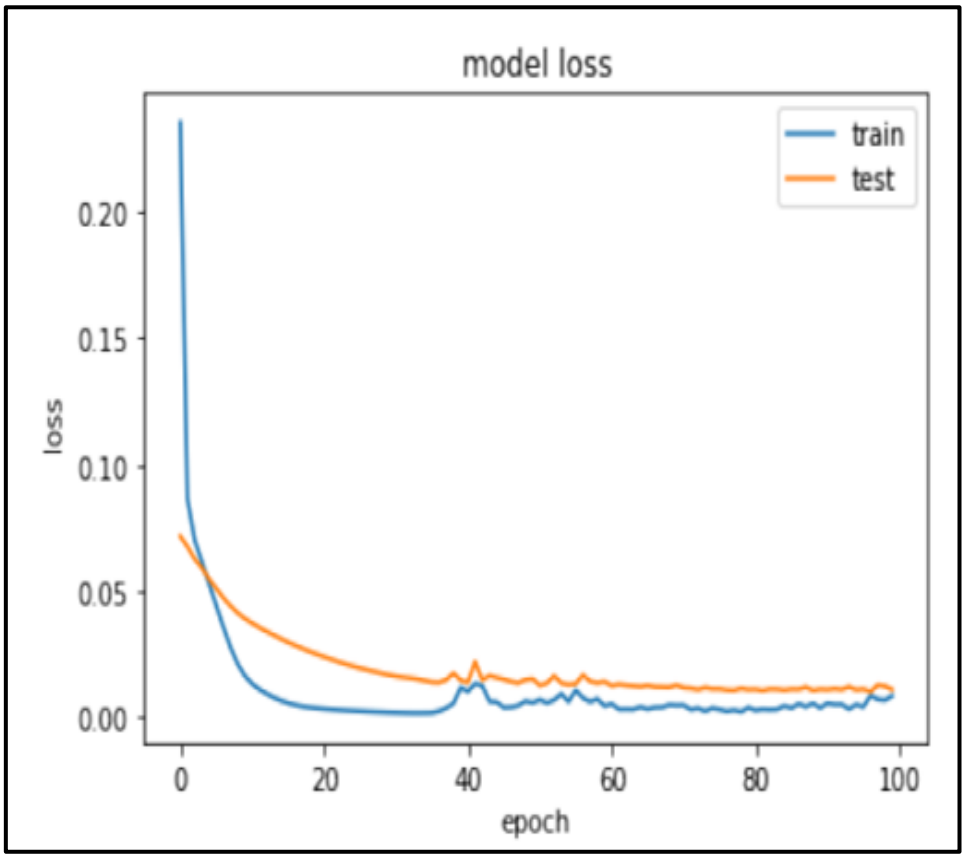
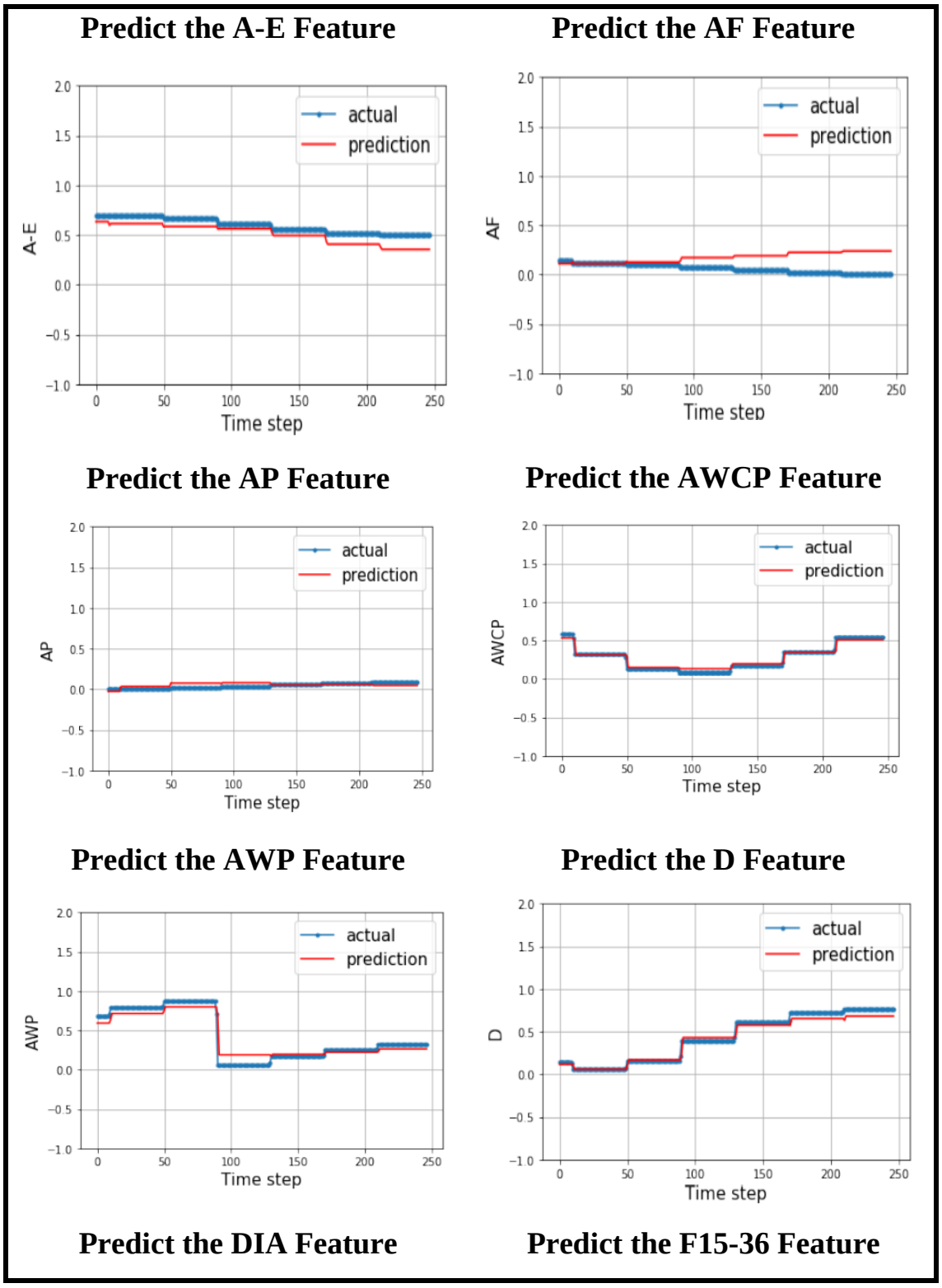
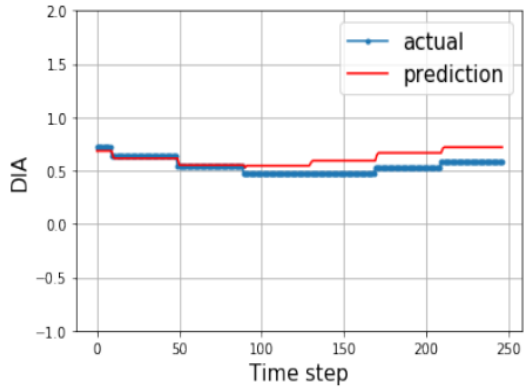
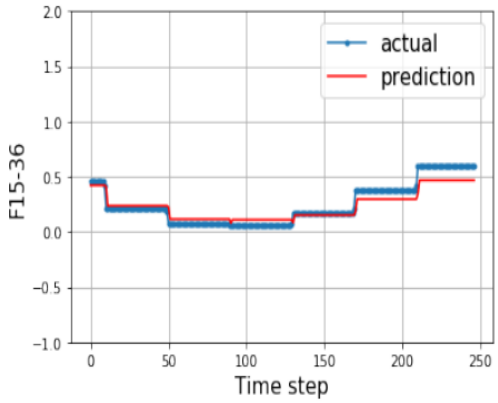


Figure (4.7): Loss of the LSTM Model.

