T.R.
SAKARYA UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

IMPROVING THE PREDICTION OF OIL AND GAS PRODUCTION USING ARTIFICIAL INTELLIGENCE ALGORITHMS

MSc THESIS

Azhar Naji Muhajir ALYAHYA

Software Engineering Department

Software Engineering Program

APRIL 2025

T.R.
SAKARYA UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

IMPROVING THE PREDICTION OF OIL AND GAS PRODUCTION USING ARTIFICIAL INTELLIGENCE ALGORITHMS

MSc THESIS

Azhar Naji Muhajir ALYAHYA

Software Engineering Department

Software Engineering Program

Thesis Advisor: Dr.Öğr.Üyesi Gülüzar ÇİT

APRIL 2025

The thesis work titled "**IMPROVING THE PREDICTION OF OIL AND GAS PRODUCTION USING ARTIFICIAL INTELLIGENCE ALGORITHMS**" prepared by Azhar Naji Muhajir ALYAHYA was accepted by the following jury on 22/04/2025 by unanimously/majority of votes as a MSc THESIS in Sakarya University Graduate School of Natural and Applied Sciences, Software Engineering Department.

Thesis Jury

Head of Jury:	**Dr. Gülüzar ÇİT (Advisor)** Sakarya University	.....
Jury Member:	**Dr. Beyza EKEN** Sakarya University	.....
Jury Member:	**Dr. Muhammed Ali Nur ÖZ** Sakarya Applied Sciences University	.....

STATEMENT OF COMPLIANCE WITH THE ETHICAL PRINCIPLES AND RULES

I declare that the thesis work titled " IMPROVING THE PREDICTION OF OIL AND GAS PRODUCTION USING ARTIFICIAL INTELLIGENCE ALGORITHMS ", which I have prepared in accordance with Sakarya University Graduate School of Natural and Applied Sciences regulations and Higher Education Institutions Scientific Research and Publication Ethics Directive, belongs to me, is an original work, I have acted in accordance with the regulations and directives mentioned above at all stages of my study, I did not get the innovations and results contained in the thesis from anywhere else, I duly cited the references for the works I used in my thesis, I did not submit this thesis to another scientific committee for academic purposes and to obtain a title, in accordance with the articles 9/2 and 22/2 of the Sakarya University Graduate Education and Training Regulation published in the Official Gazette dated 20.04.2016, a report was received in accordance with the criteria determined by the graduate school using the plagiarism software program to which Sakarya University is a subscriber. I accept all kinds of legal responsibility that may arise in case of a situation contrary to this statement.

(22/04/2025)

Azhar Naji Muhajir ALYAHYA

To my family, children, and friends.

ACKNOWLEDGMENTS

I would like, first, to thank Allah and praise Him for His generosity and care, which we experience daily. For all the good people who were supportive when I needed them, I would like to start by expressing my sincere gratitude to my advisor, Dr. Gülüzar Çit, who spared no effort to help me through this study, and for her patience and guidance.

To all my friends who were a source of inspiration throughout the difficulties, thank you.

I want to express my deepest gratitude to my family my dear mother, who never stopped being supportive in every possible way, my children thank you from the bottom of my heart.

Azhar Naji Muhajir ALYAHYA

TABLE OF CONTENTS

Page

ABBREVIATIONS

Adam : Adaptive Moment Estimation
AI : Artificial Intelligence
ANN : Artificial Neural Network
BP : Back-Propagation
CARTs : Classification and Regression Trees
CNN : Convolutional Neural Network
DCN : Decline Curve Analysis
DL : Deep Learning
DTR : Decision Tree Regression
EOR : Enhanced Oil Recovery
GA : Genetic Algorithm
GB : Gradient Boosting
GBDT : Gradient Boosting Decision Trees
GRU : Gated Recurrent Unit
KNN : K-Nearest Neighbor
Light GBM : Light Gradient Boosting Machine
LR : Logistic Regression
LSTM : Long Short-Term Memory
MAE : Mean Absolute Error
ML : Machine Learning
MLP : Multilayer Perceptron
MSE : Mean Square Error
NRS : Numerical Reservoir Simulation
PCA : Principal Component Analysis
PLR : Polynomial Linear Regression
Relu : Rectified Linear Unit
RF : Random Forest
RFR : Random Forest Regression
RNN : Recurrent Neural Network
SVR : Support Vector Regression

Tanh : Hyperbolic tangent

TCN : Temporal Convolutional Network

XGBoost : Extreme Gradient Boosting

SYMBOLS

$\boldsymbol{\alpha}$	: The learning rate
Attr	: Set of descriptive attributes
$\mathbf{B_{output}}$	: Bias for the output layer
$\mathbf{B_{hidden}}$	: Bias for the hidden layer
$\boldsymbol{b_j}$	: Bias component
$\mathbf{b_o}$	: Bias for output gate
$\mathbf{E(n)}$	: Root node
$\mathbf{h_t}$	: Hidden state at time t
$\boldsymbol{i_t}$	: Input gate activation vector
$\boldsymbol{o_t}$	: Output gate activation vector
$\boldsymbol{n_{output}}$	: Number of output neurons
$\boldsymbol{n_{input}}$	: Number of input features
$\boldsymbol{n_{hidden}}$	: Number of hidden layer neurons
$\boldsymbol{n_{estimator}}$	: Number of trees
$\mathbf{R^2}$	: Coefficient of determination
t	: Time [Unit]
$\boldsymbol{w_{ij}}$	: Weight term
$\boldsymbol{W_{input\_hidden}}$	: Weights from input to hidden layer
$\mathbf{W_{hidden\_output}}$	: Weights from hidden to output layer
$\boldsymbol{x_t}$	: Input value
y_i	: True value
$\bar{\boldsymbol{y}}$	: Predicted value
$\boldsymbol{\sigma}$	: The function of sigmoid activation

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ALGORITHMS

Page

IMPROVING THE PREDICTION OF OIL AND GAS PRODUCTION USING ARTIFICIAL INTELLIGENCE ALGORITHMS

SUMMARY

The energy industry heavily relies on accurate oil and gas production forecasting, which is crucial in optimizing resource allocation, production planning, and operational efficiency. Predicting future production rates is fundamental to the petroleum industry, enabling businesses to enhance the extraction and refinement of fuels such as gasoline and diesel while minimizing risks and maximizing profits. Reliable forecasting models help stakeholders make informed decisions regarding investment strategies, infrastructure development, and regulatory compliance. As global energy demands continue to rise, the need for precise, data-driven forecasting methodologies becomes increasingly important to ensure stability and sustainability in the petroleum sector.

Traditionally, forecasting the future production of energy-rich natural gas and oil has been a widely studied topic in petroleum engineering. Due to their similar extraction processes and market demands, these two resources are often analyzed together. Historically, researchers have relied on conventional techniques such as decline curve analysis and numerical reservoir simulation to estimate future production levels. While these approaches have been widely used, they suffer from several critical limitations. The high computational costs, the extensive time required to complete the simulations, and the dependence on multiple assumptions often result in unreliable and inconsistent predictions. Additionally, these traditional models struggle to adapt to the complex, nonlinear nature of oil and gas production, particularly when dealing with fluctuating reservoir conditions, market dynamics, and external environmental factors. As a result, there is a growing need for more advanced, efficient, and accurate predictive methodologies.

In recent years, artificial intelligence (AI) has emerged as a transformative tool in the energy sector, offering the potential to enhance the speed, accuracy, and adaptability of production forecasting. AI-based approaches have significantly reduced the computational time required for forecasting while improving prediction accuracy through automated feature selection, pattern recognition, and model optimization. By leveraging machine learning, ensemble learning, and deep learning techniques, AI-driven models can effectively process large volumes of historical production data to identify trends and make precise forecasts. The integration of AI into petroleum engineering has enabled more efficient decision-making processes, optimized resource utilization, and minimized uncertainties in production planning.

This thesis aims to enhance the accuracy and efficiency of oil and gas production forecasting by implementing a robust AI-driven predictive framework. The proposed system employs a combination of machine learning, ensemble learning, and deep learning techniques to develop highly accurate predictive models. By leveraging past production data, the system aims to generate precise future outcome predictions, enabling companies and stakeholders to improve strategic planning, optimize

resource allocation, and mitigate market risks. The research focuses on evaluating the performance of multiple AI models to determine the most effective methodology for forecasting oil and gas production with minimal error rates.

Eleven different methodologies were used in the proposed system. These consist of five machine learning models: Decision Tree Regressor (DTR), Random Forest Regressor (RFR), K-Nearest Neighbors (KNN), Extreme Gradient Boosting (XGBoost), and Gradient Boosting Regressor (GBR), as well as three ensemble learning models: Bagging, Boosting, and Stacking. Additionally, there are three models in deep learning: Artificial Neural Network (ANN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM). By incorporating these methodologies, the system ensures that multiple aspects of data-driven forecasting are considered, thereby improving robustness and overall performance.

The results of this thesis demonstrate that ensemble learning techniques, particularly the stacking model, offer superior predictive performance compared to standalone machine learning or deep learning models. The ensemble-based approaches significantly reduced error rates in key evaluation metrics, including Mean Absolute Error (MAE) and Mean Squared Error (MSE). Among all the models tested, the stacking model achieved the highest level of accuracy, with an R-squared value of 99%, indicating its exceptional ability to capture complex patterns and dependencies in oil and gas production data. This finding highlights the effectiveness of stacking as a model integration technique, which combines the predictive power of multiple base learners to produce a highly accurate final prediction.

Beyond academic contributions, this thesis has significant practical implications for the petroleum industry. The integration of AI-driven forecasting models into oil and gas production workflows can lead to substantial improvements in decision-making efficiency, cost reduction, and risk management. By utilizing intelligent predictive analytics, industry professionals can make data-driven investment decisions, optimize drilling and extraction strategies, and improve overall operational performance. Furthermore, AI-based models provide greater adaptability to changing reservoir conditions and external economic factors, ensuring more reliable and sustainable production forecasts in the long term. The ability to dynamically adjust to real-time production data, external market fluctuations, and environmental constraints makes AI-powered forecasting a critical component of the future energy sector.

The adoption of AI-based forecasting methodologies has the potential to revolutionize the petroleum industry, ensuring more precise and reliable production planning in an increasingly complex and dynamic energy market. Additionally, continued advancements in AI and data science will enable the integration of real-time monitoring systems, improving responsiveness to changing market demands and environmental conditions. Generating highly accurate and adaptive predictions will be crucial in meeting global energy needs while promoting efficiency and sustainability in the oil and gas sector. Furthermore, the expansion of AI applications in the energy sector may open avenues for automation, enhanced reservoir management, and predictive maintenance, further optimizing extraction and refining processes. Future research will focus on incorporating additional real-time operational parameters, expanding the dataset to include diverse geological formations, and developing hybrid AI architectures that combine the strengths of multiple methodologies to further enhance forecasting performance.

Overall, this thesis provides a comprehensive framework for AI-driven oil and gas production forecasting, demonstrating the immense potential of integrating machine learning, deep learning, and ensemble learning techniques to achieve highly accurate and efficient predictions. The continued evolution of AI methodologies will play a crucial role in shaping the future of energy production, ensuring more sustainable, efficient, and intelligent resource management strategies in the petroleum industry.

YAPAY ZEKA ALGORİTMALARINI KULLANARAK PETROL VE GAZ ÜRETİM TAHMİNLERİNİN İYİLEŞTİRİLMESİ

ÖZET

Enerji sektörü büyük ölçüde kaynak tahsisini, üretim planlamasını ve operasyonel verimliliği optimize etmede hayati önem taşıyan doğru petrol ve gaz üretim tahminlerine dayanmaktadır. Gelecekteki üretim oranlarını tahmin etmek petrol endüstrisi için temel bir öneme sahiptir ve işletmelerin benzin ve dizel gibi yakıtların çıkarılmasını ve rafine edilmesini geliştirirken riskleri en aza indirip karları en üst düzeye çıkarmasını sağlar. Güvenilir tahmin modelleri, paydaşların yatırım stratejileri, altyapı geliştirme ve mevzuat uyumluluğu konusunda bilinçli kararlar almasına yardımcı olur. Küresel enerji talepleri artmaya devam ettikçe, petrol sektöründe istikrar ve sürdürülebilirliğin sağlanması için kesin, veriye dayalı tahmin yöntemlerine olan ihtiyaç giderek daha önemli hale geliyor.

Geleneksel olarak, enerji açısından zengin doğal gaz ve petrolün gelecekteki üretiminin tahmin edilmesi, petrol mühendisliğinde yaygın olarak çalışılan bir konudur. Benzer çıkarma süreçleri ve pazar talepleri nedeniyle bu iki kaynak sıklıkla birlikte analiz edilir. Tarihsel olarak araştırmacılar, gelecekteki üretim seviyelerini tahmin etmek için düşüş eğrisi analizi ve sayısal rezervuar simülasyonu gibi geleneksel tekniklere güvendiler. Bu yaklaşımlar yaygın olarak kullanılmasına rağmen, birçok kritik sınırlamaya sahiptirler. Yüksek hesaplama maliyetleri, simülasyonları tamamlamak için gereken uzun süre ve çoklu varsayımlara bağımlılık çoğu zaman güvenilmez ve tutarsız tahminlerle sonuçlanır. Ek olarak, bu geleneksel modeller, özellikle dalgalanan rezervuar koşulları, piyasa dinamikleri ve dış çevresel faktörlerle uğraşırken, petrol ve gaz üretiminin karmaşık, doğrusal olmayan doğasına uyum sağlamakta zorlanır. Sonuç olarak, daha gelişmiş, etkili ve doğru tahmin yöntemlerine olan ihtiyaç giderek artıyor.

Son yıllarda yapay zeka (AI), enerji sektöründe dönüştürücü bir araç olarak ortaya çıktı ve üretim tahmininin hızını, doğruluğunu ve uyarlanabilirliğini artırma potansiyeli sunuyor. Yapay zeka tabanlı yaklaşımlar, otomatik özellik seçimi, model tanıma ve model optimizasyonu yoluyla tahmin doğruluğunu artırırken tahmin için gereken hesaplama süresini önemli ölçüde azalttı. Yapay zeka odaklı modeller, makine öğrenimi, topluluk öğrenimi ve derin öğrenme tekniklerinden yararlanarak, eğilimleri belirlemek ve kesin tahminler yapmak için büyük hacimli geçmiş üretim verilerini etkili bir şekilde işleyebilir. Yapay zekanın petrol mühendisliğine entegrasyonu, daha verimli karar verme süreçlerini mümkün kıldı, kaynak kullanımını optimize etti ve üretim planlamasındaki belirsizlikleri en aza indirdi.

Bu tez, güçlü bir yapay zeka odaklı tahmin çerçevesi uygulayarak petrol ve gaz üretimi tahmininin doğruluğunu ve verimliliğini artırmayı amaçlamaktadır. Önerilen sistem, yüksek doğruluklu tahmin modelleri geliştirmek için makine öğrenimi, topluluk öğrenimi ve derin öğrenme tekniklerinin bir kombinasyonunu kullanır. Sistem, geçmiş üretim verilerinden yararlanarak geleceğe yönelik kesin sonuç tahminleri oluşturmayı, şirketlerin ve paydaşların stratejik planlamayı geliştirmesine,

kaynak tahsisini optimize etmesine ve pazar risklerini azaltmasına olanak sağlamayı amaçlıyor. Araştırma, minimum hata oranlarıyla petrol ve gaz üretimini tahmin etmek için en etkili metodolojiyi belirlemek amacıyla birden fazla yapay zeka modelinin performansını değerlendirmeye odaklanıyor.

Önerilen sistemde on bir farklı metodoloji kullanılmıştır. Bunlar: Karar Ağacı Regresörü (DTR), Rastgele Orman Regresörü (RFR), K-En Yakın Komşular (KNN), Extreme Gradient Boosting (XGBoost) ve Gradient Boosting Regressor (GBR) olmak üzere beş makine öğrenme modeli ve Bagging, Boosting ve Stacking olmak üzere üç topluluk öğrenme modelinden oluşur. Ayrıca derin öğrenmede de üç model bulunmaktadır: Yapay Sinir Ağı (ANN), Tekrarlayan Sinir Ağı (RNN) ve Uzun Kısa Süreli Bellek (LSTM). Sistem, bu metodolojileri birleştirerek veriye dayalı tahminin birçok yönünün dikkate alınmasını sağlar ve böylece sağlamlığı ve genel performansı artırır.

Bu tezin sonuçları, topluluk öğrenme tekniklerinin, özellikle de istifleme modelinin, bağımsız makine öğrenimi veya derin öğrenme modellerine kıyasla üstün tahmin performansı sunduğunu göstermektedir. Topluluk temelli yaklaşımlar, Ortalama Mutlak Hata (MAE) ve Ortalama Karesel Hata (MSE) dahil olmak üzere temel değerlendirme ölçümlerindeki hata oranlarını önemli ölçüde azalttı. Test edilen tüm modeller arasında istifleme modeli, %99'luk R-kare değeriyle en yüksek doğruluk düzeyine ulaştı; bu, petrol ve gaz üretim verilerindeki karmaşık kalıpları ve bağımlılıkları yakalama konusundaki olağanüstü yeteneğini gösteriyor. Bu bulgu, son derece doğru bir nihai tahmin üretmek için çoklu temel öğrenicilerin tahmin gücünü birleştiren bir model entegrasyon tekniği olarak istiflemenin etkinliğini vurgulamaktadır.

Akademik katkıların ötesinde, bu tezin petrol endüstrisi için önemli pratik çıkarımları vardır. Yapay zeka odaklı tahmin modellerinin petrol ve gaz üretim iş akışlarına entegrasyonu, karar verme verimliliğinde, maliyet azaltmada ve risk yönetiminde önemli iyileştirmelere yol açabilir. Sektör profesyonelleri, akıllı tahmine dayalı analitiği kullanarak veriye dayalı yatırım kararları alabilir, sondaj ve çıkarma stratejilerini optimize edebilir ve genel operasyonel performansı iyileştirebilir. Ayrıca yapay zeka tabanlı modeller, değişen rezervuar koşullarına ve dış ekonomik faktörlere daha fazla uyum sağlayarak uzun vadede daha güvenilir ve sürdürülebilir üretim tahminleri sağlar. Gerçek zamanlı üretim verilerine, dış pazar dalgalanmalarına ve çevresel kısıtlamalara dinamik olarak uyum sağlama yeteneği, yapay zeka destekli tahminleri gelecekteki enerji sektörünün kritik bir bileşeni haline getiriyor.

Yapay zeka tabanlı tahmin metodolojilerinin benimsenmesi, giderek karmaşıklaşan ve dinamik hale gelen enerji pazarında daha kesin ve güvenilir üretim planlaması sağlayarak petrol endüstrisinde devrim yaratma potansiyeline sahiptir. Ek olarak, yapay zeka ve veri bilimindeki sürekli ilerlemeler, gerçek zamanlı izleme sistemlerinin entegrasyonunu sağlayarak değişen pazar taleplerine ve çevresel koşullara yanıt verme yeteneğini geliştirecek. Son derece doğru ve uyarlanabilir tahminler üretmek, küresel enerji ihtiyaçlarının karşılanması ve petrol ve gaz sektöründe verimliliğin ve sürdürülebilirliğin desteklenmesi açısından hayati önem taşıyacak. Ayrıca, enerji sektöründe yapay zeka uygulamalarının genişletilmesi, otomasyon, gelişmiş rezervuar yönetimi ve öngörücü bakım için yollar açarak, çıkarma ve rafinaj süreçlerini daha da optimize edebilir. Gelecekteki araştırmalar, ek gerçek zamanlı operasyonel parametrelerin dahil edilmesine, veri kümesinin çeşitli

jeolojik oluşumları içerecek şekilde genişletilmesine ve tahmin performansını daha da artırmak için birden fazla metodolojinin güçlü yönlerini birleştiren hibrit yapay zeka mimarilerinin geliştirilmesine odaklanacak.

Genel olarak bu tez, yapay zeka destekli petrol ve gaz üretimi tahmini için kapsamlı bir çerçeve sunarak, son derece doğru ve verimli tahminler elde etmek için makine öğrenimi, derin öğrenme ve topluluk öğrenme tekniklerini entegre etmenin muazzam potansiyelini ortaya koyuyor. Yapay zeka metodolojilerinin sürekli gelişimi, petrol endüstrisinde daha sürdürülebilir, verimli ve akıllı kaynak yönetimi stratejileri sağlayarak enerji üretiminin geleceğini şekillendirmede önemli bir rol oynayacaktır.

xxx

1. INTRODUCTION

This chapter provides a broad introduction that begins with an overview of oil and gas production, an output projection, a brief review of some related work, and issue statements. In addition, this chapter addresses the aims of this thesis, and the thesis layout sequentially.

1.1. Overview

Oil is the primary fuel source and an essential component of many production processes. Many automobiles, airplanes, trucks, ships, machinery, and electricity must all function [1]. Oil and its derivatives have had a major impact on the world's energy supply. According to projections, this trend is expected to continue in the years to come, leading to a rise in the demand for oil relative to other energy sources [2]. The oil production process consists of a methodical sequence of steps from site exploration to extraction to product delivery to enterprises and the general public, [3].

To collect gas and oil from the subsurface for use as an energy source, a number of procedures must be carried out. In the oil and gas industry, the three main production processes are:

- Upstream sector phase: Exploration and extraction are included in the initial stages of the petroleum process. The goal of upstream operations is to locate underground areas that have the potential to produce gas and oil. Drilling, production, and preliminary exploration of oil reserves are all included in the upstream industry.
- Midstream industry phase: Crude oil is transported from producing wells to refineries and facilities for downstream processing by the midstream sector. Trains, cars, oil tankers, and pipelines are all examples of practical forms of transportation. Crude oil and natural gas are among the petroleum and natural gas commodities that are stored and distributed by the midstream industry.

- Downstream industry phase: The downstream sector's main responsibilities include separating or fractionating crude oil and carrying out the required chemical treatments. Petroleum is then combined to create a finished product that is used as fuel [4]. Additionally, downstream refineries are responsible for purifying natural gas. Oil refineries, petrochemical plants, distribution hubs, and retail stores are examples of downstream operations [4]. Figure 1.1 illustrates three phases of the primary production processes.

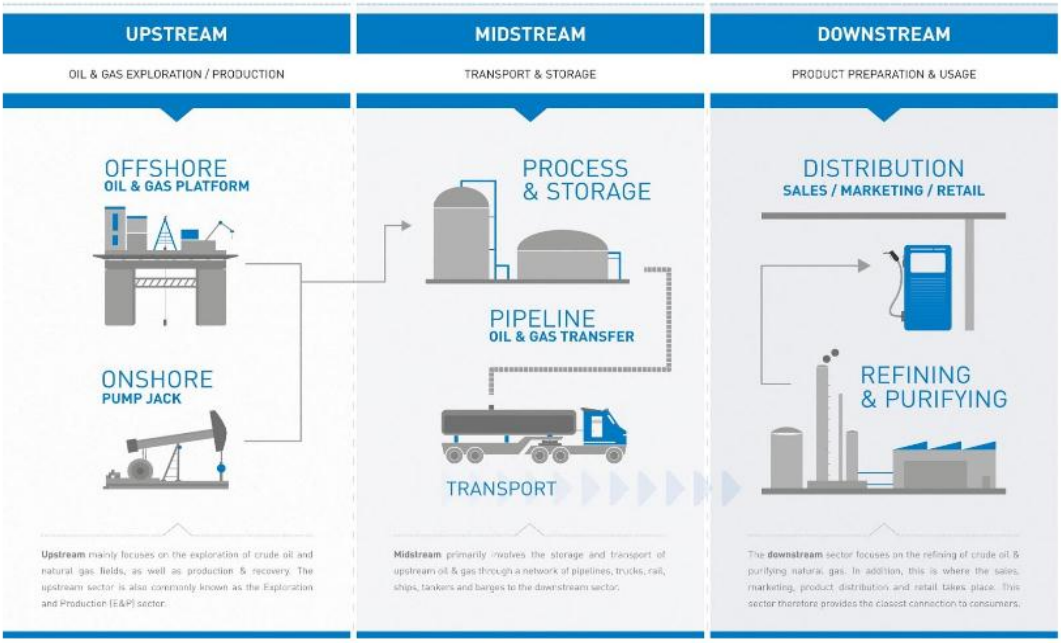

Figure 1.1. The fundamental oil and gas production processes [3].

Since it provides production data for facility capacity design, sets a drilling schedule, and evaluates economic viability, forecasting is crucial to creating a field development plan. A production calculation dependent on previous performance data from both effective and inactivity wells [5]. For governmental and organizational entities to create the appropriate economic plans, production forecasting is a crucial process [6]. A sophisticated numerical modeling of reservoirs and related engineering studies are required for the oil and gas industry's predicted output [7]. Accurate forecasting is an important and unique endeavor aimed at managing and enhancing oil resources. The petroleum industry uses a number of traditional techniques, such as Decline Curve Analysis (DCA) and Numerical Reservoir Simulation (NRS) [8]. These models, which were impacted by reservoir complexity and field size, took a significant amount of time and helped with decision-making [9]. For a long time, the NRS method has been used to forecast oil

production. However, NRS models have significant drawbacks, such as labor-intensive requirements, a large number of dynamic model parameters, the need for an accurate static model, and their time-consuming nature [10]. A common method for predicting future oil and gas production is DCA [11]. The simulated cumulative production rate was fitted to determine the DCA model's parameters [12]. Using historical production data, DCA seeks to forecast future well or field output [13]. Artificial Intelligence (AI) systems trained on pertinent data from the field can substantially aid in expediting asset appraisal while ensuring it remains independent of expert influence [14]. The sectors of oil and gas are changing as a result of machine learning and deep learning. Potential areas for future AI development in the petroleum industry are represented by professional software platforms [15]. In order to provide intelligent and trustworthy solutions to complex and nonlinear problems, the field of computer science known as artificial intelligence combines human intellect with computational capability. Computers with artificial intelligence can reason and make decisions on their own [16]. Increased computer processing speeds made effective AI techniques possible in the early 21st century, allowing for accurate oil field output forecasts. A somewhat good approximation of the problem and analytics can be obtained by training machine learning on large historical datasets [17]. Many academics have expressed a great deal of interest in the use of machine learning [18]. The input and the anticipated outcome can be correlated using machine learning techniques. The physical behavior of the system is not altered in any way by the machine learning technique [19]. The availability of large datasets and high-performance computing resources has made machine learning methodology a feasible approach for modeling manufacturing processes [20]. The creation of computer algorithms that improve performance by learning from data or previous examples is another definition of machine learning [21]. For rapid assessment and production forecasting in the oil sector and petroleum business, scholars have recently used machine learning and deep learning techniques [22]. More accurate results can be obtained faster when predictive and inferential data analytics are included into machine learning and deep learning procedures. Data analytics is being used quickly by the oil and gas sector to enhance the calibre of its decision-making procedures [23].

1.2. Problem Statement

Researchers have frequently used simulation tools like DCA and NRS to project oil and gas output. There were some difficulties with these programs. Uncertainty over production dynamics, well performance, and reservoir behavior can have a big impact on how accurate DCA-based predictions are. The approach involves doing accurate estimations and developing robust decline curve models for these uncertainties. Large sets of equations must be numerically solved for NRS reservoir simulation, which can be time-consuming and computationally taxing. Effective simulation execution often requires access to high-performance computing resources, which poses challenges for smaller businesses or research institutions with limited funding. Data constraints and insufficient input: Accurate and thorough input data are crucial for both DCA and NRS. However, obtaining current and reliable data could be difficult. Inadequate information or incorrect input parameters might result in inaccurate forecasts and judgments. The results of DCA and NRS depend on intricate equations that could lead to challenging simulations and derivations. Traditional methods of extracting oil and gas are laborious, sometimes taking hours or days, and their accuracy is unpredictable.

1.3. Objectives

The thesis aims to validate the oil and gas output's optimistic forecast. Predicting oil and gas production is important for many reasons, including:

- Better Production Optimization: To optimize production operations, artificial intelligence algorithms can review a massive amount of data from several sources, consisting of sensors, production records, and geological surveys. Higher recovery rates and more efficiency from already-existing wells may result from this.
- Predictive maintenance: Machine learning models can anticipate equipment breakdowns before them happen by analysing past data on equipment performance. This lowers operating costs, minimizes downtime, and enables proactive maintenance.
- Reservoir Characterization and Modeling: Accurate reservoir models can be created by analyzing production data, well logs, and seismic data. These

models make it possible to comprehend the characteristics and behavior of reservoirs, which helps with well site optimization and recovery strategy improvement.

- Reducing Uncertainty in Exploration: Artificial intelligence algorithms are capable of analyzing geological data to more accurately identify possible drilling locations. This raises the success rate of drilling operations and lowers the risk involved in exploration.
- Cost Reduction: Drilling, production, and transportation are just a few of the phases of oil and gas production where cost reductions can be achieved by automating repetitive operations and streamlining procedures using machine learning and deep learning models.
- Improved Decision Making: Machine learning and deep learning algorithms facilitate improved decision-making processes at all levels of oil and gas operations, from asset maintenance to reservoir management, by offering insights from enormous volumes of data.
- Adaptation to Changing Conditions: Oil and gas businesses can modify their production plans in response to shifting market conditions, legal regulations, or environmental considerations because machine learning models are able to continuously learn from fresh data.
- The overarching goals of applying artificial intelligence algorithms to forecast oil and gas production are to enhance productivity, cut expenses, lower risks, and optimize the extraction and use of hydrocarbon resources.
- The relationship between time and accuracy in machine learning and deep learning models is essential. Generally, achieving higher accuracy requires more complex models, which in turn demands more computational time. However, advancements in artificial intelligence (AI) techniques are focused on optimizing this balance, enabling precise predictions in shorter timeframes. Figure 1.2 shows the time with accuracy by using machine learning and deep learning.

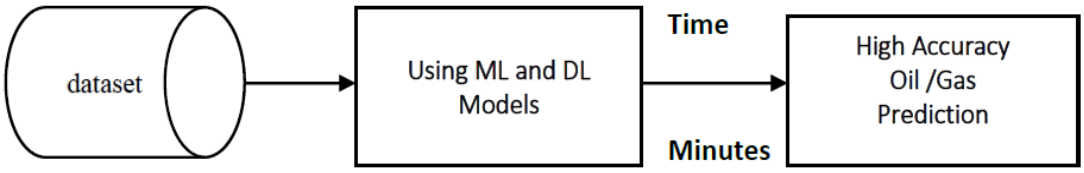

Figure 1.2. Oil and gas production prediction using ML and DL.

1.4. Literature Review

A long with ensemble learning approaches, machine learning and deep learning have become increasingly potent instruments for forecasting oil and gas output in recent years, demonstrating their potential across various formations and datasets. For example, Kim et al. used a variety of models for predicting cumulative gas production (CGP) in the Montney Formation (Canada), including ANN, 1D convolutional neural networks (1D-CNN), LSTM networks, and hybrid 1D-CNN-LSTM models. The hybrid model showed better performance in enhancing gas production projections by utilizing well information, fracture treatment parameters, and early production data [24].

Similarly, using data from the Volve field in Norwegian, Zanjani et al. assessed the effectiveness of ANN, LR, and SVR. Even though ANN performed better than the other models in forecasting the production of hydrocarbons from well NO159-F-1C, the study made clear how crucial it is to customize model selection for particular datasets because no single algorithm is always better [25].

Tan et al. expanded on machine learning techniques by using six algorithms to forecast production in the WY shale gas block: RF, BP, SVR, XGBoost, LightGBM, and multivariable linear regression (MLR). With an R^2 value of 0.87, XGBoost resulted as the most effective model, with the study's limitations due to its small dataset from a single location. [26].

On the other hand, Hui et al. used four different approaches linear regression, neural networks, XGBoost, and the extra trees to evaluate shale gas output in the Fox Creek region. The results showed that extra trees had the highest accuracy R^2 equal to 0.809 [27].

DTR, SVR, MLR, XGBoost, PLR, RFR, RNN, and ANN are the eight models that were investigated in Saudi Arabia by N. M. Ibrahim et al. The dataset, which was

supplied by Saudi Aramco, indicated that ANN, XGBoost, and RNN produced the best results, with R^2 values for water, gas, and oil being 0.9627, 0.9012, and 0.926, respectively [28].

Furthermore, utilizing historical production data from an oil field in Southern Vietnam, Lan Mai-Cao et al. assessed four deep neural networks (multilayer perceptron, CNN, LSTM, and gated recurrent unit) in addition to four traditional machine learning models (RF, SVR, KNN, and GB). According to the study, the SVR model was a good option for predicting petroleum output since it provided a compromise between high accuracy and computational efficiency [29].

Al-Aghbari et al. created a DL model that combines temporal convolutional networks (TCN) and LSTM to estimate oil output in a single step, lowering computational complexity while increasing accuracy, in order to meet the difficulties of traditional reservoirs [30]. On the other hand, Liu et al. developed regression and classification models for reservoir identification and production prediction using a dataset from CNPC, and XGBoost outperformed ANN in terms of speed and accuracy [31].

For time-series data, S. Hosseini et.al. Proposed a hybrid LSTM-1D CNN model for predicting oil production in the Volve field, with the LSTM model achieving an R^2 score of 0.98. While promising, the study emphasized the need for further investigation into the model's generalizability across different wells [32].

In another study, Song et al. used a range of machine learning models, such as LR, XGBoost, LightGBM, BP, Neural Network, and LSTM, to thoroughly analyze the productivity of 394 offshore oil wells in China. Because of its exceptional stability and capacity for generalization across several datasets, XGBoost was the model with the best performance. LightGBM, on the other hand, showed overfitting problems, highlighting how crucial it is to choose machine learning algorithms. The study emphasizes that, especially in intricate offshore situations, the model selection can have a substantial impact on the accuracy and dependability of production forecasts [33].

Furthermore, in order to anticipate and optimize rate of penetration (ROP) during gas well drilling in Xinjiang, China, Liu et al. suggested a layered generalization ensemble model. SVR, RF, Extremely Randomized Trees (ET), GB, LightGBM, and

XGBoost are the six ML methods that were merged into this stacked ensemble model. The stacked generalization model fared better than any of the individual models, obtaining an R^2 of 0.9568 on the test set, however the ET model showed the best individual performance with an R^2 of 0.9268. In complicated activities like drilling, where high predicted accuracy is essential for maximizing operational efficiency, the study successfully illustrated the increased predictive capability of merging different models [34].

Similarly, F. Ye et al. using a dataset that contained geology, reservoir, engineering, and production data from over 200 wells in gas field A and 207 wells in shale gas field B. The study used a blending ensemble learning model in conjunction with a number of machine learning algorithms, such as RF, Extremely Randomized Trees (ET), LightGBM, and GBR. The blending model outperformed individual models in terms of predicted accuracy and generality, especially for shale gas output. The effectiveness of combining several models to improve prediction accuracy in complex reservoir settings was highlighted by the ensemble learning approach's remarkable R^2 of 0.9524 for shale gas production [35].