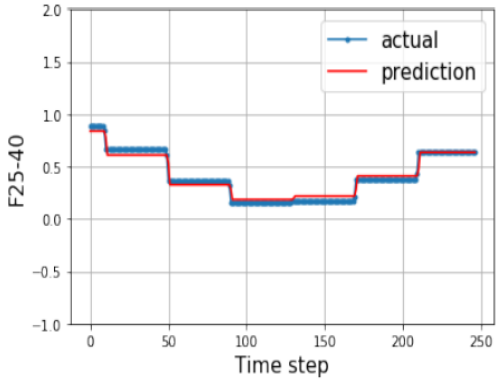
Figure (4.8) illustrated the accuracy of prediction for all features using the LSTM model.



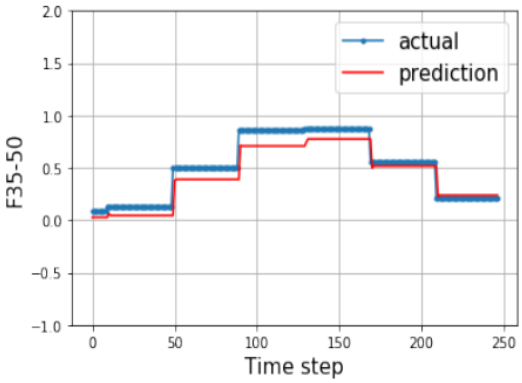




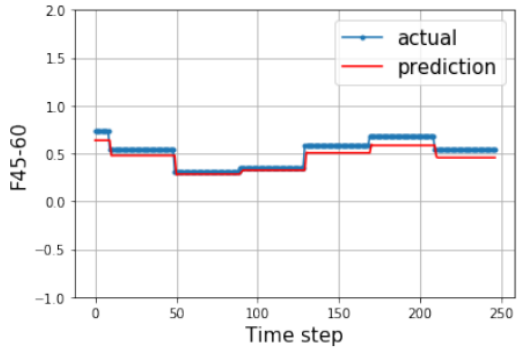
Predict the F25-40 Feature



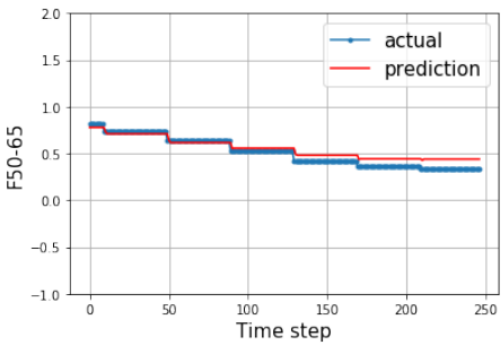
Predict the F35-50 Feature



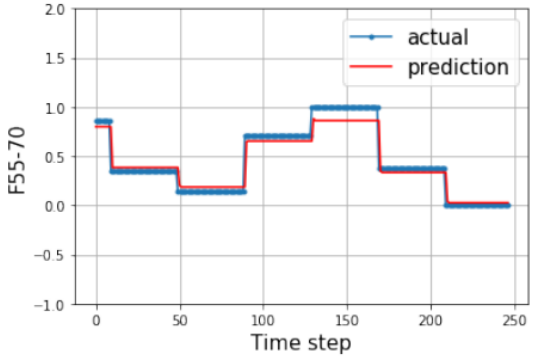
Predict the F45-60 Feature



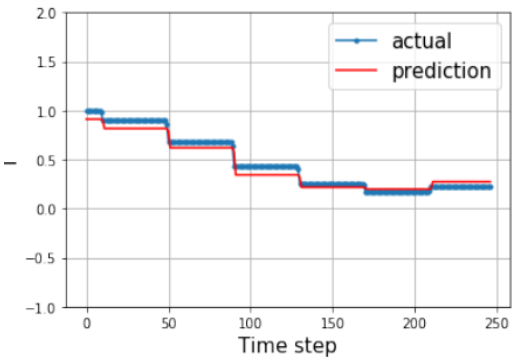
Predict the F50-65 Feature



Predict the F55-70 Feature



Predict the I Feature



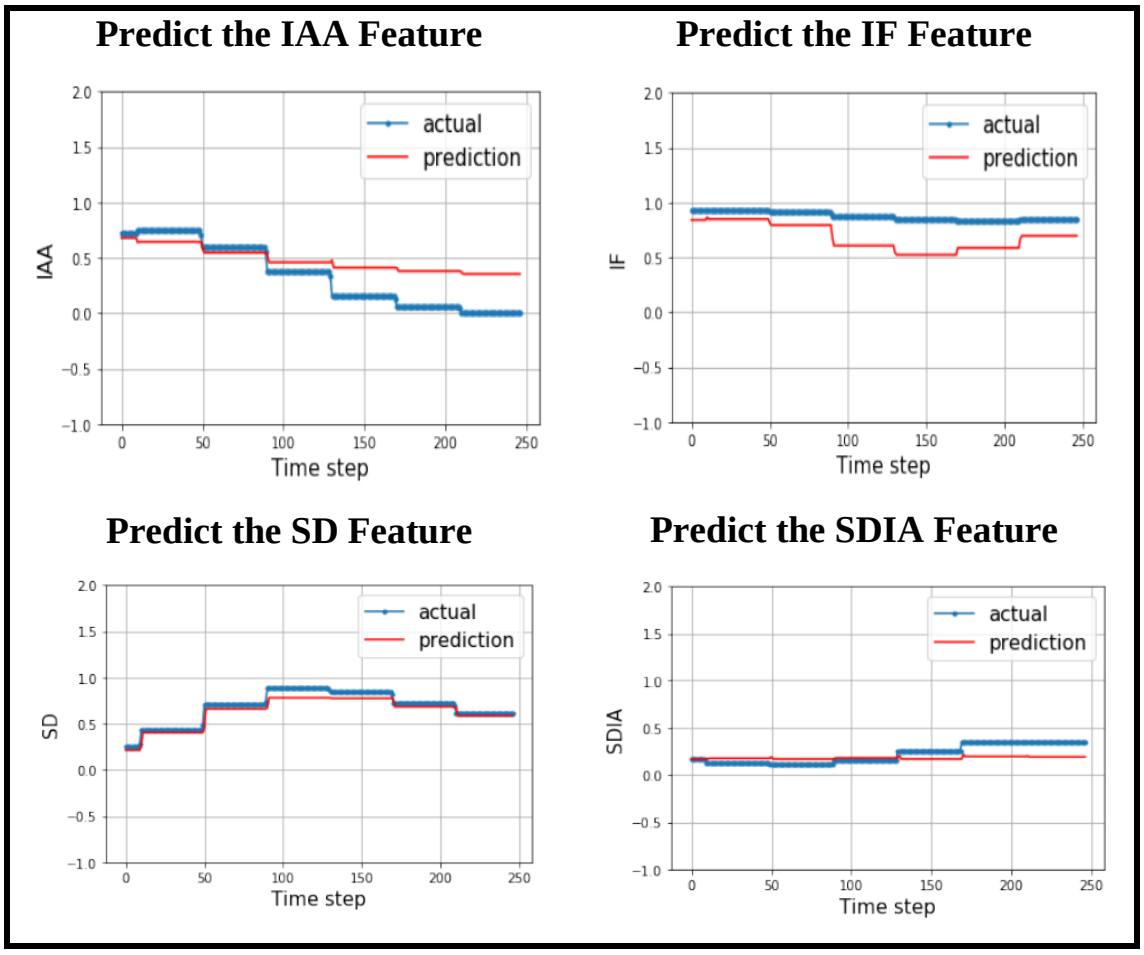


Figure 4.8: Accuracy of Predicted Results in The LSTM Model.

4.4.2.2 Results of the Hybrid Prediction Model

The proposed hybrid prediction model is consisting three layers that are fully connected, the first layer is hidden layer one which has 1000 nodes of LSTM and the second layer is hidden layer two which also has 1000 nodes of GRU. In the first hidden layer, input and output data are in 3D format, while in the second hidden layer, input data is in 3D format and output data is in 2D format. This is because the outputs from the first hidden layer are as inputs to the second hidden layer. The dense layer which is the third and last layer from which the output was obtained, has 18 nodes.

Table (4.6) shows results of the performance of the Hybrid prediction model using (MAE) and (RMSE) in four cases of splitting dataset.

Table 4.6 Results of MAE and RMSE of the Prediction in Hybrid Model

Dataset Splitting Ratios	MAE	RMSE
60% Training &40 % Testing	0.132	0.156
70% Training &30 % Testing	0.135	0.155
80% Training &20 % Testing	0.103	0.123
90% Training &10 % Testing	0.101	0.111

As shown in the table above, the performance of the predictive model was studied and its performance was evaluated based on the results of error measures in four cases of database partitioning. The study shows the best performance of the Hybrid model with 90% Training & 10 % Testing case, where the MAE=0.101 and RMSE=0.111.While in case 80% Training & 20% Testing the MAE =0.103 and RMSE =0.123 .The worst performance of the proposed hybrid model in case of the 60% Training & 40% Testing the MAE=0.135 and RMSE=0.155 and with 70% Training and 30% Testing, where MAE=0.132 and RMSE=0.156.

Figure (4.13) shows the loss of the proposed hybrid model, the loss for training is approximately =0.25 and for testing is approximately = 0.10 in 100 epochs.

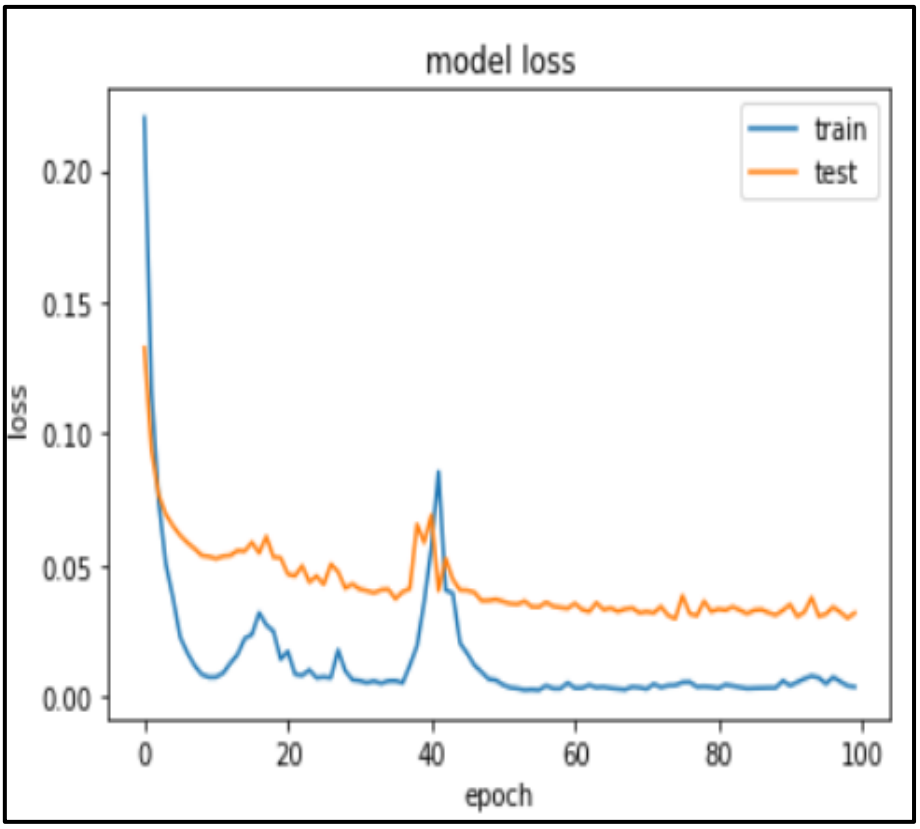
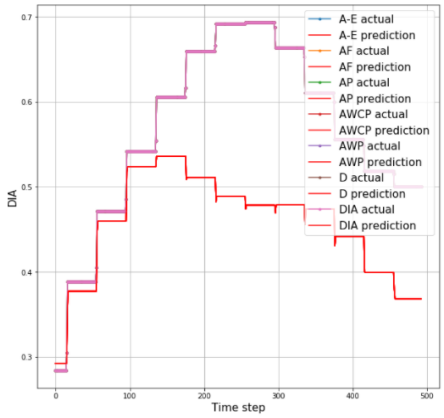


Figure (4.9): Loss of the Proposed Hybrid Model**.**

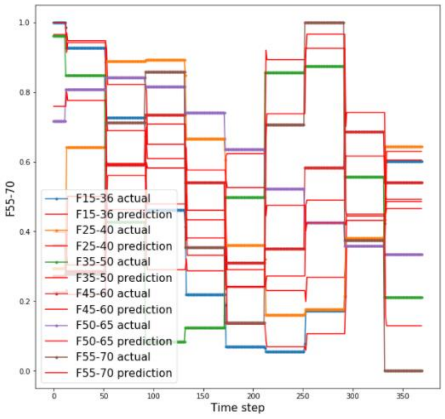
Figure (4.14) illustrated the accuracy of prediction for all features using the proposed hybrid model.

Predict of the DIA Feature



Predict SDIA

Predict set of Features : (F15-36,F25-40,F35-F50,F45-60,and F55-70)