Lastly, Y. Hou, the dataset utilized in this research includes production records from 242 Chinese shale gas wells. LightGBM, RFR, SVM, and stacking model were the models employed. According to the study, the stacking model with feature weighting performed better than separate models, offering improved forecasting accuracy and generality for shale gas output. Outperformed individual models and obtained the highest prediction accuracy with an R^2 value of 0.92 on test data [36].

These studies demonstrate the versatility and potential of artificial intelligence algorithms in enhancing production forecasts for various oil and gas fields. The findings highlight the importance of selecting models tailored to specific data and field conditions to maximize prediction accuracy.

1.5. Organization

There are five chapters in this thesis, which are arranged as follows:

- **Chapter One.** General Introduction: This chapter reviews the scientific background and forecasts oil and gas output, outlining the primary steps of the extraction process briefly. Additionally, predictions for oil and gas results

use traditional techniques, related works, and problem statements. We propose solutions utilizing ensemble models that combine various ML and DL models for production forecasting. It offers a brief overview of related studies.

- **Chapter Two.** Theoretical Background: This chapter provides theoretical background about ML, Ensemble Learning: Bagging, Boosting, and Stacking models, and DL. Such as: RFR, DTR, KNN, XGBoost, GBR, ANN, RNN, and LSTM. In addition, the evaluating metrics and activation functions used in this thesis are illustrated.
- **Chapter Three.** Methodology of the proposed system: This chapter defines the methodology for predicting oil and gas production, as well as the design and execution of the proposed system. The system consists of stages and algorithms. Initially, we describe the dataset. Subsequently, forecast oil and gas output utilizing eleven various techniques. To enhance model performance through the selection of effective parameters, with particular emphasis on identifying suitable activation functions for deep learning models.
- **Chapter Four.** Experimental Results and Discussion: This chapter explains the results of ML, ensemble learning, and DL models. They are compared to determine the effectiveness of the model with the results. The comparison and examination of these models are presented to provide a comprehensive understanding of their performance.
- **Chapter Five.** Conclusions and Future Works: This chapter offers a concise summary of the thesis and some recommendations for future works.

Each chapter is logically sequenced to build upon the insights gained from the preceding sections, ensuring a coherent flow of information throughout the thesis. This structured approach facilitates a comprehensive understanding of the integration and challenges of artificial intelligence algorithms within a data set for oil and gas production for prediction, thereby enhancing the academic rigor of the study.

2. THEORETICAL BACKGROUND

2.1. Introduction

Programming robots to mimic human reasoning and comprehension, including understanding, learning, and adaptation, was proposed by a group of computer scientists in 1956 [37]. Human ingenuity led to the creation of several machines. By simplifying manufacturing, computing, and transport, these devices improved human life [38]. Recent years have seen a significant increase in machine learning, primarily due to improvements in computing power and storage capacity. This development has enhanced computers' ability to predict future events using information and data from the past. [39]. In terms of technological advancement, artificial intelligence (AI) represents a significant turning point in human history [40].

2.2. Machine Learning (ML) Models

The goal of machine learning, a branch of artificial intelligence (AI), is to use statistical models constructed with algorithms and big datasets to solve real-world issues [41]. Conventional programming entails manually constructing a program and outlining the procedures required to address an issue. Rather than writing the steps to solve the problem, ML is inspired by human learning. The relationship between ML and deep learning is depicted in Figure 2.1. DL is a part of ML, which is a part of AI [42]. A branch of artificial intelligence called machine learning has gained a lot of attention lately and is now a crucial component of many digitalization solutions [43]. In circumstances that typically call for extensive human tuning or lengthy lists of rules, ML can frequently improve efficiency while simplifying code when compared to traditional approaches. The most sophisticated machine learning techniques can resolve issues for which there is no viable answer using traditional techniques [44].

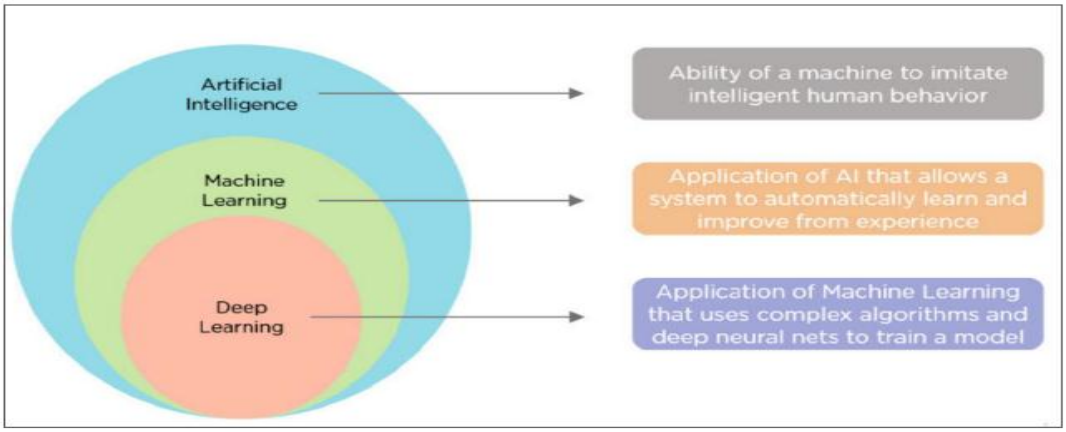

Figure 2.1. The relation between AI, ML, and DL [42].

Like humans, machines can learn and grow via trial and error with the aid of machine learning. Prediction, categorization, extraction, and decision-making from a given collection of data are the main tasks of machine learning [45]. Building successful models in a variety of application areas can benefit greatly from the use of several types of ML [46]. Three main categories of machine learning are depicted in Figure 2.2. To have a rough idea of the kinds of problems that machine learning can solve, it is helpful to characterize the three primary groups of learning problems supervised learning, unsupervised learning, and reinforcement learning [47].

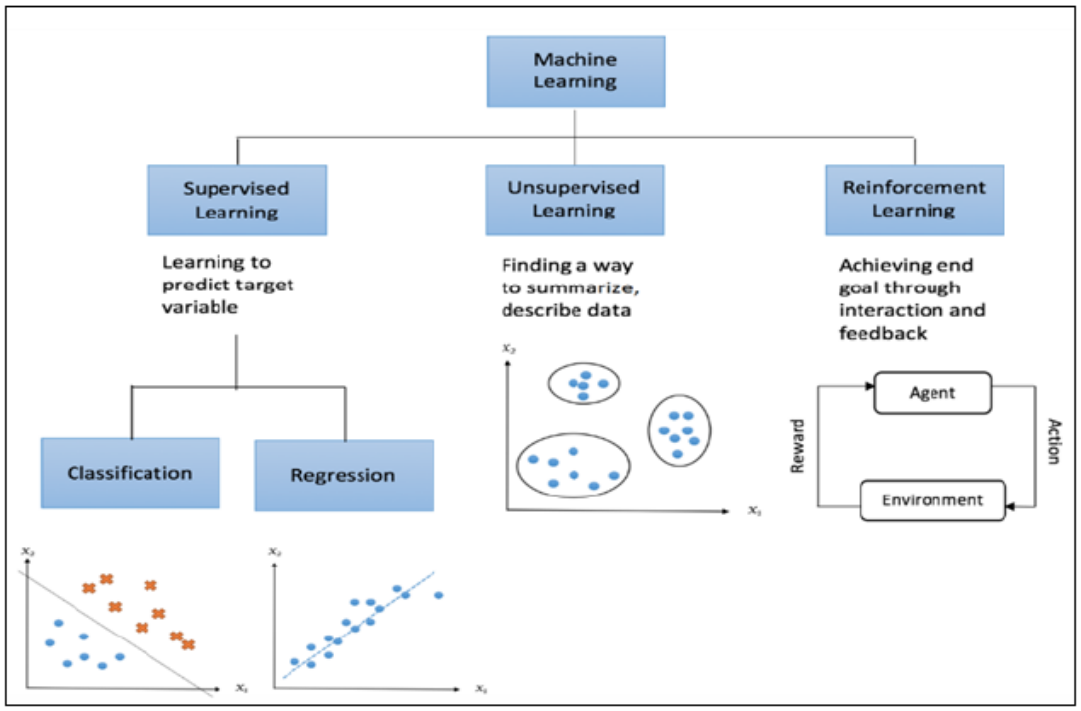

Figure 2.2. Types of ML models [48].

- Supervised learning is a ML technique that trains a model on a labeled dataset to predict a target variable. There are two main types of this process: regression and classification. Using predictor characteristics, supervised learning seeks to maximize category label models. The problem is referred to as a regression problem when the output is continuous. When the output variable is continuous, the regression technique is used [49]. Human-defined categories will be manually applied to a subset of data using supervised algorithms. The ML algorithm's tasks include identifying patterns and creating models [50]. Regression algorithms are frequently used to solve problems involving the generation of numerical and continuous outcomes. These are usually used to deal with problems related to questions like what number or what quantity. There are two sorts of classification problems, the more common being True/False and Yes/No [51].
- Unsupervised learning is utilized when the available data consists of input variables without any associated output variables. These algorithms are designed to capture the inherent patterns within the data, enabling a deeper understanding of its properties. Clustering is considered to be one of the primary categories of unsupervised algorithms. This methodology involves the identification of intrinsic clusters within the dataset, which are subsequently utilized to forecast outcomes for unobserved inputs [52].
- Reinforcement learning this particular approach belongs to ML, where in the algorithm is trained through a system of rewards and punishments. The algorithm involves the agent acquiring knowledge from the surrounding environment. When an agent executes its tasks accurately, it receives rewards, whereas when it does them inaccurately, it incurs penalties. The process of agent learning involves the optimization of reward maximization and penalty minimization [53].
- We will give a thorough description of the implementation of several machine learning methods used to forecast oil and gas production. These techniques include Decision Tree Regression (DTR), Random Forest Regression (RFR), K-Nearest Neighbors (KNN), Extreme Gradient Boosting (XGBoost), and Gradient Boosting Regressor (GBR).

2.2.1. The decision tree regression (DTR) model

Decision tree regression is a successful ML technique for classification and regression issues. There are two sections to the original data. First, a tree is built using the training data. Using a binary split approach comes in second. On the newly formed branches, the separation process is carried out until a branch is no longer separable, at which time the neighboring node reaches the minimum size and becomes a terminal node [54]. Mean Square Error (MSE) is used by DTR to subdivide nodes. The method uses a binary tree to partition the data and choose the value. MSE can be used individually to evaluate each subgroup. A minimal MSE value is chosen by the tree. Equations 2.1 show the relationship between the MSE for nodes n with N observations and $\bar{y}$ (predicted value) as the expected value:

$$E(n) = \frac{1}{N}\sum_{i=1}^{N}[y_i(n) - \bar{y}(n)]^2 \quad (2.1)$$

Figure 2.3 shows DTR's construction. It demonstrates that the root node, which can be further subdivided into other nodes, is the main node that indicates the entire sample. The inside nodes show the characteristics of a data collection, while the branches represent the decision criteria. The outcome is shown by the Leaf Nodes [55].

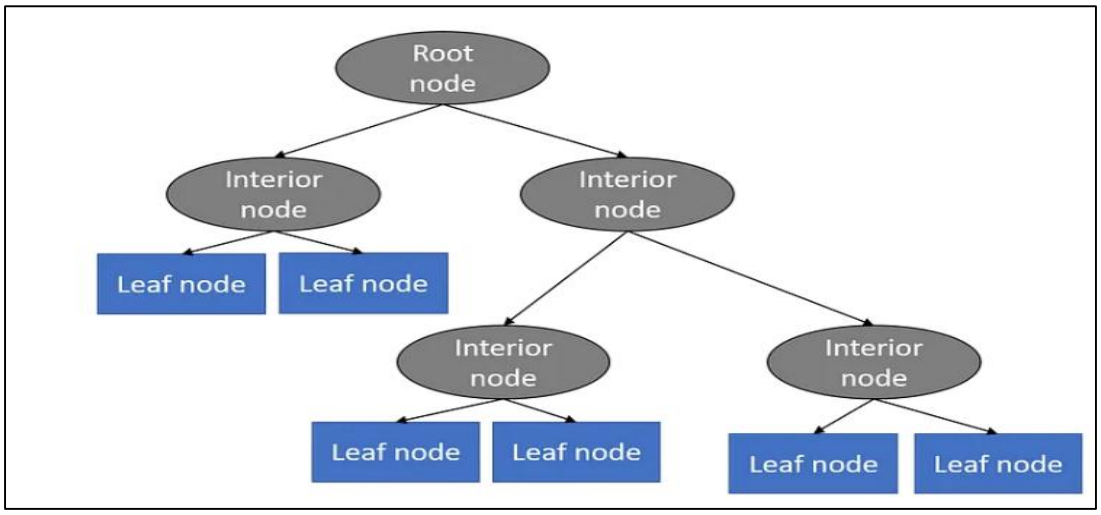

Figure 2.3. The structure of a DTR model [55].

2.2.2. The random forest regression (RFR) model

Regression and classification issues have been effectively handled by the random forest algorithm [56]. Professor Leo Breiman of the University of California,

Berkeley originally presented it in 2001. This technique is also known as random decision forests [57]. It begins with input data, trains many of models, gathers predictions from each model, and then votes to determine which model is best. This method depends on the DT; the average of the results from many DTR is used to determine the final forecast. To determine the forecast outcome, the average of each tree's outputs is computed [56]. Depending on how many trees we want, the input is divided into many samples. An easy prediction model is then built inside each segment, and the output of these models is then combined using a bagging technique to provide the final forecast [58]. Figure 2.4 shows the architecture for a Random Forest Regression.

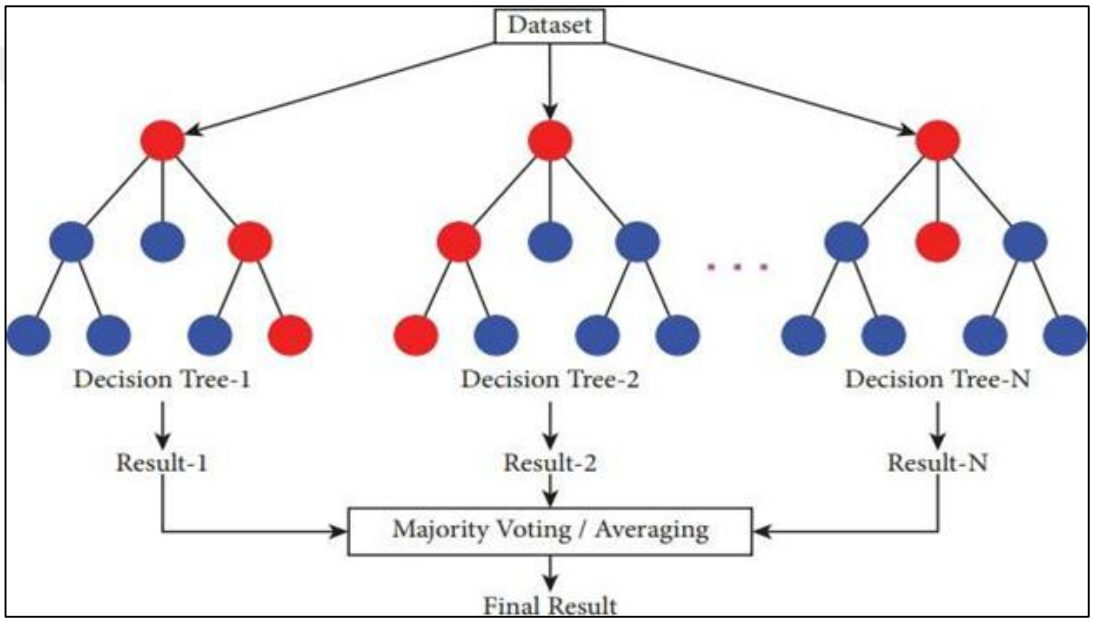

Figure 2.4. Random forest regression architecture [59].

RFR is a technique that uses the idea of collective learning to merge many trees into a single model [60]. Important RFR model parameters include the minimum number of samples at a node with a leaf and this mean the node size, the number of predictor variables to be tested at each splitting node, predicted numbers of trees N, and the size of the node. [61]. In a regression problem, a dependent variable is estimated using a Random Forest of simple trees. The method generates K separate regression trees $h_k(x)$ based on the input variable x. Equation 2.2 shows the model's forecast based on the average of each forest tree's predictions for the inputs (k = 1,.., K). A technique known as bootstrapping is used to increase the diversity of the trees in

order to reduce the likelihood that their combined results may be associated with those of other trees [62].

$$\text{RFR prediction} = \frac{1}{k} \sum_{k=1}^{k} h_k\,(x) \tag{2.2}$$

To prevent trees from correlating with one another, enhance their diversity by training them with separate subsets of the training data using bootstrap aggregating or bagging [63].