Predict I feature

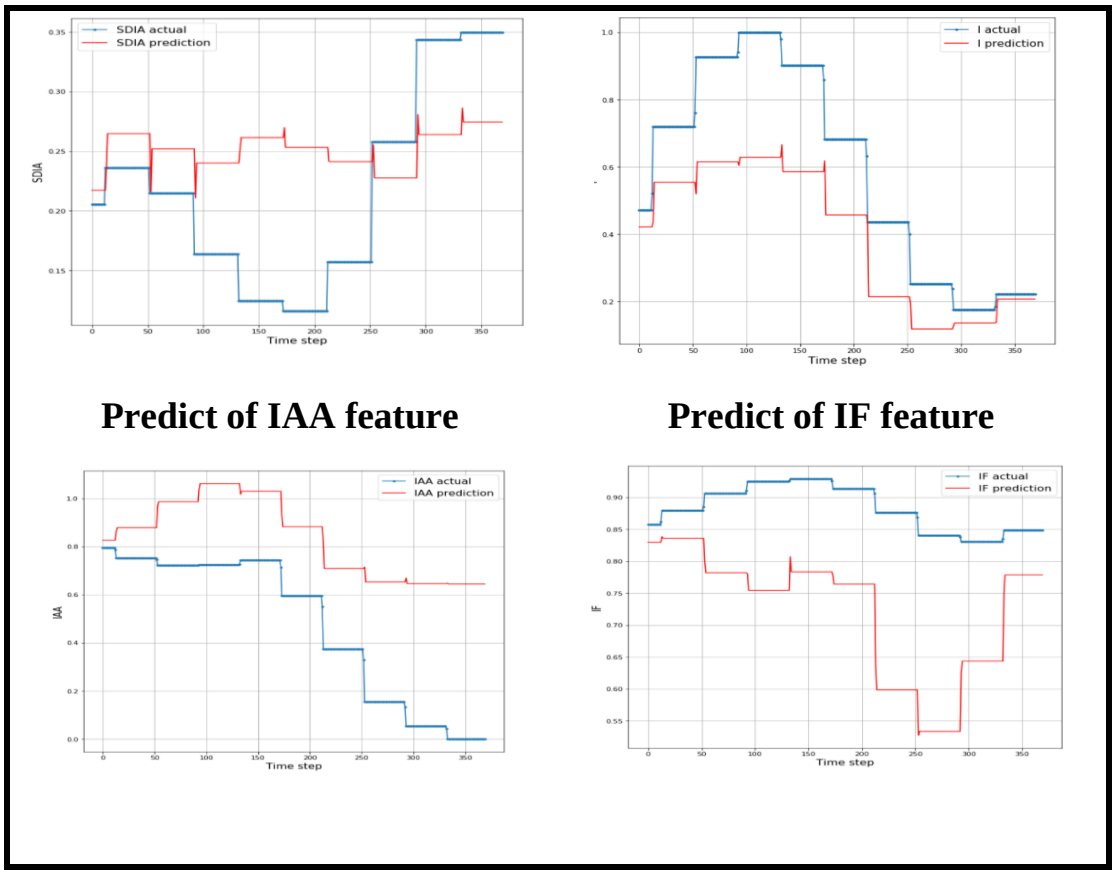


Figure 4.10: Accuracy of Predicted Results in The Hybrid Model.

A comparison of the proposed LSTM, GRU, and ANN model and LSTM model based on the results of error measures that discussion in Tables (4.4) and (4.5) is shown in Figures (4.11) and (4.12).

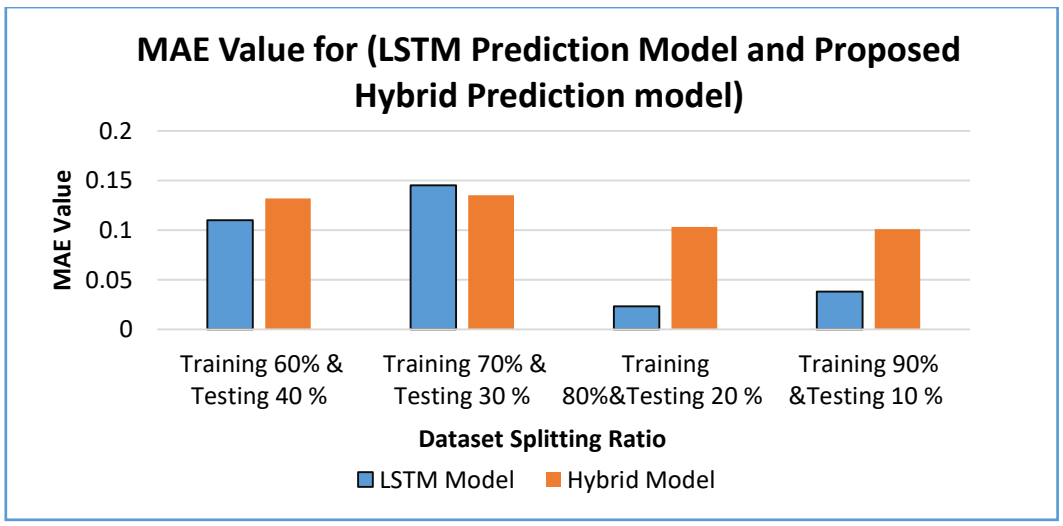


Figure (4.11): Compare Between LATM and Hybrid Model Based on MAE Metrics.

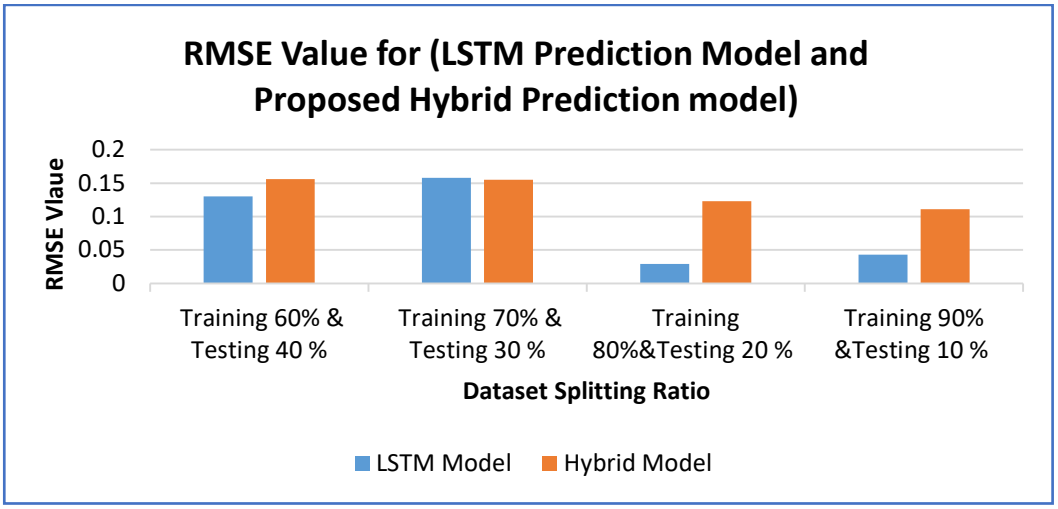


Figure (4.12): Compare Between LATM and Hybrid Model Based on RMSE Metrics.

As shown in Figures (4.11) and (4.12), proved the LSTM, in general, has the best accuracy more than the proposed hybrid prediction model. The best value of the MAE =0.023 for the LSTM model while the best value of the MAE =0.101 for the proposed hybrid model.

The best value of the RMSE =0.029 for the LSTM model while the best value of the RMSE =0.111 for the proposed hybrid model.

4.5 Comparison of Performance Evaluation Between Machine Learning and Deep Learning

Since the proposed model consists of two sub-models, this section will compare them to assess which model was the most accurate in predicting porosity resulting from the seismic data set of oil reservoirs. The comparison is made based on the values of the error measures that were used to evaluate the performance of all the algorithms used in building the prediction model.

Deep learning is the first model, which uses algorithms (LSTM, proposed hybrid LSTM, GRU, and ANN), and the second model is machine learning, which uses algorithms (SVR, KNR). All algorithms used in each model are the most efficient and common algorithms with time-series data.

Based on the results shown in sections (4.3.6) and (4.3.7), Figure (4.13) illustrates the comparison between the machine learning model and deep learning model based on values of MAE for each algorithm in four cases of splitting dataset.