2.2.3. The K-Nearest neighbors (KNN) model

KNN is a widespread technique; it is easy to use effectively for recognition and classification [64]. The KNN technique is widely used in the field of ML by comparing values to the nearest neighbors [65]. The primary idea behind the closest neighbors is to place an object into a group with the most similar characteristics to its surrounding elements [66]. The value of an object is determined by the mean values of the K closest neighbor. Assigning weights to the contributions of neighboring entities can prove to be advantageous, especially in cases where closer neighbors are deemed to have a more significant impact on the overall average than those situated farther away [67]. The KNN method prevents overfitting by adjusting a parameter K, whose value is proportional to the error rate [68]. Equation 2.3 describes the most popular distance function. The Euclidean distance space between (x, y) is used in the KNN model and m is the number of dimensions in the space.

$$Dis\ (\text{A,B}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)\text{^}2} \tag{2.3}$$

where, A = x_1, x_2,……, x_m, and B = y_1, y_2,....., y_m [64]. Figure 2.5 shows an example of a K -Nearest Neighbors model.

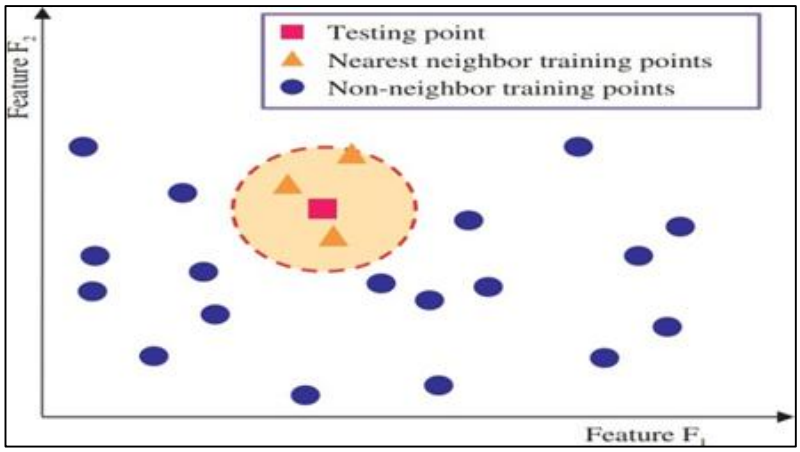

Figure 2.5. Example of a K-Nearest neighbors model [64].

2.2.4. The extreme gradient boosting (XGBoost) model

For both regression and classification issues, XGBoost produces a decision tree-style predictive model [69]. XGBoost regression is an acronym for extreme gradient boost regression [70]. XGBoost, created by Chen and Guestrin in 2016, is an effective approach for solving machine learning problems among the several tree gradient boosting implementations [71]. One boosting technique that falls within the category of supervised learning is XGBoost. Gradient-boosted trees serve as the foundation for this hybrid methodology. The boosting technique aims to train successive learners by employing modified iterations of the training samples that have been enhanced based on the previous learner's training outcomes [72]. Compared to other machine learning technologies, XGBoost performs better and learns faster [73]. Figure 2.6 shows the XGBoost topology, which consists of internal nodes, branches, several root nodes, and leaf nodes.

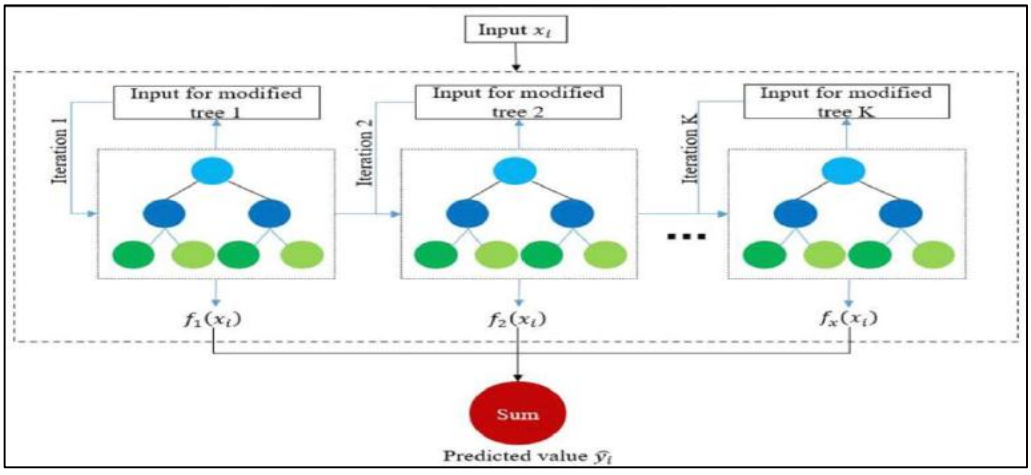

Figure 2.6. The structure of XGBoost [74].

Equation 2.4 demonstrates the estimated score $\hat{y}_i$ which is obtained by summing all f_k values.

$$\hat{y}_i = \propto \sum_{k=1}^{k} f_k \ (x_i) \tag{2.4}$$

where by categorization and regression trees (CARTs) are used to make initial decisions using the *i-th* parameter, x_i. Future decisions are indicated by branch points, whilst internal nodes make subsequent decisions. For XGBoost model prediction, leaf nodes combine single CART model predictions. The actual value of the experiment is represented by the XGBoost's mathematical term, yi. Indicates the expected value $\hat{y}_i$ linked to the input xi in this situation. The learning rate of each regression tree is indicated by the symbol a, and the total number of regression trees (CARTs) in use is indicated by the sign K. Furthermore, the output of the k-th regression tree is denoted by fk [74].

2.2.5. The gradient boosting regressor (GBR) model

Gradient Boosting Regressors combines a collection of weak learners, usually decision trees, to produce an effective prediction model [75]. The fundamental principle of gradient boosting is to increase the overall prediction accuracy by successively fitting new models to the residual errors produced by previous models [76]. Figure 2.7 shows the topology of Gradient Boosting Regressor. The model is effective in handling non-linear relationships and capturing complex patterns in data due to its sequential error-correction approach. This makes GBR very powerful in terms of accuracy and flexibility.

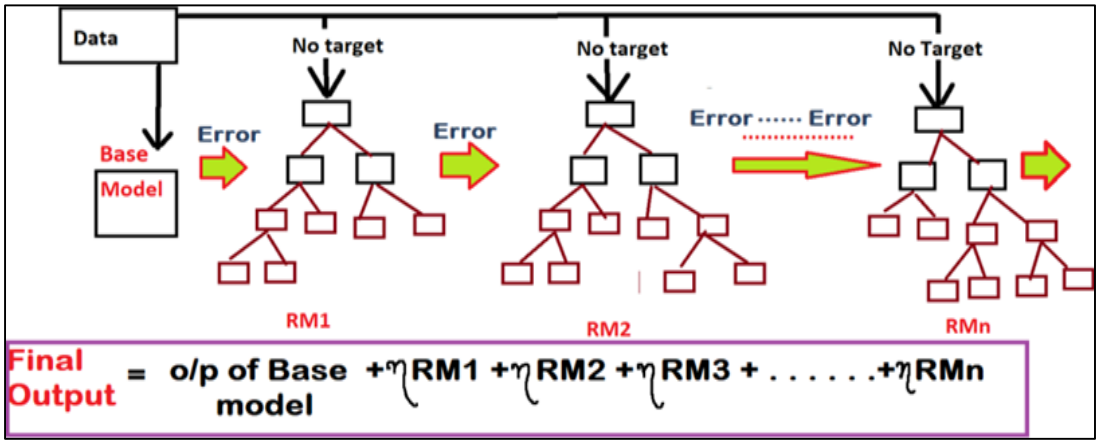

Figure 2.7. Gradient boosting explained for regression [76].

Equation 2.5 shows the GBR's mathematical formulation:

$$f_m\,(x) = f_{m-1}\;\;(x) + y\,.\,h_m\;\;(x) \tag{2.5}$$

where the $f_m\,(x)$ is the final prediction at iteration m. and $f_{m-1}(x)$ is the ensemble prediction from the previous iteration (m-1). The symbol $h_m\;\;(x)$ is the new model (or weak learner) that was trained to forecast the residuals of the old model are represented by the symbol. The last factor y is the learning rate, which regulates how much each weak learner contributes to the finished model [77].

2.3. Ensemble Learning Techniques (EL)

Ensemble learning improves the effectiveness of machine learning models. The idea behind it is simple. To create a more accurate model, several machine learning models are integrated. The three most widely used ensemble learning strategies are stacking, boosting, and bagging. Ensemble learning aims to increase prediction resilience overall, decrease errors, and enhance performance on a variety of tasks [81].

2.3.1. The bagging model

Bagging, known as Bootstrap Aggregation, is an ensemble learning strategy that enhances a machine-learning model's prediction performance by combining the advantages of bootstrapping and aggregation. In bagging, we initially extract equal-sized subsets of data from a dataset using bootstrapping, which involves sampling with replacement. Subsequently, we employ those subsets to independently train multiple weak models [78]. A weak model has low predictive accuracy. Figure 2.8 shows the structure of the bagging model. Here M1, M2, and M3 represent training distinct models on each data subset.

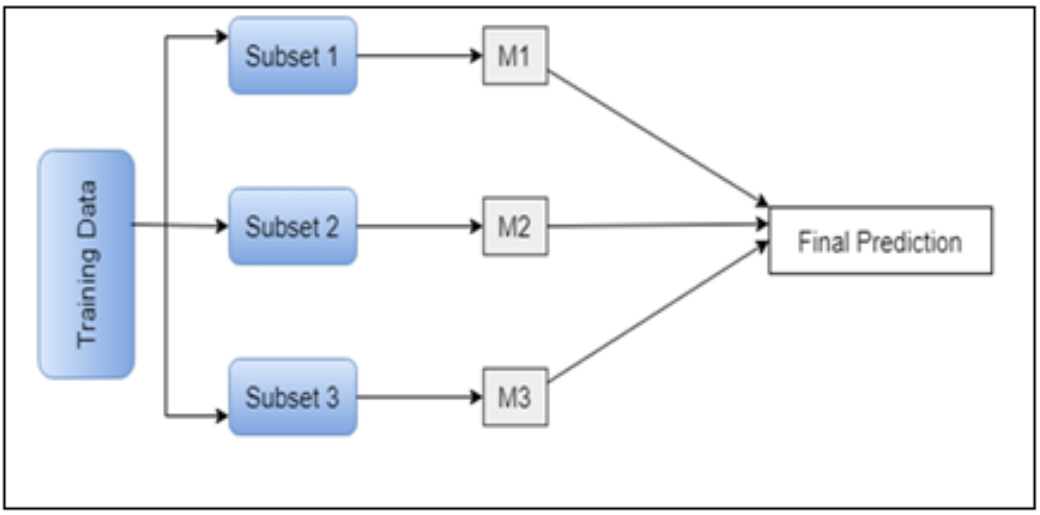

Figure 2.8. The structure of a bagging model [78].

2.3.2. The boosting model

Boosting is a supervised machine learning technique creates a strong ensemble model by combining the predictions of multiple weak models, or base models. The boosting model focuses on sequentially training the base models to decide on misclassified samples from previous iterations. Figure 2.9 shows the topology of a boosting model [79]. Boosting requires the specification of a weak model (such as regression or shallow decision trees) and subsequently enhances its performance [80]. In this thesis, we propose to combine two models, Extreme Gradient Boosting (XGBoost) and Gradient Boosting Regressor (GBR), to build a boosting model and improve the prediction of oil and gas production.

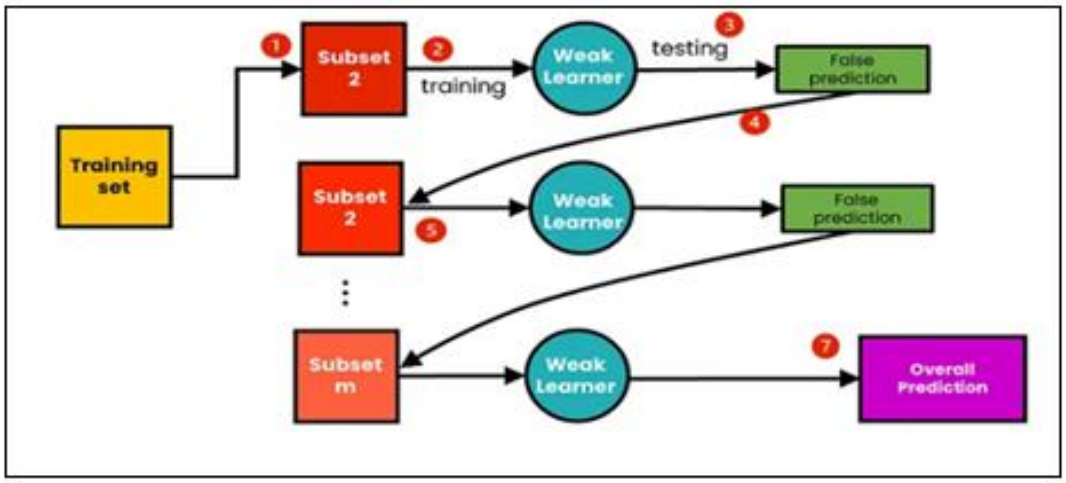

Figure 2.9. The topology of a boosting model [79].

2.3.3. The stacking model

The stacking model is an advanced ensemble learning method that merges multiple models to enhance prediction performance. The first crucial stage in the process is training several base models, also known as level-0 models, using the same dataset. The second is using the basic models' predictions to train a meta-model level-1. The meta-model learns to combine these outputs to increase the prediction's overall accuracy using the basic models' predictions as input attributes. Figure 2.10 shows the scheme of a stacking model [81]. Three machine learning techniques are being integrated to implement stacking in the framework of this thesis. The dataset will be processed independently by each base model, producing predictions that the meta-model will employ.

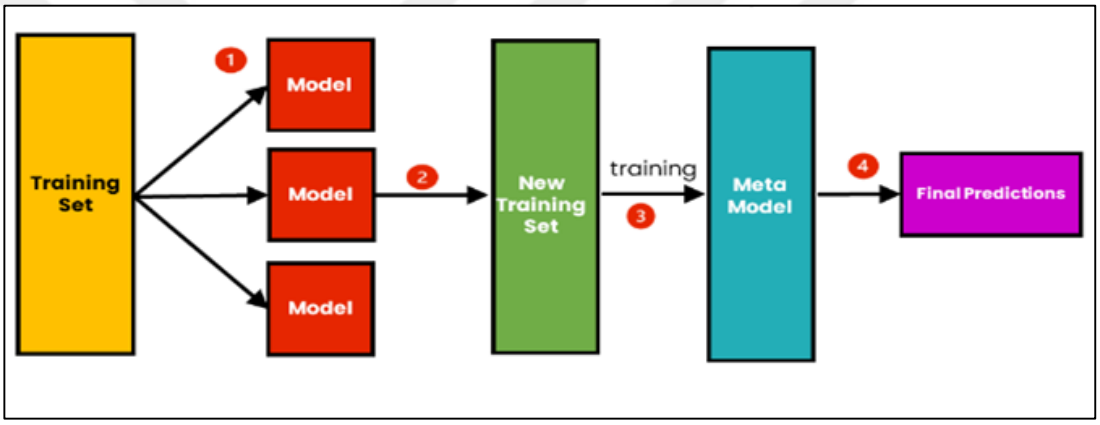

Figure 2.10. The structure of a stacking model [81].

2.4. Deep Learning (DL) Models

Deep learning is a unique type of machine learning. It works like machine learning, but with different capabilities, methodologies, and the difficulty of the problem [82]. Deep learning is a type of ML that uses ANN. Deep learning models often outperforms more conventional data analysis methods and ML models [83]. DL refers to the idea of building representations up in layers. The depth of a model is the number of layers contributing to the model of the data. In the field of DL, the acquisition of layer representations is typically achieved through the utilization of neural networks, which are structured as literal layers. The term Neural Network draws its inspiration from neurobiology; however, it is important to note that certain fundamental principles in Deep Learning have been influenced by our knowledge of the brain [84]. DL employs representation learning, which is also referred to as

feature learning, to establish a mapping between input features (comparable to prediction variables in traditional statistics) and an output. The process of mapping takes place within interconnected layers, each consisting of multiple neurons. Every individual neuron functions as a computational unit with mathematical capabilities. When all neurons are integrated, their collective purpose is to acquire knowledge about the correlation between the input characteristics and the resulting output [85]. Figure 2.11 shows a computational neuron that combines summation and activation functions.

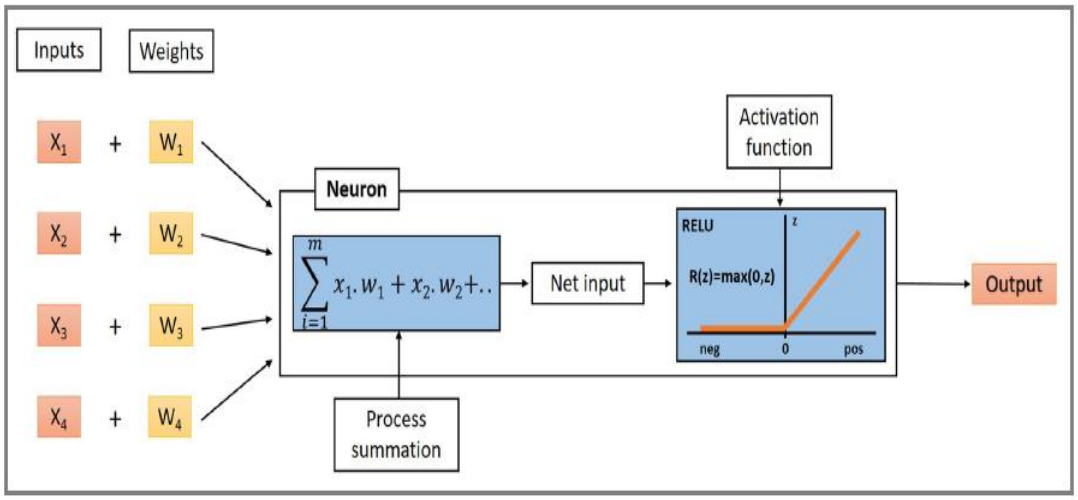

Figure 2.11. The deep learning structure [85].