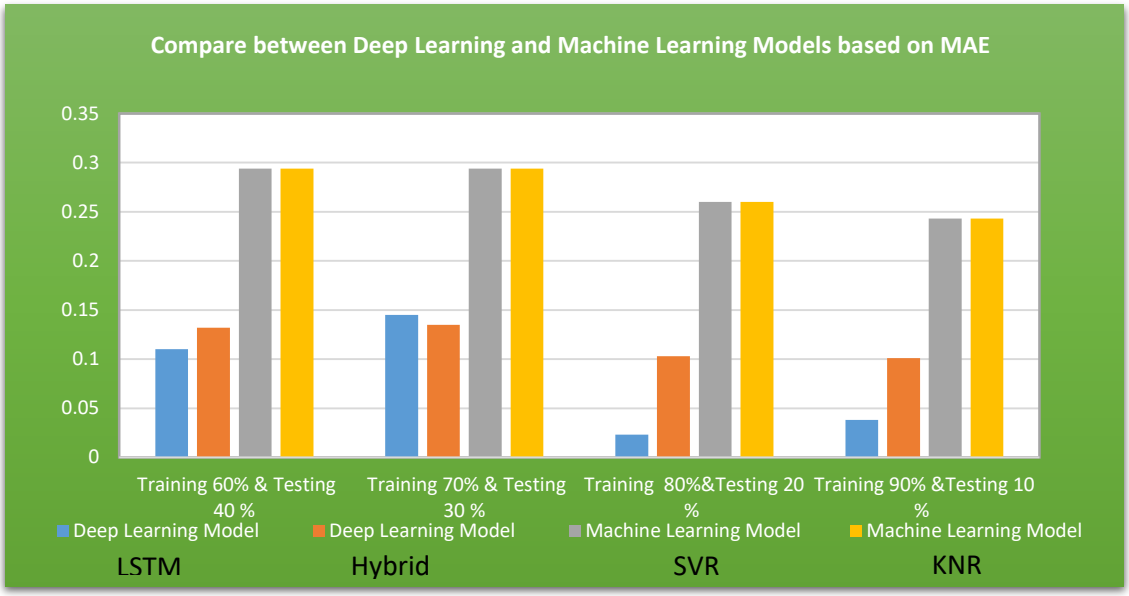
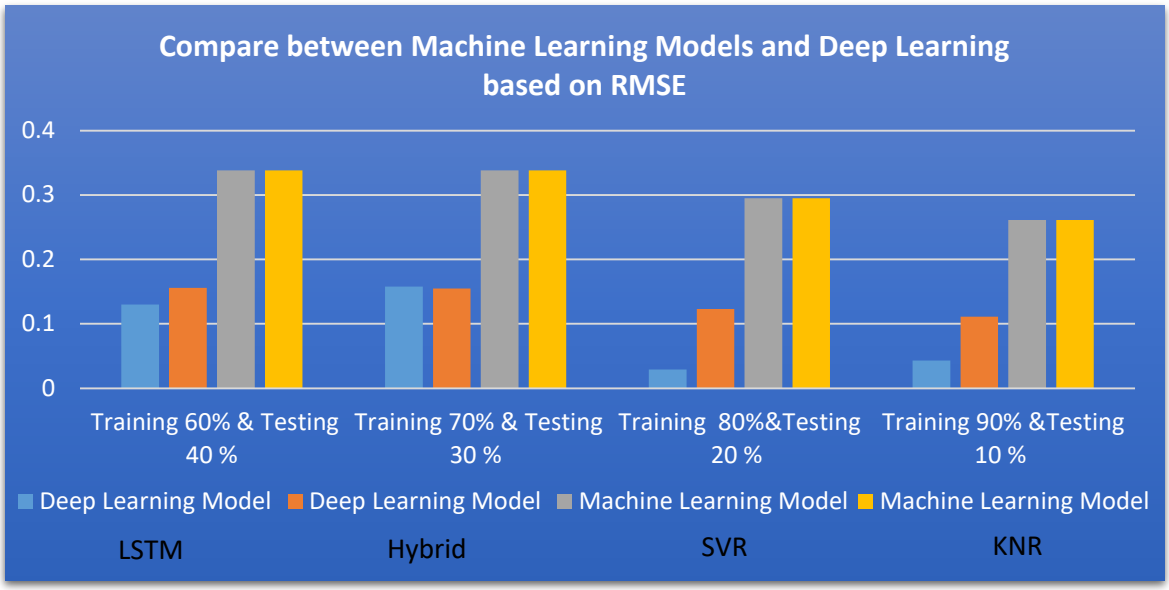


Figure (4.13): Comparison between deep learning and machine learning algorithms used in the proposed model based on absolute error ratio values (MAE).

Figure (4.14) shows the comparison between the machine learning model and the Deep learning model based on values of RMSE for each algorithm in four cases of splitting dataset.



Figure(4.14): Comparison Between Machine Learning and Deep Learning Algorithms used in the Proposed Model Based on Root Means Square Error (RMSE).

Figures (4.13) and (4.14) prove that the performance of deep learning algorithms is better than the performance of machine learning algorithms. The LSTM algorithm obtains the best results overall testing cases of splitting dataset and (MAE & RMSE) metrics.

The main reason that explains the efficiency of deep learning in predicting the porosity generated by the seismic data set of oil reservoirs is the type and size of the data, especially when there are sequential data. The presence of memory in deep learning networks helps to increase their performance and also when the amount of trained data increases, for this reason, the performance of both (LSTM) and the proposed mixed (LSTM, GRU, and ANN) is better than (SVR and KNR).

Chapter Five

Conclusions and Future Works

Chapter Five

Conclusions and Future Works

5.1 Introduction

This chapter concludes with some conclusions about the implementation and results of the proposed porosity prediction system. These conclusions are given in section (5.2). Section (5.3) outlines suggestions for future work.

5.2 Conclusions

Some conclusions can be drawn from the results and tests of this work as follows:

1. This study showed that the preprocessing stage of the data does not necessarily improve the accuracy of the used model because all the seismic data do not have any signs other than the area to be explored.
2. The first methodology used was the Pearson relationship that extracted traits with better performance in DL than in ML.
3. The proposed model for the seismic data set was made to obtain the results using a different prediction algorithm and then their results are compared with each other. The proposed model is built based on the most efficient and effective machine learning prediction algorithms, namely SVR and KNR, whereby the SVR algorithm achieved the lowest error rate results but not the best results.
4. In addition, the proposed model is built based on deep learning prediction algorithms, namely LSTM, LSTM proposal, GRU, and mixed ANN, in which the LSTM algorithm achieved the

lowest error rate results. Where the LSTM achieved the best results compared to the hybrid.

5. By dividing the seismic data set into four types of partitions, the best performance of the prediction model using machine learning in case the 90% training and 10% testing. The best results of the MAE =0.243 and RMSE=0.261 with SVR while the KNR algorithms achieved MAE=0.246 and RMSE =0.261.
6. The best performance of the proposed model is based on deep learning algorithms obtained with seismic dataset partition case the 80% training and 20% testing data. The LSTM achieved the best values of the MAE=0.023 and RMSE=0.029.
7. The performance of the proposed model using deep learning algorithms is better than the performance of the proposed model using machine learning algorithms. The LSTM algorithm obtains the best results for test cases (MAE & RMSE) in all cases of the data set partition.

5.2 Suggestions for Future Works

For future work, it is recommended to take into consideration the following points:

1. Test other types of seismic data that have a large number of features to get the best prediction accuracy.

Use other methods to preprocess the data, apply it to the data, and then test their effect on the accuracy of the prediction.

2. Try to combine the PSO with another supervised learning approach for prediction purpose instead the (SVR), K Nearest Neighbors (KNR) etc., and compare the obtained result with those prediction algorithms.

3. Use other techniques to extract seismic data features such as ANOVA test .

4. As a further development of the model, we aim to extend the capability of multiple platforms by using and experimenting with pool datasets from wells to enable professionals to gain feedback on forecast results.

References

[1] E. Graham and J. S. Ovadia, “Oil exploration and production in Sub-Saharan Africa, 1990-present: Trends and developments,” *Extra. Ind. Soc.*, vol. 6, no. 2, pp. 593–609, 2019, doi: https://doi.org/10.1016/j.exis.2019.02.001.

[2] D. R. Cox *et al.*, “Geohazard detection using 3D seismic data to enhance offshore scientific drilling site selection,” *Sci. Drill.*, vol. 28, pp. 1–27, 2020.

[3] F. Anifowose, A. Abdulraheem, and A. Al-Shuhail, “A parametric study of machine learning techniques in petroleum reservoir permeability prediction by integrating seismic attributes and wireline data,” *J. Pet. Sci. Eng.*, vol. 176, pp. 762–774, 2019, doi: 10.1016/j.petrol.2019.01.110.

[4] Z. Zhong and T. R. Carr, “Application of a new hybrid particle swarm optimization-mixed kernels function-based support vector machine model for reservoir porosity prediction: A case study in Jacksonburg-Stringtown oil field, West Virginia, USA,” *Interpretation*, vol. 7, no. 1, pp. T97–T112, 2019, doi: 10.1190/int-2018-0093.1.