Each neuron performs a weighted addition of input values before sending the resulting sum to an activation function. Changing the activation function changes the neuron's output. For the neuron in the equation, the output value is zero if the activation input is zero or less, and it is proportional to the input value if the total is greater than zero. Deep learning works by continually improving the weight values during backpropagation until the combination of values that accurately correlates the inputs with the main output has been found. [85].

2.4.1. The activation function

In a neural network, the most fundamental units are the inputs, which undergo processing and transformation to produce a final result known as the unit's activation via a scalar-to-scalar transformation known as the activation function, threshold function, or transfer function [86]. Activation functions come in various varieties, including Sigmoid, Tanh, and ReLU.

2.4.1.1. The sigmoid activation function

The activation function is frequently used since it is a non-linear function. Equation 2.6 shows the sigmoid function mappings with output values ranging from 0 to 1.

$$f(x) = \frac{1}{e^{-x}} \tag{2.6}$$

Additionally, all of the neural output values will have the same sign because the sigmoid function is not symmetric around zero. Increasing the size of the sigmoid function will fix this problem. Figure 2.12 shows the sigmoid function is an S-shaped that is continuously differentiable [86]. Equation 2.7 shows the sigmoid function's for the S-shaped function.

$$f(x) = 1 - sigmoid\ (x) \tag{2.7}$$

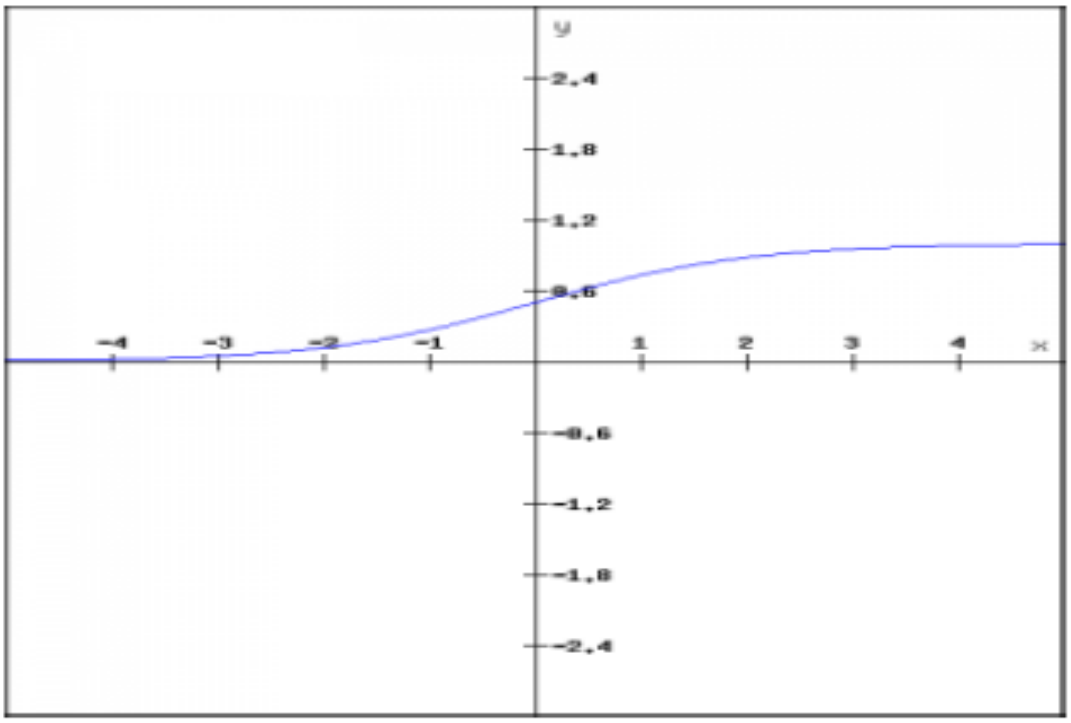

Figure 2.12. The sigmoid activation function [86].

2.4.1.2. The hyperbolic tangent function (Tanh)

Hyperbolic tangent function also known as tanh function. In contrast to the sigmoid function, the tanh function is symmetric about the origin. This causes outputs from lower layers to have opposite indications from inputs in higher layers. As shown in Figure 2.13 the values of the differentiable and continuous tanh function are between

-1 and 1. The tanh function is preferred over the sigmoid function [86]. Equation 2.8 shows the hyperbolic tangent function where the output values range between -1 and 1.

$$(x) = 2sigmoid\ (2x) - 1 \qquad (2.8)$$

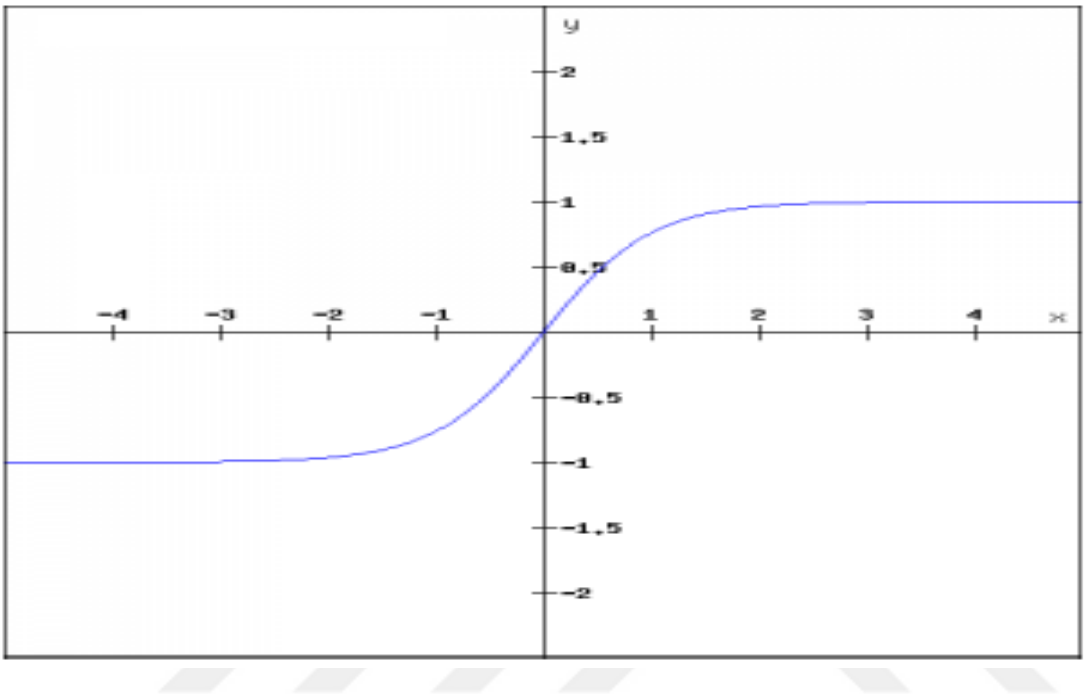

Figure 2.13. The Hyperbolic tangent function [86].

2.4.1.3. The rectifier linear unit function (ReLU)

ReLU is a rectified linear unit. ReLU has many qualities [87]. It fixes the issue of the vanishing gradient without a boundary in a minimum of one direction and this allows for sparse encoding because the number of actively engaged neurons is typically quite small. A higher resistance to noise and greater invariance to small changes in input outline the advantages of sparsity [88]. As shown in Figure 2.14 one for positive inputs and zero for negative ones defines the gradient while a zero gradient seems to be problematic at first, it has the potential to improve network sparsity by preserving important connections [89]. Equations 2.9 and 2.10 show the rectified linear unit function where the output values range between 0 and ∞.

$$f(\mathrm{x}) = max\ (0, \mathrm{x}) = \mathrm{x} \qquad \text{where } \mathrm{x} >= 0 \qquad (2.9)$$

$$f(x) = max\ (0, x) = 0 \qquad \text{where } \mathrm{x} < 0 \qquad (2.10)$$

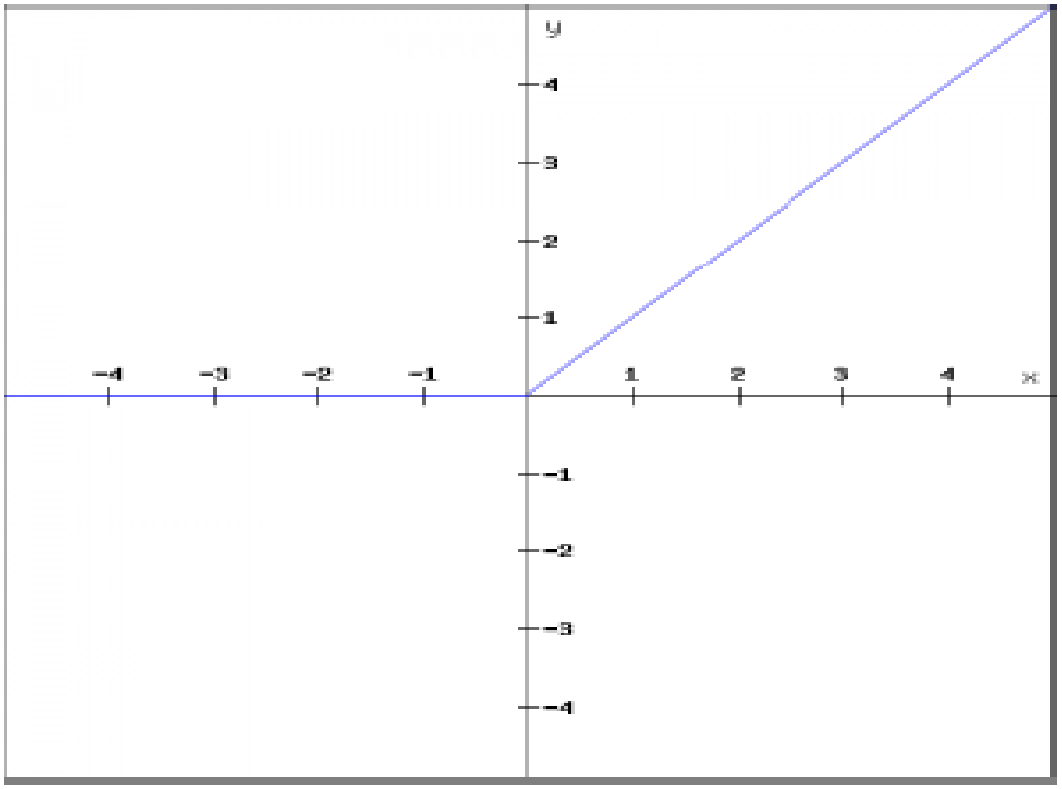

Figure 2.14. The Rectifier linear unit structure [86].

Here, we will explain various deep learning models for predicting oil and gas production. These methodologies include Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM). Each model will be examined on its theoretical basis and implementation.

2.4.2. Artificial neural networks (ANN)

Neural networks are a kind of machine learning that draws inspiration from the architecture and operation of the human brain. The architecture the ANN consists of:

- The input values must be received by the input layer.
- A group of neurons located between a neural network's input and output layers are referred to as hidden layers. From one layer to several layers, the number of layers might change.
- A neural network's output layer is in charge of generating its final output [90].
- Figure 2.15 shows the operation of an ANN consists of three phases weight multiplication, summation, and application activation function.

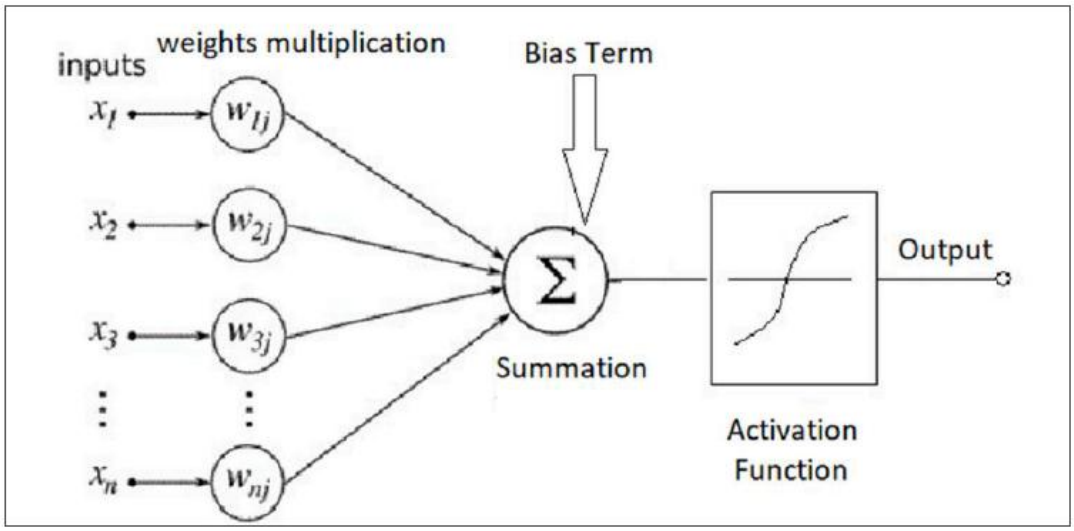

Figure 2.15. The architecture of ANN [91].

In the first phase, the inputs x_i of the ANN were multiplied by the weight coefficients w_{ij} . The weight values were employed to determine the importance of each input phrase. Equation 2.11 shows a summation function was utilized to compute the total of all weighted inputs in conjunction with a bias component b_j.

$$N_j = \sum_{i=1}^{n} w_{ij}\,(x_i) + b_j \tag{2.11}$$

Equation 2.12 shows the $Out\ put\ j$ represents the final stage, which involves applying an activation function to the summation term, which produces an output [91].

$$Output_j = f\,(N_j) \tag{2.12}$$

2.4.3. Recurrent neural networks (RNN)

A recurrent neural network is an example of an artificial neural network characterized by its built-in memory and capacity to analyze complex temporal sequences [92]. Figure 2.16 shows the architecture of the RNN.

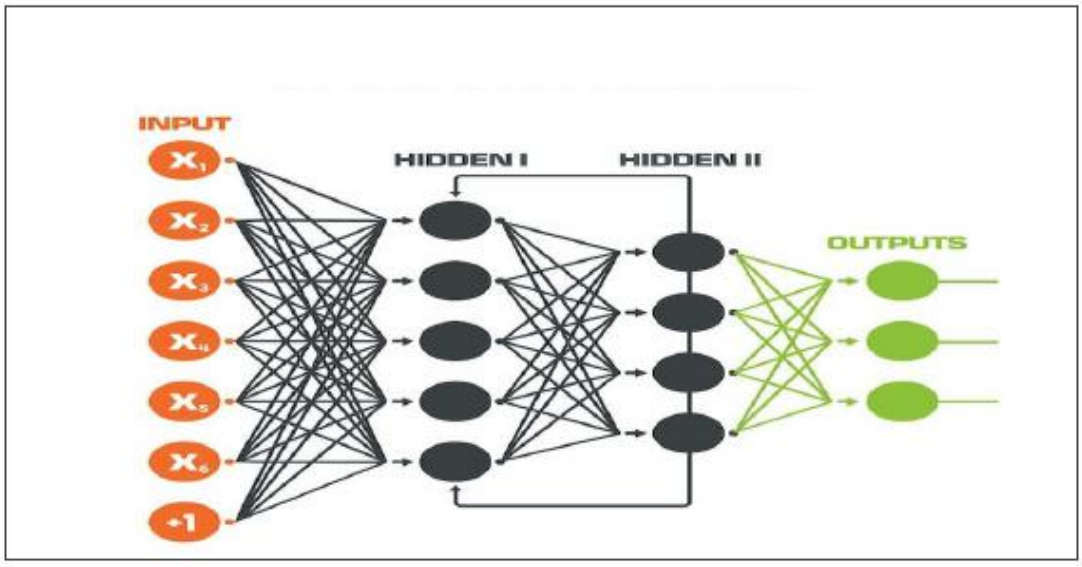

Figure 2.16. The layout of a RNN [93].

In recurrent neural networks, the outcomes of previous iterations serve as input. Figure 2.17 shows the input x_t a segment of the neural network (A) generates the value ht, and a loop enables the transmission of data between nodes within the network. The hiding state is adjusted correspondingly when new data is produced in a preceding state [94]. The hidden layers of the RNNs that can save information for subsequent utilization.

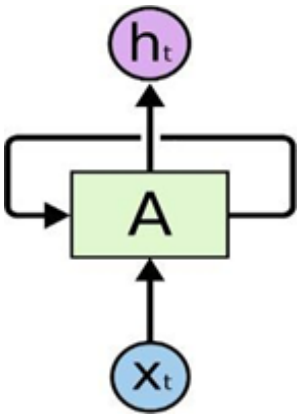

Figure 2.17. The loop structure of a RNN [94].

As a result of these repetitions, the RNNs may appear confusing. However, upon further reversal, it becomes clear. To understand how a RNN works, it helps to imagine numerous identical networks, each of which transmits data to the next one. The layout of any RNN consists of a series of repeating modules. A single tanh layer is an example of how this repeating module is often structured in a

regular RNN [95]. Figure 2.18 shows the part of the neural network illustration that processes input x_t and returns the result h_t.

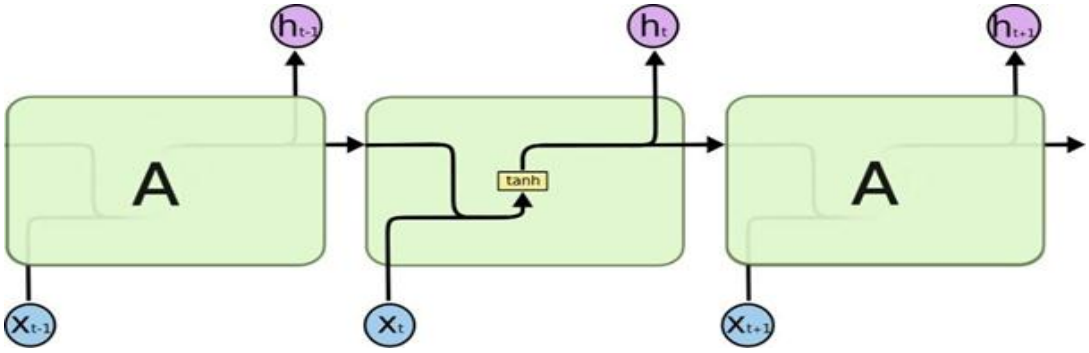

Figure 2.18. A part of the RNN structured model [94].

2.4.4. Long short-term memory (LSTM)

The LSTM neural network is another form of RNN. LSTM is a superior RNN model capable of addressing issues related to gradient disappearing and long-term dependency problems [96]. The cell state and gate architecture are fundamental to the LSTM algorithm. Cell states possess the capacity to convey information and transcend the constraints of short-term memory. The LSTM comprises three separate gate structures: the input gate, the forget gate, and the output gate. Each of these gates has a distinct purpose within the overall system [96]. In contrast to traditional neural networks, which simply map the inputs to outputs, LSTMs can be trained to learn a mapping function that connects inputs and outputs over time. The structure of the LSTM, which is like that of the RNN, but, the configuration of the LSTM's repeating module is organized [96]. Figure 2.19 shows the LSTM diagram where xt is the LSTM's input vector, t is the LSTM's forget gate activation vector, and it is the LSTM's input gate activation vector, ot is the LSTM's output gate activation vector, ct is vector cell state vector, $h\,t$ is the LSTM unit output. When training LSTM networks backpropagation in time was employed to prevent the vanishing gradient problem from occurring [97].

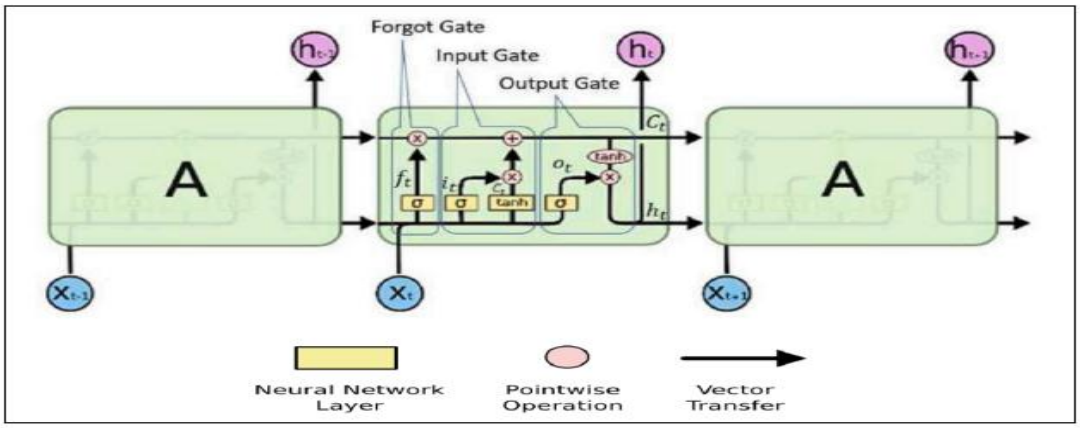

Figure 2.19. The LSTM diagram [94].

2.5. Measures of Evaluation

The mathematical foundations the MAE, MSE, and R^2 are used to assess a regression model's performance.

2.5.1. Mean absolute error (MAE)

When data contains outliers that suggest tainted values, the MAE application is appropriate. In fact, the mean absolute error provides a thorough and constrained performance statistic for the model without unduly penalizing training outliers. On the other hand, if the test set has a large number of outliers, the model will perform better [98]. Equation 2.13 shows the formula of mean absolute error. Here, Positive infinity is the worst value, while zero is the best.

$$\mathbf{MAE} = \frac{1}{N}\sum_{i=1}^{N}|x_i - y_i| \tag{2.13}$$

Here, x_i is the predicted value while y_i is the observed value.

2.5.2. Mean square error (MSE)

In situations where it is required to identify outliers, the Mean Squared Error might be used. In fact, MSE works very well for giving the data points more weight [98]. Equation 2.14 shows the formula of mean square error. Here, Positive infinity is the worst value, while zero is the best.

$$\mathbf{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \tag{2.14}$$

Here, x_i is the predicted value while y_i is the observed value.

2.5.3. Coefficient of determination (R-squared or R^2)

In linear regression models, the amount of observed variation in the dependent variable that can be assigned to the known predictors is measured using an accuracy statistic called the coefficient of determination. In this case, the coefficient of determination has a best value of +1 and a worst value of minus infinity. Equation 2.15 shows the formula of coefficient of determination R^2.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(x_i - y_i)^2}{\sum_{i=1}^{N}(\bar{y} - y_i)^2} \quad (2.15)$$

Here, x_i is the predicted value while y_i is the observed value.

3. METHODOLOGY OF THE PROPOSED SYSTEM

3.1. Introduction

This chapter establishes the fundamental structure of the proposed ML, ensemble learning, and DL models for predicting oil and gas production. The initial phase contains the proposed system's design, including detailed information regarding the system's architecture. The subsequent phase explained the oil and gas production dataset totally, along with the pre-processing step. Furthermore, the utilized models and parameters have been defined, explaining the hardware requirements and software implementation in the suggested system. The next chapter offers a detailed description of each phase.

3.2. The Proposed System

The oil sector places significant importance on both time management and accuracy. As a result, companies tend to use AI instead of traditional software solutions. The proposed system employed in this thesis begins with collecting data for oil and gas production and data preprocessing techniques. Then ML, ensemble learning, and DL models are applied to the preprocessed data. Finally, the data is evaluated to assess the results of the thesis. Figure 3.1 shows core architecture of the proposed system. There are multiple stages in the proposed framework: Data collection for oil and gas production and data preparation techniques are the first steps in the suggested system used in this thesis. The preprocessed data is then subjected to ML, ensemble learning, and DL models. Lastly, the data is examined to evaluate the thesis's results.

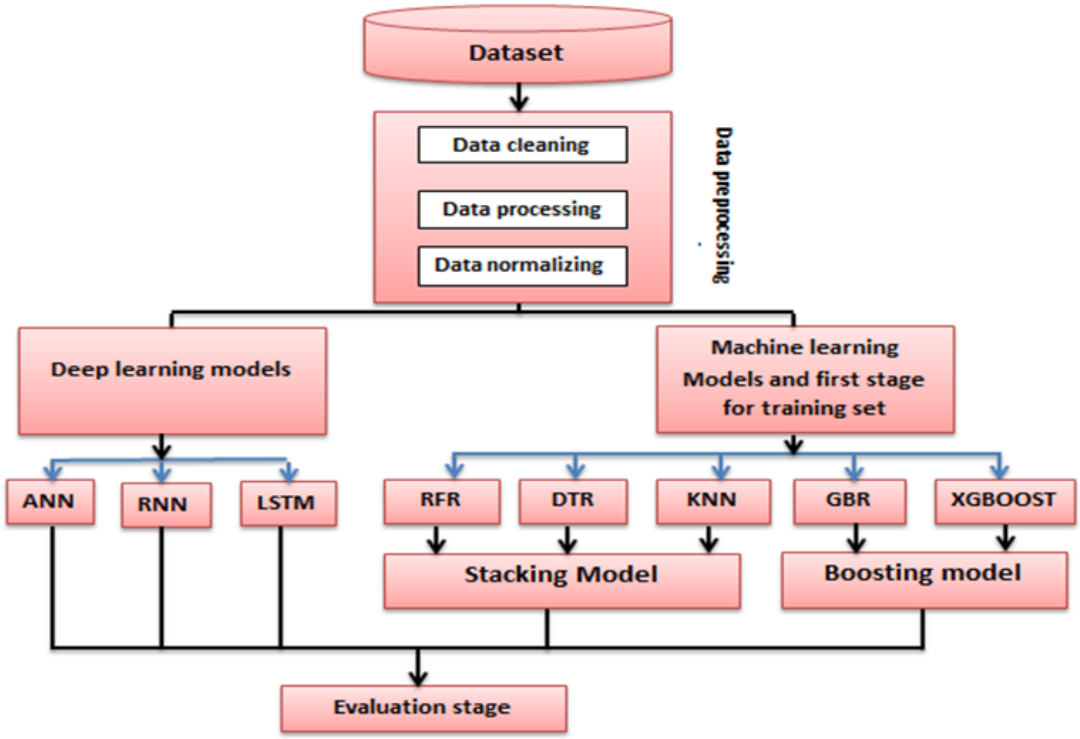

Figure 3.1. The proposed oil and gas production prediction system.

3.3. Dataset Description

Data collection is the first and most crucial step in the thesis process. More than 75,000 wells dug in New York State, USA, between 2001 and the present make up the dataset used in this thesis. The dataset was sourced from the U.S. government's official website and is publicly available [100]. The dataset is represented in 302K rows and 18 columns such as: Reporting Year, Company Name, Production Field, County, Well Type Code, Well Status Code, etc.

Figure 3.2 and Figure 3.3 show a segment of the oil and gas dataset with the focus on data waves for varying types of features, such as active gas and oil wells. Figure 3.4 shows the dataset section of the oil and gas.

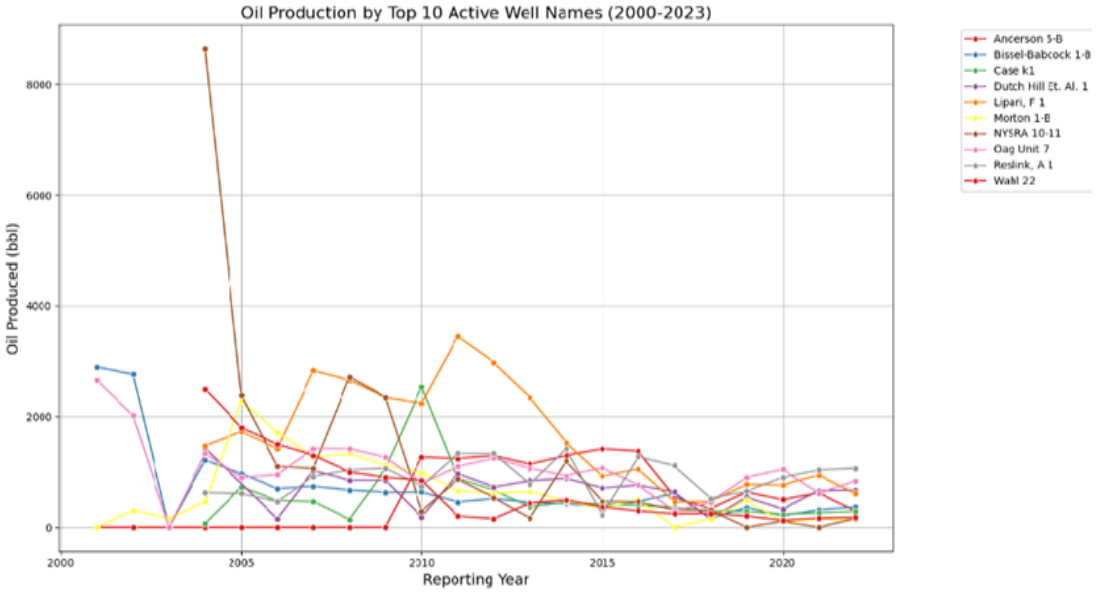

Figure 3.2. Illustrated oil production of top 10 active wells.

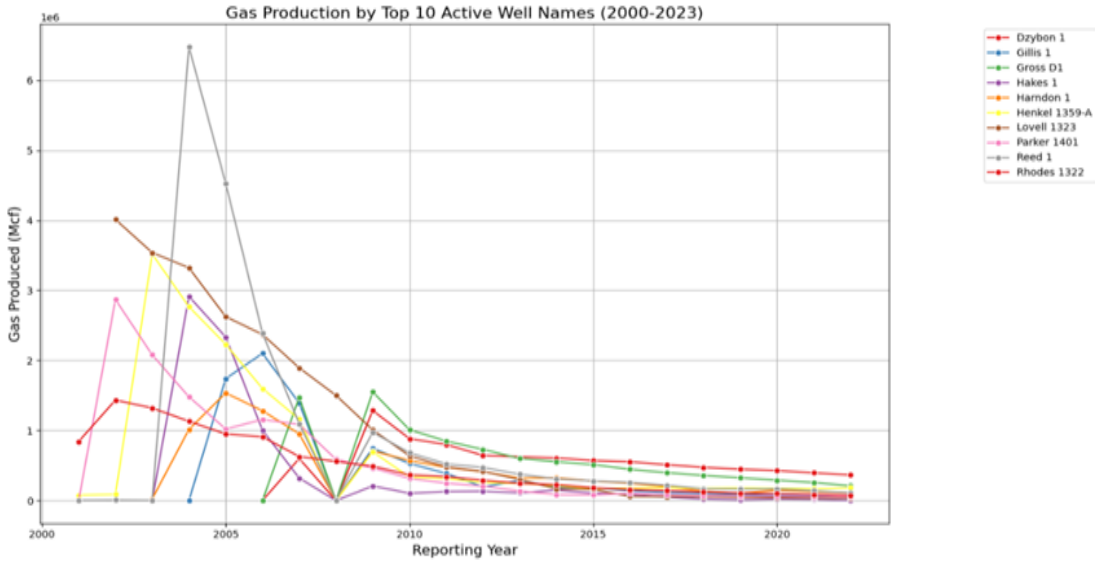

Figure 3.3. Illustrated gas production of top 10 active wells.

	API Well Number	County	Company Name	API Hole Number	Sidetrack Code	Completion Code	Well Type Code	Production Field	Well Status Code	Well Name	Town	Producing Formation	Months in Production	Gas Produced, Mcf	Water Produced, bbl	Oil Produced, bbl	Reporting Year
0	31013172880000	Chautauqua	REI-NY, LLC	17288	0	0	OD	Gerry-Charlotte	AC	Piazza #1	Gerry	Oriskany	0.0	0.0	0.0	0.0	2001
1	31013201220000	Chautauqua	REI-NY, LLC	20122	0	0	GW	Ellery	SI	Bargar #8	Ellery	Akron	0.0	0.0	0.0	0.0	2001
2	31029202970000	Erie	GFS Energy, Inc.	20297	0	0	GD	Elma	PA	Lexo #1	Elma	Grimsby	0.0	0.0	0.0	0.0	2001
3	31029223900000	Erie	Envirogas, Inc.	22390	0	0	GD	Elma	UL	Henderson H #1	Elma	Grimsby	NaN	NaN	NaN	NaN	2001
4	31121141630000	Wyoming	Secor, Allen F. Jr.	14163	0	0	GD	Java	AC	Roche #1	Java	Medina	NaN	NaN	NaN	NaN	2001

Figure 3.4. The dataset section of the oil and gas.

3.4. Data Preprocessing Stage

Preprocessing of the data is necessary before manipulation. This stage contains four basic steps to preprocess data. Figure 3.5 shows the data preprocessing stage, including data cleaning, processing, normalizing, and splitting.

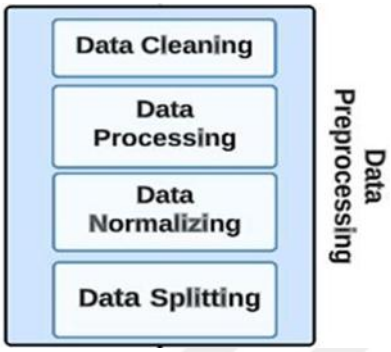

Figure 3.5. Data preprocessing stage.

3.4.1. Data cleaning

Data analysis needs cleaning the data, a critical step in the preprocess. Data cleaning entails the elimination of inaccurate or redundant data from a dataset. This process is crucial for verifying the dataset's accuracy and is utilized to address data noise. It is crucial to guarantee that the dataset is devoid of errors and free from inaccurate or misleading information. The process involves two steps:

- Identifying and establishing any missing or incomplete data points within the dataset.
- Determining suitable measures for addressing missing values by substituting them with relevant alternatives.

Figure 3.6 shows the quantity of unclean information across all dataset attributes and, consequently, that the dataset has been cleaned and is clear of mistakes.