[5] L. KUANG *et al.*, “Application and development trend of artificial intelligence in petroleum exploration and development,” *Pet. Explor. Dev.*, vol. 48, no. 1, pp. 1–14, 2021, doi: 10.1016/S1876-3804(21)60001-0.

[6] I. H. Sarker, “Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions,” *SN Comput. Sci.*, vol. 2, no. 6, pp. 1–20, 2021, doi: 10.1007/s42979-021-00815-1.

[7] N. Pham, X. Wu, and E. Zabihi Naeini, “Missing well log prediction using convolutional long short-term memory network,” *Geophysics*, vol. 85, no. 4, pp. WA159–WA171, 2020, doi: 10.1190/geo2019-0282.1.

[8] S. R. Na’imi, S. R. Shadizadeh, M. A. Riahi, and M. Mirzakhanian, “Estimation of reservoir porosity and water saturation based on seismic attributes using support vector regression approach,” *J. Appl. Geophys.*, vol. 107, pp. 93–101, 2014, doi: 10.1016/j.jappgeo.2014.05.011.

[9] A. Gholami and H. R. Ansari, “Estimation of porosity from seismic attributes using a committee model with bat-inspired optimization algorithm,” *J. Pet. Sci. Eng.*, vol. 152, no. March, pp. 238–249, 2017, doi: 10.1016/j.petrol.2017.03.013.

[10] S. P. Maurya, K. H. Singh, and N. P. Singh, “Qualitative and quantitative comparison of geostatistical techniques of porosity prediction from the seismic and logging data: a case study from the Blackfoot Field, Alberta, Canada,” *Mar. Geophys. Res.*, vol. 40, no. 1, pp. 51–71, 2019, doi: 10.1007/s11001-018-9355-6.

[11] X. Zhou, M. Tyagi, G. Zhang, H. Yu, and Y. Chen, “Data driven modeling and prediction for reservoir characterization using seismic attribute analyses and big data analytics,” *Proc. - SPE Annu. Tech. Conf. Exhib.*, vol. 2019-Septe, 2019, doi: 10.2118/195856-ms.

[12] Q. Zhang, C. Wei, Y. Wang, S. Du, Y. Zhou, and H. Song, “Potential for prediction of water saturation distribution in reservoirs utilizing machine learning methods,” *Energies*, vol. 12, no. 19, 2019, doi: 10.3390/en12193597.

[13] W. Liu, W. D. Liu, and J. Gu, “Forecasting oil production using ensemble empirical model decomposition based Long Short-Term Memory neural network,” *J. Pet. Sci. Eng.*, vol. 189, no. August 2019, p. 107013, 2020, doi: 10.1016/j.petrol.2020.107013.

[14] A. Ogbamikhumi and J. O. Ebeniro, “Reservoir properties estimation from 3D seismic data in the Alose field using artificial intelligence,” *J. Pet. Explor. Prod.*, vol. 11, no. 3, pp. 1275–1287, 2021, doi: 10.1007/s13202-021-01105-5.

[15] A. Nyangarika, “Influence oil price towards macroeconomic indicators in Russia,” 2019.

[16] S. Li *et al.*, “Deep-learning inversion of seismic data,” *arXiv Prepr. arXiv1901.07733*, 2019.

[17] A. Makama, K. Kuladinithi, and A. Timm-Giel, “Wireless geophone networks for land seismic data acquisition: A survey, tutorial and performance evaluation,” *Sensors*, vol. 21, no. 15, 2021, doi: 10.3390/s21155171.

[18] H. Attia *et al.*, “Wireless Geophone Sensing System for Real-Time Seismic Data Acquisition,” *IEEE Access*, vol. 8, pp. 81116–81128, 2020, doi: 10.1109/ACCESS.2020.2989280.

[19] A. A. Al-Shuhail and W. A. Mousa, “Processing of Seismic Reflection Data Using MATLAB,” in *74th EAGE Conference and Exhibition-Workshops*, 2012, p. cp-295.

[20] M. L. Brodie, “The First Law of Data Science: Do Umbrellas Cause Rain?,” *Oper. Database Manag. Syst. KDNuggets*, 2014.

[21] M. L. Brodie, “Applied Data Science,” *Appl. Data Sci.*, no. September, 2019, doi: 10.1007/978-3-030-11821-1.

[22] V. Nourani, M. Sayyah-Fard, M. T. Alami, and E. Sharghi, “Data pre-processing effect on ANN-based prediction intervals construction of the evaporation process at different climate regions in Iran,” *J. Hydrol.*, vol. 588, p. 125078, 2020, doi: https://doi.org/10.1016/j.jhydrol.2020.125078.

[23] M. Natrella, “NIST/SEMATECH e-handbook of statistical methods, 2010,” *URL http//goo. gl/8ElJs*, 2010.

[24] J. T. Behrens, “Principles and procedures of exploratory data analysis.,” *Psychol. Methods*, vol. 2, no. 2, p. 131, 1997.

[25] J. Qian, R. Wang, G. Jin, and Y. Li, “Breaking Boundaries among Hardware Curriculums by EDA Technology,” in *2008 The 9th International Conference for Young Computer Scientists*, 2008, pp. 2536–2540, doi: 10.1109/ICYCS.2008.327.

[26] T. Nakaya and K. Yano, “Visualising crime clusters in a space-time cube: An exploratory data-analysis approach using space-time kernel density estimation and scan statistics,” *Trans. GIS*, vol. 14, no. 3, pp. 223–239, 2010.

[27] V. Cox, “Exploratory data analysis,” in *Translating Statistics to Make Decisions*, Springer, 2017, pp. 47–74.

[28] D. EDMONDS, “Chapter 13 - Commodity trading advisors and their role in managed futures,” in *Quantitative Finance*, E. Acar and S. B. T.-A. T. R. (Second E. Satchell, Eds. Oxford: Butterworth-Heinemann, 2002, pp. 367–387.

[29] J. Mun, “Chapter 3 - A Primer on Quantitative Risk Analysis,” M. Glantz and R. B. T.-M.-A. R. M. Kissell, Eds. San Diego: Academic Press, 2014, pp. 63–118.

[30] D. Nettleton, “Chapter 6 - Selection of Variables and Factor Derivation,” D. B. T.-C. D. M. Nettleton, Ed. Boston: Morgan

Kaufmann, 2014, pp. 79–104.

[31] B. Süssmuth, “Time Series Analysis for Macroeconomics and Finance Welcome !,” no. 773, pp. 1–10, 2010.

[32] P. J. Brockwell, “Time Series Analysis,” P. Peterson, E. Baker, and B. B. T.-I. E. of E. (Third E. McGaw, Eds. Oxford: Elsevier, 2010, pp. 474–481.

[33] H. Madsen, *Time Series Analysis*. 2008.

[34] D. E. Shasha and Y. Zhu, *High performance discovery in time series: techniques and case studies*. Springer Science & Business Media, 2004.

[35] J. Li *et al.*, “Networked human motion capture system based on quaternion navigation,” *BodyNets Int. Conf. Body Area Networks*, vol. 45, no. 1, 2017, doi: 10.1145/0000000.0000000.

[36] S. Choubey and G. P. Karmakar, “Artificial intelligence techniques and their application in oil and gas industry,” *Artif. Intell. Rev.*, vol. 54, pp. 1–19, Jun. 2021, doi: 10.1007/s10462-020-09935-1.

[37] M. Supriya and V. K. Chattu, “A review of artificial intelligence, big data, and blockchain technology applications in medicine and global health,” *Big Data Cogn. Comput.*, vol. 5, no. 3, 2021, doi: 10.3390/bdcc5030041.