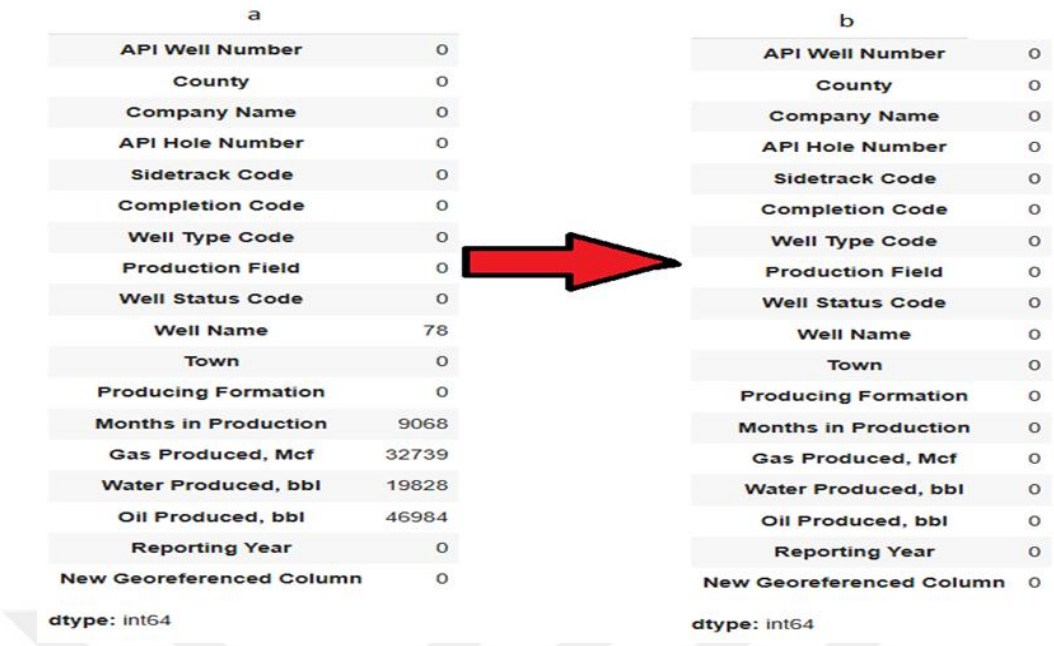

Figure 3.6. The number of unique data before and after cleaning.

3.4.2. Data processing

At this preprocessing stage, an index has been allocated to each feature. This index will be divided into two distinct sets, X and Y. This approach facilitates the attainment of high-accuracy findings with ML, ensemble learning, and DL models following a sequence of experiments.

3.4.3. Data normalizing

Data normalization is a preprocessing technique that enhances the performance of models. It is works by adjusting feature values in the dataset to a standard scale. This process also simplifies data analysis and modeling. One common normalization method is L2 scaling, also known as least squares, which converts numerical values into a range between -1 and 1. The data that needs normalization, such features as company name, well type code, production field, well status code, well name, town, producing formation, and reporting year. Figure 3.7 shows "2-d distributions" are used to explain the distribution of various variables such as Company Name, Well Type Code, Production Field, and Well Status Code against each other and the effects of the normalization process on the data.

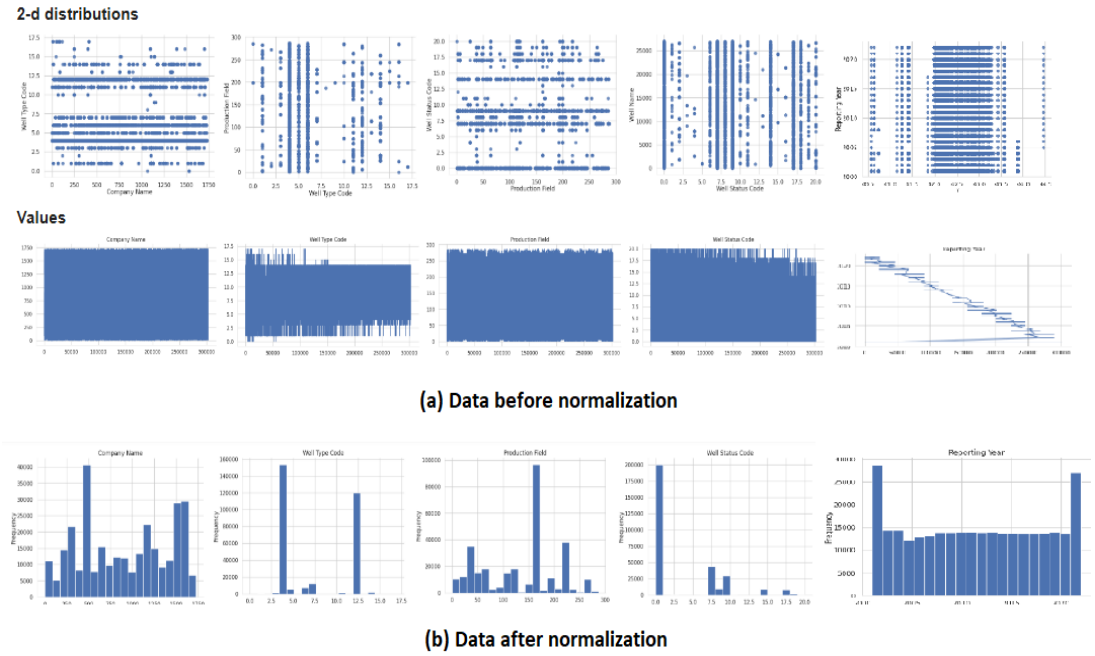

Figure 3.7. The part of the dataset before and after normalization.

3.4.4. Data splitting

Dividing the dataset into three separate sets the training, validation, and testing sets is the final step in the preprocessing process. Usually, it includes 75% of the data. Throughout the training phase, the validation set is used to evaluate the model's performance and modify hyperparameters. The test set, which usually makes up 25% of the dataset, evaluates how well the model performs on brand-new data.

3.5. Machine Learning Algorithms

This section illustrates supervised ML methods for forecasting oil and gas production after applying data preparation tasks, such as cleaning, processing, normalizing, and partitioning the dataset. Machine learning models can be utilized for estimating oil and gas output. Uniform procedures must be followed for every output.

3.5.1. The decision tree regression (DTR) algorithm

The decision tree is a supervised regression approach that combines feature values to create a tree model for predicting continuous outcomes. This technique is used in oil and gas prediction, beginning with data collection and preprocessing, followed by the application of the DTR model. Oil and gas production utilize identical criteria for both oil and gas outputs. Table 3.1 shows the parameter values of the DTR assessed

to get elevated predictive accuracy. The values of these parameters are essential in assessing model performance. Initially, multiple experiments were conducted using manually selected parameters, in order to complicate the parameter selection procedure in this area.

The parameter max_depth determines the maximum depth of the DTR, with a greater value facilitating a more complex tree structure featuring additional levels and enhanced outcomes. The parameter has been configured to 500, signifying the possibility for the decision tree to reach a depth of 500 levels. The parameter random_state has been configured to 33, which controls random number generation. Algorithm 3.1 shows the pseudocode of the DTR model.

Table 3.1. The DTR parameters for oil and gas outputs.

Parameters	value
max_depth	500
random_state	33

Algorithm 3.1. The DTR model [101]

Input: training data (oil and gas production data)

Output: Predicted oil and gas production data
(TR, Target, Attr) Tree-Learning
TR: examples of training

Target: the desired quality

Attr: the collection of descriptive attributes

{
Make the tree's root node.
If TR's target attribute value ti is the same,
The single-node tree is then returned, i.e. Root, with ti as the target attribute

When there are no available descriptive attributes, or
when Attr = empty,
The single-node tree is then returned, i.e. Root, having the most typical Target's value in TR
Otherwise

{

Using an entropy-based metric, choose attribute A

from Attr that best categorizes TR.

Make A the Root attribute.

For every A's legal value, do, v_i, do

{

Add a branch below Root, equivalent to A = v_i

Let be the subset of TR that has A = v_i

If TR_{vi} is empty,

Next, create a leaf node beneath the branch with the target value equal to the most prevalent Target value in TR.

If not, include the subtree that Tree-Learning has learned beneath the branch (TR_{vi}, Target, Attr).

}

}

Return (Root)

}

3.5.2. The random forest regression (RFR) algorithm

Random Forest Regression generates forecasts using a bootstrapping technique, dividing the data into multiple sub-datasets. Subsequently, decision trees are created for each subset of the dataset. The final step involves aggregating predictions from each sub-decision tree to produce a unified forecast, representing the complete random forest model. This model can handle data with numerous dimensions, relies on generalized error estimation. Table 3.2 shows basic parameters to evaluate performance for the RFR model.

Table 3.2. The RFR parameters for oil and gas outputs.

Parameters	value
n_estimators	50
max_depth	15
random_state	33

Here, the n_estimators parameter determines the quantity of trees within the random forest ensemble. Typically, a higher value can improve performance and the max_depth parameter, similar to the counterpart in the decision tree regressor

controls the maximum depth of each decision tree within the forest established here at 15. Finally, the random_state parameter ensuring result reproducibility by fixing the seed for random number generation remains constant at 33. Algorithm 3.2 shows the pseudocode of the RFR model.

Algorithm 3.2. The RFR model [102]

Input: Oil and gas production data

Output: Predicted oil and gas production data

To produce examples of C bootstraps:

For i =1 to c do

To create $D1$, randomly choose training data D with replacement.

Make a root node, $N1$, that contains $D1$.

Build tree call ($N1$)

Finish for

Construct a tree (N)

If there are just one class's instances in N, then

Return

Otherwise

To produce f child nodes of N, $N1,\ldots, Nf$, choose the feature F with the biggest information gain using a random selection of the potential splitting features in N. F can have f possible values ($F1,\ldots, Ff$).

For i= 1 to f do

Call the build tree ($N1$) and set the contents of $N1$ to Di, where Di is all instances in N that match $F1$.

Call build tree (N_1)

Finish for

Finish if

3.5.3. The K-nearest neighbors (KNN) algorithm

KNN uses a distance measure known as the Euclidean distance to identify the k nearest neighbors of new data. It then determines the weighted mean of these nearby data points' target variable values. The estimated value for the new data is derived from the data mean. The parameter values in this case are n_neighbors = 13. 'Uniform' weights and the number of nearest neighbors to take into account while formulating predictions are specified by this option. The weight of the neighbors' contribution to the forecast is determined by this parameter. When 'uniform' is selected, all 13 nearest neighbors have the same impact on the forecast because they are all weighted equally. The algorithm ultimately equals "auto." The algorithm used to determine the nearest neighbors is specified by this parameter. When 'auto' is selected, the KNN model automatically determines the best algorithm based on the values of n_neighbors and the size of the dataset. Algorithm 3.3 shows the pseudocode for the regression of a KNN model.

Algorithm 3.3. The KNN model [103]

Begin

1: load the test and training data.

2: Select the K value.

3. Determine each training data point's Euclidean distance.

4: Sort and store the Euclidean distances in a list.

5: Select the initial k points.

6: Based on the majority of classes, assign a class to the test point.

7: For every point in the test data, repeat Steps 3, 4, 5, and 6.

End

3.5.4. The extreme gradient boosting (XGBoost) algorithm

Before applying XGBoost, optimizing parameters is crucial to ensure that the model performs its best. Many tests were initially conducted with parameters chosen manually. The parameters are objective equal to 'reg:squarederror'. This parameter specifies the learning task and the corresponding objective function to be minimized

during training and n_estimators equal to 50. This parameter defines the number of boosting rounds or trees that the model will build. The learning_rate equals 0.1, the learning rate controls the weight of each new tree added during the boosting process. Finally, seed equal to 300 this parameter sets the random seed for reproducibility. By fixing the seed, the model will produce the same results every time it is run, given the same input data and parameters. This is useful for debugging and ensuring consistent results during experimentation. These parameter values were chosen in order to optimize and achieve high preformance. Algorithm 3.4 shows the pseudocode of the XGBoost model.

Algorithm 3.4. The XGBoost model [104]

Input: Oil and gas production data

Output: Predicted oil and gas production data

Set input data: x (feature matrix) and y (target oil production or gas production)

1. Set the number of base learners ($n_{estimators}$)
2. Initialize ensemble predictions with zero values: y_{pred} = 0

   For each base learner, i from 1 to $n_{estimators}$
3. Calculate residuals: residuals = y-y_{pred}
4. Learn the base learner on residuals and features x
5. Get the forecast of the current base learner:

   Y = basic $_{learner.predict}$ (x)
6. Update ensemble predictions: $y_{pred} = y_{pred} + learning\ _{rate} * y_{pred\ i}$
7. Return ensemble predictions: y_{pred}

3.5.5. The gradient boosting regressor (GBR) algorithm

The Gradient Boosting Regressor combines many weak learners, typically decision trees, into a single strong predictive model. Algorithm 3.5 shows the steps of GBR model for regression. The parameter values of n_estimators are assigned to 50, learning_rate to 0.1, and verbose to be 1, respectively. Here, n_estimators means that 50 trees will be added one by one to gradually improve the model's accuracy. Learning_rate is moderate, balancing the trade-off between the number of trees

needed and model accuracy. The verbose a hyperparameter that controls the contribution of each tree to the final prediction.

Algorithm 3.5. The GBR Model [77]

Begin

1. Initialize predictions:

   $\hat{y}^{(0)}$ = mean (y_{train})

   The initial prediction for every instance is set to the mean of the target values y_{train} in the training dataset. This provides a starting point for the algorithm and helps to reduce bias in the initial stages.

2. For each estimator m in range 1 to M :

   Here, M represents the total number of weak learners that the algorithm will use. Each iteration adds a new tree that corrects the errors (residuals) of the previous model.

   a. Calculate residuals: $r_m = y_{train} - \hat{y}^{(m-1)}$

   b. Fit a weak learner (decision tree) h_m(x) to the residuals r_m

   Train the decision tree h_m using x_{train} and residuals r_m.

   c. Update predictions:

   $\hat{y}^{(m)} = \hat{y}^{(m-1)}$ + learning_rate * h_m (x_{train})

3. Final predictions

   $\hat{y}^{(M)} = \hat{y}^{(M-1)}$ + learning_rate * h_M (x_{train})

   After M iterations, the final prediction model is obtained, which is then used to predict oil and gas data.

End

3.6. Ensemble Learning Algorithms

This section explains supervised ensemble learning algorithms for oil and gas production forecasts.

3.6.1. The bagging algorithm

The Bagging Regressor model with a Random Forest Regressor estimator merges two ML techniques the bagging (Bootstrap Aggregating) and Random Forest. This technique creates multiple subsets of data samples with replacement from the original dataset trains a model (in this thesis a Random Forest Regressor) on each subset and then averages the predictions from all models to get a final prediction. The parameter value is estimator equal to Random Forest Regressor, while n_estimators equal to 50 and random-state equal to 42. These parameters specify the base estimator used in each bagged model. Algorithm 3.6 shows the pseudocode of the bagging model.

Algorithm 3.6. The bagging model [78]

Begin

1. Initialize: Set M is the number of estimators and defines the base estimator in this thesis the base estimator is RandomForestRegressor model.

2. Training Phase: For each estimator m (where m ranges from 1 to M)

a- Sampling: Randomly select a subset of the training data (x_{train} , y_{train}) with replacement to form a unique training subset (x_m , y_m). This process is known as bootstrap sampling.

b- Training: Train the RandomForestRegressor on the sampled subset (x_m , y_m) using its specified parameters n_estimators equal to 50, which indicates the number of trees in the RandomForest model.

3. Prediction for each test instance:

a- Run all base models on the test instance.

b- Average predictions across all models for the final output.

4. The final prediction for each test instance is the average of the predictions from all the base estimators. This averaged prediction is the output of the bagging model. End

3.6.2. The boosting algorithm

Boosting model like XGBoost and Gradient Boosting work well with complex datasets and can model intricate relationships between features, making them highly effective for tasks like oil and gas production prediction.

The parameters for two models are as follows:

- XGBoostModel ➡ objective='reg:squarederror', n_estimators=50, learning_rate=0.1, seed = 300.
- GradientBoostingRegressorModel ➡ n_estimators=50, learning_rate=0.1, verbose=1.

Here, the XGBRegressor model is utilized with the parameter objective equal to'reg: squarederror'. The objective function, a squared error, sometimes referred to as the mean squared error, is specified by this parameter. Because it calculates the average squared difference between the expected and actual values, this decision qualifies the model for regression tasks.

For both the XGBoost and GBR models, the parameter n_estimators is set to 50. Each model will build an ensemble of 50 weak learners, typically decision trees. The parameter controls the number of trees or weak learners to be added sequentially to the model. Each tree aims to correct the errors made by the previous ones, contributing to the final prediction in a step-by-step manner.

In this thesis, XGBoost is managed by the seed parameter, and for GBR, it's typically done with the verbose parameter. These parameters control how the data is split or how the trees are initialized, ensuring that the results can be reproduced by providing a consistent random number generator seed. This process allows both models to learn from the data iteratively and improve predictions while ensuring that the results are consistent across multiple runs when the same random seed is used. Algorithm 3.7 shows the steps of boosting model for regression.

Algorithm 3.7. The boosting model [80]

Begin

1. Set a fixed value to start the initial predictions, usually the y train mean: $y^{(0)}$ = mean (y train)
2. For each estimator m = 1 , ..., M:

a- Compute residuals (errors from previous predictions):

$$r_{m=} y_{train-y^{(m-1)}}$$

b- Fit the residuals to a weak learner, often a short decision tree r_m :

$$h_{m\,(x)=} r_m$$

c- Add the new weak learner's contribution to the model predictions, scaled by the learning rate α. :

$$y^m = y^{(m-1)} + \alpha \, . h_m \, (x)$$

3. Final prediction: After M iterations, the final prediction is the sum of all models:

$$y^{(m)} = \sum_{m=1}^{M} \alpha \, . h_m \, (x)$$

4. Output: Final prediction values $y^{(M)}$

End

3.6.3. The stacking algorithm

An advanced ensemble learning method called a stacking regressor combines the outputs of several base models to increase prediction accuracy. In this thesis, the selected base models are the RFR, DTR, and KNN. These models are chosen for their diverse approaches to learning patterns from the data, thereby increasing the overall robustness and accuracy of the predictions. The stacking regressor works by using the predictions from these base models as input features for a final model, often called the meta-model or final estimator. The meta-model is trained to learn the relationships between the base models' predictions and the actual target variable, in order to improve the final prediction performance. In this the stacking model, the RFR is used as the meta-model. It is configured with n_estimators equal to 50, which means it builds an ensemble of 50 decision trees, and random_state equal to 42, ensuring the results are reproducible by fixing the randomness in the process. The overall procedure involves training the base models on the training data and then using their predictions as new features for the meta-model. The meta-model learns from these new features to make the final prediction. This technique leverages the

strengths of each base model, allowing the stacking regressor to make more accurate and robust predictions by combining the diverse predictions from the base models. Algorithm 3.8 shows the steps of stacking model for regression.