[38] A. Sircar, K. Yadav, K. Rayavarapu, N. Bist, and H. Oza, “Application of machine learning and artificial intelligence in oil and gas industry,” *Pet. Res.*, no. xxxx, 2021, doi: 10.1016/j.ptlrs.2021.05.009.

[39] A. S. Lundervold and A. Lundervold, “An overview of deep learning in medical imaging focusing on MRI,” *Z. Med. Phys.*, vol. 29, no. 2, pp. 102–127, 2019, doi: https://doi.org/10.1016/j.zemedi.2018.11.002.

[40] R. R. R. Robinson and C. Thomas, “Ranking of machine learning algorithms based on the performance in classifying DDoS attacks,” *2015 IEEE Recent Adv. Intell. Comput. Syst. RAICS 2015*, no. December, pp. 185–190, 2016, doi: 10.1109/RAICS.2015.7488411.

[41] A. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, “Speech Recognition Using Deep Neural Networks: A Systematic Review,” *IEEE Access*, vol. PP, p. 1, Feb. 2019, doi:

10.1109/ACCESS.2019.2896880.

[42] Q. Liu and Y. Wu, “Supervised Learning BT - Encyclopedia of the Sciences of Learning,” N. M. Seel, Ed. Boston, MA: Springer US, 2012, pp. 3243–3245.

[43] M. Abukmeil, S. Ferrari, A. Genovese, V. Piuri, and F. Scotti, “Unsupervised Learning From Limited Available Data by β-NMF and Dual Autoencoder,” in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 81–85, doi: 10.1109/ICIP40778.2020.9191252.

[44] A. Subramanya and P. P. Talukdar, *Graph-Based Semi-Supervised Learning*. Morgan & Claypool, 2014.

[45] M. Miyashita, T. Kondo, and S. Yano, “Reinforcement learning with constraint based on mirror descent algorithm,” *Results Control Optim.*, vol. 4, no. June, p. 100048, 2021, doi: 10.1016/j.rico.2021.100048.

[46] M. Awad and R. Khanna, “Support vector regression,” in *Efficient learning machines*, Springer, 2015, pp. 67–80.

[47] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, “Support vector regression machines,” *Adv. Neural Inf. Process. Syst.*, vol. 9, pp. 155–161, 1997.

[48] V. Blanco, A. Japón, and J. Puerto, “Optimal arrangements of hyperplanes for SVM-based multiclass classification,” *Adv. Data Anal. Classif.*, vol. 14, no. 1, pp. 175–199, 2020, doi: 10.1007/s11634-019-00367-6.

[49] J. P. Karmy and S. Maldonado, “Hierarchical time series forecasting via Support Vector Regression in the European Travel Retail Industry,” *Expert Syst. Appl.*, vol. 137, pp. 59–73, 2019, doi: https://doi.org/10.1016/j.eswa.2019.06.060.

[50] X. Ding, M. Hasanipanah, H. Nikafshan Rad, and W. Zhou, “Predicting the blast-induced vibration velocity using a bagged support vector regression optimized with firefly algorithm,” *Eng. Comput.*, vol. 37, no. 3, pp. 2273–2284, 2021, doi: 10.1007/s00366-020-00937-9.

[51] M. Sabzekar and S. M. H. Hasheminejad, “Robust regression using support vector regressions,” *Chaos, Solitons and Fractals*, vol. 144, p. 110738, 2021, doi: 10.1016/j.chaos.2021.110738.

[52] Y. Song, J. Liang, J. Lu, and X. Zhao, “An efficient instance selection algorithm for k nearest neighbor regression,” *Neurocomputing*, vol. 251, pp. 26–34, 2017, doi: 10.1016/j.neucom.2017.04.018.

[53] W. T. Ho and F. W. Yu, “Chiller system optimization using k nearest neighbour regression,” *J. Clean. Prod.*, vol. 303, 2021, doi: 10.1016/j.jclepro.2021.127050.

[54] K. Shah, H. Patel, D. Sanghvi, and M. Shah, “A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification,” *Augment. Hum. Res.*, vol. 5, no. 1, 2020, doi: 10.1007/s41133-020-00032-0.

[55] S. Zhang, D. Cheng, Z. Deng, M. Zong, and X. Deng, “A novel kNN algorithm with data-driven k parameter computation,” *Pattern Recognit. Lett.*, vol. 109, pp. 44–54, 2018, doi: 10.1016/j.patrec.2017.09.036.

[56] K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” *Comput. Electron. Agric.*, vol. 145, no. September 2017, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009.

[57] D. A. Bashar, “Survey on Evolving Deep Learning Neural Network Architectures,” *J. Artif. Intell. Capsul. Networks*, vol. 2019, no. 2, pp. 73–82, 2019, doi: 10.36548/jaicn.2019.2.003.

[58] M. Bahi and M. Batouche, “Deep Learning for Ligand-Based Virtual Screening in Drug Discovery,” *Proc. - PAIS 2018 Int. Conf. Pattern Anal. Intell. Syst.*, no. October 2018, 2018, doi: 10.1109/PAIS.2018.8598488.

[59] T. Lei, Y. Zhang, S. I. Wang, H. Dai, and Y. Artzi, “Simple recurrent units for highly parallelizable recurrence,” *Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018*, pp. 4470–4481, 2020, doi: 10.18653/v1/d18-1477.

[60] A. M. Alaa and M. Van Der Schaar, “Frequentist uncertainty in recurrent neural networks via blockwise influence functions,” *37th Int. Conf. Mach. Learn. ICML 2020*, vol. PartF16814, pp. 152–167, 2020.

[61] A. Graves, A. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *2013 IEEE international conference on acoustics, speech and signal processing*, 2013, pp.

6645–6649.

[62] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Gated feedback recurrent neural networks,” in *International conference on machine learning*, 2015, pp. 2067–2075.

[63] R. Zhang, Z. Yuan, and X. Shao, “A New Combined CNN-RNN Model for Sector Stock Price Analysis,” in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, 2018, vol. 02, pp. 546–551, doi: 10.1109/COMPSAC.2018.10292.

[64] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[65] P. Zhang, J. Xue, C. Lan, W. Zeng, Z. Gao, and N. Zheng, “EleAtt-RNN: Adding attentiveness to neurons in recurrent neural networks,” *IEEE Trans. Image Process.*, vol. 29, pp. 1061–1073, 2020, doi: 10.1109/TIP.2019.2937724.

[66] J. Chung, S. Ahn, and Y. Bengio, “Hierarchical multiscale recurrent neural networks,” *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, no. c, pp. 1–13, 2017.

[67] J. Zheng, C. Xu, Z. Zhang, and X. Li, “Electric load forecasting in smart grids using Long-Short-Term-Memory based Recurrent Neural Network,” *2017 51st Annu. Conf. Inf. Sci. Syst. CISS 2017*, no. October, 2017, doi: 10.1109/CISS.2017.7926112.

[68] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, “Recent Advances in Recurrent Neural Networks,” pp. 1–21, 2017, [Online]. Available: http://arxiv.org/abs/1801.01078.

[69] F. Landi, L. Baraldi, M. Cornia, and R. Cucchiara, “Working Memory Connections for LSTM,” *Neural Networks*, vol. 144, pp. 334–341, 2021, doi: 10.1016/j.neunet.2021.08.030.

[70] L. Li, Y. Wang, X. Yuan, C. Yang, and W. Gui, “Quality prediction model for process sequential data of irregular measurements with sampling-interval-attention LSTM,” *Proc. - 2020 Chinese Autom. Congr. CAC 2020*, pp. 7186–7191, 2020, doi: 10.1109/CAC51589.2020.9327654.

[71] R. C. Staudemeyer and E. R. Morris, “Understanding LSTM--a tutorial into Long Short-Term Memory Recurrent Neural Networks,” *arXiv Prepr. arXiv1909.09586*, 2019.

[72] I. K. Ihianle, A. O. Nwajana, S. H. Ebenuwa, R. I. Otuka, K. Owa, and M. O. Orisatoki, "A deep learning approach for human activities recognition from multimodal sensing devices," *IEEE Access*, vol. 8, pp. 179028–179038, 2020, doi: 10.1109/ACCESS.2020.3027979.

[73] X. Li, L. Peng, X. Yao, S. Cui, C. You, and T. Chi, "Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation," *Environ. Pollut.*, vol. 231, pp. 997–1004, Sep. 2017, doi: 10.1016/j.envpol.2017.08.114.

[74] Z. Yu, D. S. Moirangthem, and M. Lee, "Continuous Timescale Long-Short Term Memory Neural Network for Human Intent Understanding," *Front. Neurorobot.*, vol. 11, p. 42, 2017, doi: 10.3389/fnbot.2017.00042.

[75] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, pp. 1724–1734, 2014, doi: 10.3115/v1/d14-1179.

[76] G. B. Zhou, J. Wu, C. L. Zhang, and Z. H. Zhou, "Minimal gated unit for recurrent neural networks," *Int. J. Autom. Comput.*, vol. 13, no. 3, pp. 226–234, 2016, doi: 10.1007/s11633-016-1006-2.

[77] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," pp. 1–9, 2014, [Online]. Available: http://arxiv.org/abs/1412.3555.

[78] R. Dey and F. M. Salem, "Gate-Variants of Gated Recurrent Unit (GRU)," vol. 784.

[79] B. Wang, W. Kong, H. Guan, and N. N. Xiong, "Air Quality Forecasting Based on Gated Recurrent Long Short Term Memory Model in Internet of Things," *IEEE Access*, vol. 7, pp. 69524–69534, 2019, doi: 10.1109/ACCESS.2019.2917277.

[80] R. B. Ivry and R. M. C. Spencer, "The neural representation of time," *Curr. Opin. Neurobiol.*, vol. 14, no. 2, pp. 225–232, 2004.

[81] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, 2017, pp. 1597–

1600.

[82] A. Taherkhani, A. Belatreche, Y. Li, G. Cosma, L. P. Maguire, and T. M. McGinnity, “A review of learning in biologically plausible spiking neural networks,” *Neural Networks*, vol. 122, pp. 253–272, 2020.

[83] E. D. Grossman and R. Blake, “Brain areas active during visual perception of biological motion,” *Neuron*, vol. 35, no. 6, pp. 1167–1175, 2002.

[84] M. M. Hussain, S. H. Bari, I. Mahmud, and M. I. H. Siddiquee, “Chapter 5 - Application of different artificial neural network for streamflow forecasting,” P. Sharma and D. B. T.-A. in S. F. Machiwal, Eds. Elsevier, 2021, pp. 149–170.

[85] D. Toghraie, N. Sina, N. A. Jolfaei, M. Hajian, and M. Afrand, “Designing an Artificial Neural Network (ANN) to predict the viscosity of Silver/Ethylene glycol nanofluid at different temperatures and volume fraction of nanoparticles,” *Phys. A Stat. Mech. its Appl.*, vol. 534, p. 122142, 2019, doi: 10.1016/j.physa.2019.122142.

[86] A. Zhang, H. Zhou, X. Li, and W. Zhu, “Fast and robust learning in spiking feed-forward neural networks based on intrinsic plasticity mechanism,” *Neurocomputing*, vol. 365, pp. 102–112, 2019.

[87] D. G. Chachlakis, T. Zhou, F. Ahmad, and P. P. Markopoulos, “Minimum Mean-Squared-Error autocorrelation processing in coprime arrays,” *Digit. Signal Process.*, vol. 114, p. 103034, 2021, doi: https://doi.org/10.1016/j.dsp.2021.103034.

[88] B. Ait-Amir, P. Pougnet, and A. El Hami, “6 - Meta-Model Development,” A. El Hami and P. B. T.-E. M. S. 2 Pougnet, Eds. Elsevier, 2015, pp. 151–179.

الخلاصة

أدى التطور الهائل في تشكيلات النفط الصخري إلى تغيير قواعد اللعبة. من ناحية أخرى ، يلعب التعلم الآلي (ML) والتعلم العميق (DL) دورًا مهمًا في التطور السريع لجميع الصناعات من خلال أتمته معظم العمليات الروتينية. تحصل صناعة النفط أيضًا على فوائد متساوية من ML و DL لتخطيط تطوير المكامن والدقة التشغيلية من خلال سلسلة من الأنظمة الآلية. لتطوير هذا المجال ، يتم إنشاء نماذج محاكاة حسابية ثابتة وديناميكية بناءً على الخصائص البتروفيزيائية المختلفة التي تم جمعها من خلال موارد مختلفة مستهلكة للوقت ومكلفة. تهدف هذه الدراسة إلى تقديم نموذج شامل في مجال تطبيق ML و DL لنمذجة الخصائص البتروفيزيائية باستخدام طرق وخوارزميات مختلفة. أخيرًا ، تمت مناقشة تقنيات ML و DL المتعددة التي تم اختبارها في هذه الدراسة بالتفصيل من أجل تحقيق المزيد من الدقة في نماذج المحاكاة البتروفيزيائية.، تم استخدام نموذج من خوارزميات التعلم العميق للشبكات العصبية المتكررة (ANN) الذاكرة طويلة المدى (LSTM) وتغذيتها بواسطة شبكة عصبية اصطناعية (ANN) للتنبؤ بالمسامية. لتحسين النتائج، تم استخدام طريقة هجينة (LSTM) وتم تغذيتها أيضًا بوحدة البوابات المتكررة (GRU) وتغذيتها بشبكة عصبية اصطناعية (ANN). لذلك، لتحسين النموذج، تم استخدام خوارزميات الانحدار Support Vector Regression (SVR) وانحدار K الأقرب جار (KNN)، حيث تمت مناقشة نتائج الانحدار. لوحظ أن خوارزميات التعلم العميق للشبكات العصبية المتكررة تفوقت على خوارزميات الانحدار. هذا يعني أنه باستخدام خوارزميات التعلم العميق، يمكننا الحصول على أفضل القرارات في التنبؤ بخزانات النفط وتقليل عدم الثقة في التنقيب والحفر. باستخدام مقاييس الخطأ MAS & RMSE وتقسيم مجموعة البيانات إلى أربعة أنواع من الأقسام، أظهرت النتائج أفضل أداء للنموذج المقترح بناءً على خوارزميات التعلم العميق. مع حالة تقسيم مجموعة البيانات الزلزالية، تم الحصول على ٨٠٪ من بيانات التدريب و٢٠٪ من بيانات الاختبار. حيث أن LSTM كان لها أفضل قيم MAE = 0.023 و RMSE = 0.029. يعتبر نموذج التنبؤ الذي يستخدم التعلم الآلي هو الأفضل أداءً في حالة اختبار ٩٠٪ واختبار ١٠٪. كانت أفضل النتائج هي MAE = 0.238 وRMSE = 0.255 مع SVR، بينما حققت خوارزميات KNR نتائج هي MAE=0.276 و RMSE=0.301 .

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة ديالى
كلية العلوم

التنبؤ بمسامية المكامن بناءً على سمات البيانات الزلزالية الناتجة باستخدام نهج التعلم العميق

رسالة

مقدمة الى كلية العلوم في جامعة ديالى وهي جزء من متطلبات نيل شهادة الماجستير في علوم الحاسوب

تقدم بها الباحث

محمد وهاب رحيم

بإشراف

ا.م.د. عبد الباسط كاظم شكر

١٤٤٣هـ ٢٠٢٢م