Algorithm 3.8. The stacking model [105]

Begin

1. To do cross-validation, divide the training data (x_{train} , y_{train}) into K equal parts (folds).
2. Training Base Models:
    a- For each base model M_i in the set of base models.
    b- Train M_i on K folds and predict on the remaining fold to get out – of fold (OOF) predictions. Repeat for all K folds, collecting OOF predications for the entire training set.
    c- The OOF predictions from each base model are collected as new features, forming a new dataset x_{meta} with predictions as features.
3. Train Final Estimator: Use the new dataset x_{meta} as input features to train the final estimator (in this thesis, a RandomForestRegressor).
4. Final prediction: For test data, use each trained base model to predict the target variable, and use these predictions as features to get the final prediction from the final estimator.
5. Output: final prediction for test data.

End

3.7. Deep Learning Algorithms

This section demonstrates the application of supervised DL models algorithms for forecasting oil and gas production. The same workflow and procedures are applied consistently for each target output.

3.7.1. Artificial neural network (ANN) algorithm

ANN model has two hidden layers and an output layer. The optimal of hidden layers and neurons for the ANN model was determined through many tests. Table 3.3 shows the ANN hidden layers, units of nodes, and activation function for each layer.

Table 3.3. The ANN layers and activation function in the thesis.

Number of hidden layers	Units of nodes	Activation function
First hidden layer	128	Tanh
Second hidden layer	256	Tanh
Third layer (output)	1	ReLU

To improve the model's performance, ideal parameters such as the optimizer, loss function, and evaluation metrics, batch size and epoch count are chosen after the model has been trained on the supplied training data. Algorithm 3.9 shows the steps pseudocode of the ANN model. Here's the parameters used in the ANN model are given as follows:

1- Optimizer ADAM:

The ADAM optimizer, or in other words, Adaptive Moment Estimation is worked during training to update the neural network's weights. Large datasets and high-dimensional parameter spaces are ideal for ADAM.

2- Loss Function MSE (Mean Squared Error):

The Mean Squared Error (MSE) is used as the loss function for this regression task. The mean squared difference between the actual and predicted values is computed. A smaller mean square error (MSE) in regression issues indicates that the model's predictions are closer to the actual values.

3- Batch Size 128:

The number of training samples processed prior to the model updating its parameters is determined by the batch size. The model processes 128 samples at a time before updating when the batch size is set to 128. This helps in balancing the model's learning stability and training speed.

4- Epochs 40:

An epoch is one complete pass through the entire training dataset. Setting the number of epochs to 40 means the model will iterate over the whole dataset 40 times, allowing it to learn and refine its weights with each pass.

Here's the parameters used in the algorithm are given as follows:

- n_{input} : Count of features that are input.
- n_{hidden} : Number of hidden layer neurons.
- n_{output} : Number of output neurons.
- W_{input_hidden} : Input weights for the hidden layer.
- B_{hidden} : The hidden layer's biases.
- W_{hidden_output} : Weights from hidden to output layer.
- B_{output} : Biases for the output layer.

Algorithm 3.9. Training and testing ANN model in the thesis [106]

Inputs x(t): Define the number of input features, hidden layer neurons, and output.

Build Model: The model architecture has two hidden layers, a set optimizer = "ADAM," activation function for hidden layer=" Tanh" and the output layer = "ReLU," Loss function="MSE"

For I = 1 to n # numbers of epochs

- Forward Pass:

  Calculate hidden layer input:

$$Z_{hidden} = X \cdot W_{inpt\_hidden} + B_{hidden}$$

  Apply activation function to get hidden layer output:

$$A_{hidden} = \tanh(Z_{hidden})$$

  Calculate output layer input:

$$Z_{output} = A_{hidden} \cdot W_{hidden\_output} + B_{output}$$

  Apply activation function to get predicted output:

$$\hat{Y} = relu\ (Z_{output})$$

- Calculate Loss: Compute the loss between predicted and target outputs

Loss= calculate_loss ($Y_{true,}$ $\hat{Y}$)

- Backward Pass (Backpropagation):

  Compute output layer error

$$\delta_{output} = calculate\_output\_error(Y_{true,}Y\hat{}\,)$$

Compute hidden layer error:

$$\delta_{hidden} = \delta_{output} \cdot W_{hidden\_output}$$

- Update Weights and Biases:

  Update weights and biases for the output layer.

  Calculate: MSE, MAE, R^2

Outputs: The estimated value of predicted oil and gas production.

3.7.2. The RNN model algorithm

The RNN model has a structure made up of five layers. RNN has four hidden layers and one output layer. Table 3.4 shows the RNN hidden layers and the units of each layer. In order to acquire knowledge, it is necessary to establish fundamental parameters for the construction of this model. The basic parameters used are optimizer equal to ADAM, loss equal to MSE, and metrics equal to MSE. The model uses a batch size of 128 and epochs 40. Algorithm 3.10 shows the pseudocode of the RNN model.

Table 3.4. The constituents of the hidden and output layers for the RNN model in the thesis.

Number of hidden layers	Units of nodes	Activation function
First hidden layer	64	Tanh
Second hidden layer	32	Tanh
Third hidden layer	16	Tanh
Fourth hidden layer	8	Tanh
Fifth layer (output)	1	ReLU

Here's the parameters used in the algorithm are given as follows:

- n_{input} : Count of features that are input.

- n_{hidden} : Number of hidden layer neurons.
- n_{output} : Number of output neurons.
- W_{input_hidden} : Input weights for the hidden layer.
- B_{hidden} : The hidden layer's biases.
- W_{hidden_output} : Weights from hidden to output layer.
- B_{output} : Biases for the output layer.

Algorithm 3.10. Training and testing the RNN model in the thesis [107].

Inputs x(t): Define the number of input features, hidden layer neurons,

and output neurons.

Build Model: The model architecture has five hidden layers, set optimizer = "ADAM," activation function for hidden layer=”Tanh” and the output layer = “ReLU," Los function="MSE"

For I = 1 to n # numbers of epochs

- Forward Pass:

  Calculate hidden layer input:

  $$Z^{(t)}hidden = X^{(t)} \cdot W_{input\_hidden} + H^{(t-1)} \cdot W_{hidden\_hidden} + B_{hidden}$$

  Apply activation function to get hidden layer output:

  $$H^{(t)} = tanh\ (Z^{(t)}_{hidden})$$

  Calculate output layer input:

  $$Z^{(t)}output = H^{(t)} \cdot W_{hidden\_output} + B_{output}$$

  Apply activation function to get predicted output:

  $$\hat{Y}\ (t) = ReLU\ (Z^{(t)}_{output})$$

- Calculate Loss: Compute the loss between predicted and target outputs

  $$Loss^{(t)} = calculate\_loss\ (Y^{(t)}_{true}, \hat{Y}^{(t)})$$

- Backward Pass (Backpropagation):

  Compute output layer error

  $$\delta_{output}^{(t)} = \text{calculate\_output\_error}\ (Y_{true}^{(t)}, \hat{Y}^{(t)})$$

  Compute hidden layer error

  $$\delta_{hidden}^{(t)} = (1 - (H^{(t)}) \cdot \delta_{output}^{(t)} \cdot W_{hidden\_output}^{(T)} + \delta_{hidden}^{(t+1)} \cdot W_{hidden\_output}^{(T)}$$

- Update Weights and Biases:

  Update weights and biases for the output layer and hidden layers.

Calculate: MSE, MAE, R^2

Outputs: The estimated value of predicted oil and gas production.

3.7.3. The long short-term memory (LSTM) algorithm

Three hidden layers and one output layer comprise up the LSTM model's structure, the second layer's input is the output of the first LSTM layer. The second layer's output forms the third layer's input, while the third layer's output forms as the fourth layer's input. The expected levels of oil and gas production are produced by the last LSTM layer. Several tests were carried out to find the ideal number of hidden layers and neurons for the LSTM model, and constant verification of correctness following each modification demonstrated that the four hidden layers produced an acceptable result. Table 3.5 shows the units of nodes and activation function for each hidden layer in the LSTM model. The basic parameters used are optimizer equal to ADAM, loss equal to MSE, and metrics equal to MSE. The model uses a batch size of 128 and epochs 20. Algorithm 3.11 shows the pseudocode of the LSTM model.

Table 3.5. The constituents of the hidden and output layers for the LSTM model of the thesis.

Number of hidden layers	Units of nodes	Activation function
First hidden layer	64	ReLU
Second hidden layer	64	ReLU
Third hidden layer	32	ReLU
Fourth layer (output)	1	ReLU

Here's the parameters used in the algorithm are given as follows:

- W_{xi} : The input x(t) weight matrix.
- W_{hi}: The preceding hidden state's weight matrix is h(t−1).
- b_i : Biases for the input gate.
- σ : The function of sigmoid activation to regulate the flow of information into the cell state.
- h (t): Updated hidden state at time t, which also serves as the LSTM's output for that time step.
- c (t): Updated cell state at time t.
- Candidate Value: Provides potential new content for the cell state.
- Output Gate: Modifies the hidden state and determines the final output.

Algorithm 3.11. Training and testing LSTM model in the thesis [108].

Inputs x(t): Feature matrix oil and Gas production at time t.

Build Model: The model architecture has four hidden layers, set Optimizer = "ADAM," activation function for hidden layer=”ReLU," and for output layer = “ReLU," Loss function ="MSE".

For I = 1 to n # numbers of epochs 20

Forget gate: $f(t) = \sigma (W_x f_x(t)+W_h f_h(t-1)+W_c f_c(t-1)+bf)$

Input gate: $i(t) = \sigma (W_x i_x(t)+W_h i_h(t-1)+W_c i_c(t-1)+bi)$

Candidate value**:** $g(t) = \tanh(W_x g_x(t)+W_h g_h(t-1)+bg)$

Update the cell state**:** $c(t) = f(t) \otimes c(t-1)+i(t) \otimes g(t)$,

Update the output of the LSTM:

$o(t) = \sigma (W_x o_x(t)+W_h o_h(t-1)+W_c o_c(t)+bo)$;

$h(t) = o(t) \otimes \tanh(c(t))$.

Calculate: MSE, MAE, R^2

Outputs**:** Target matrix representing the actual production oil and gas values to be predicted.

4. EXPERİMENTAL RESULT

4.1. Introduction

This chapter discusses the experimental results of prediction oil and gas production using machine learning, ensemble learning and deep leering models. All models were trained and tested within a collaborative online Python programming environment.

Initially, Section 4.4 discusses the results of machine learning models: DTR, RFR, KNN, XGBoost, and GBR. Subsequently, Section 4.5 discusses the results of ensemble learning models: Bagging , Boosting and Stacking models and Section 4.6 shows results of deep learning models: ANN, RNN, and LSTM. In Section 4.7 the results of all models are compared to identify the most effective model by analyzing their performance. Finally, section 4.8 shows a list of previous studies and our proposed methods.

4.2. Hardware Requirements

A personal computer running 64-bit Windows 10 Home with an HP ProBook 4530s Core i7 CPU and 12 GB of RAM was used as a computational resource for this thesis.

4.3. Software Requirements

The programming languages used in this thesis is Python, Anaconda Jupyter and Collaboration Online (https://colab.research.google.com) as Python environments.

The Python libraries used in this thesis and their descriptions are given as follows:

- NumPy: A package that enables array manipulation in Python.
- Tensor flow and Keras: Libraries that make writing code for Deep Learning models more effective.
- Scikit-learn (Sklearn): The most popular Machine Learning library, used for partitioning and fitting datasets into numerous machine learning models.

- Pandas: The library which is employed for importing and partitioning datasets. Additionally, it aids in data visualization with the help of the Matplotlib package, which can produce dynamic and animated charts.

4.4. Machine Learning Model Results

This section explains the result of ML models in this thesis for oil and gas production prediction output. We tested five machine learning supervised regression models: DTR, RFR, KNN, XGBoost, and GBR. The dataset was trained and tested across these five ML models. The model's results were obtained after normalization and processing of the data. Table 4.1 shows the results of machine learning and demonstrates the model's performance using the evaluating matrix MSE, R^2, and MAE.

Table 4.1. Results of models for machine learning.

ML MODELS	Output	MAE	MSE	R^2
DTR	Oil	0.000207	0.000017	0.949276
	Gas	0.000141	0.000015	0.999598
RFR	Oil	0.000509	0.000026	0.960828
	Gas	0.000186	0.000010	0.999733
KNN	Oil	0.000207	0.000017	0.938847
	Gas	0.000313	0.000014	0.999580
XGBoost	Oil	0.002263	0.000118	0.738127
	Gas	0.000298	0.000026	0.963784
GBR	Oil	0.003067	0.000266	0.391319
	Gas	0.004740	0.000141	0.996638

As seen from the table above, the RFR model runs the best with very low MAE and MSE values, in addition to R^2 values which close to one, signifying a high degree of correlation between actual and forecast values. These results indicate that the Machine Learning models successfully predict oil and gas production with RFR particularly promising based on the provided metrics. Actual oil and gas production represents the physical measurements or observations of oil and gas extracted from a well over a specific period. It would reinforce their effectiveness and reliability in predicting oil production, providing confidence in their utility for decision-making and planning purposes in the oil and gas industry. Figure 4.1 shows how closely the five machine learning models (DTR, RFR, KNN, XGBoost, and GBR) match the actual and forecast oil output.

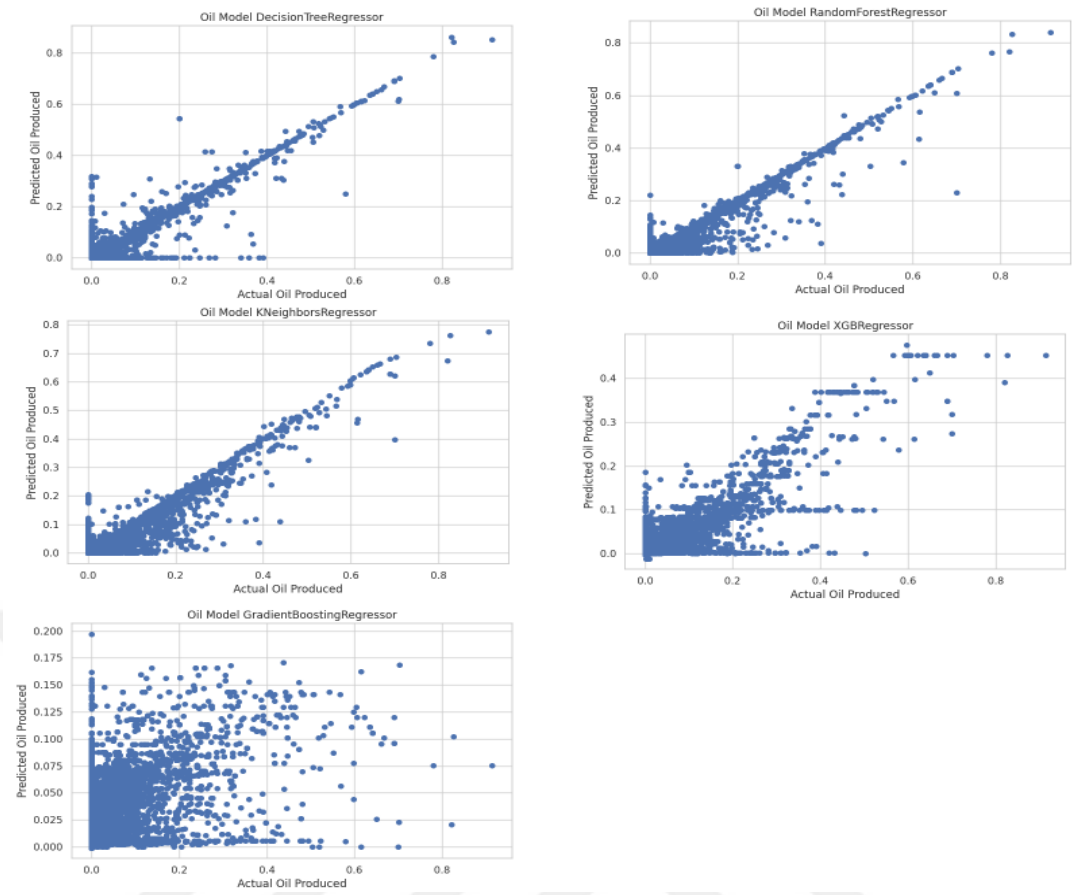

Figure 4.1. Oil production predictions and actual results for five machine learning models (DTR, RFR, KNN, XGBoost, and GBR).

Figure 4.2 shows the predicted gas production and actual gas production. The similarity between predicted gas and actual gas values for models with lower MAE, MSE, and higher R^2 values indicates their effectiveness in predicting gas production accurately.

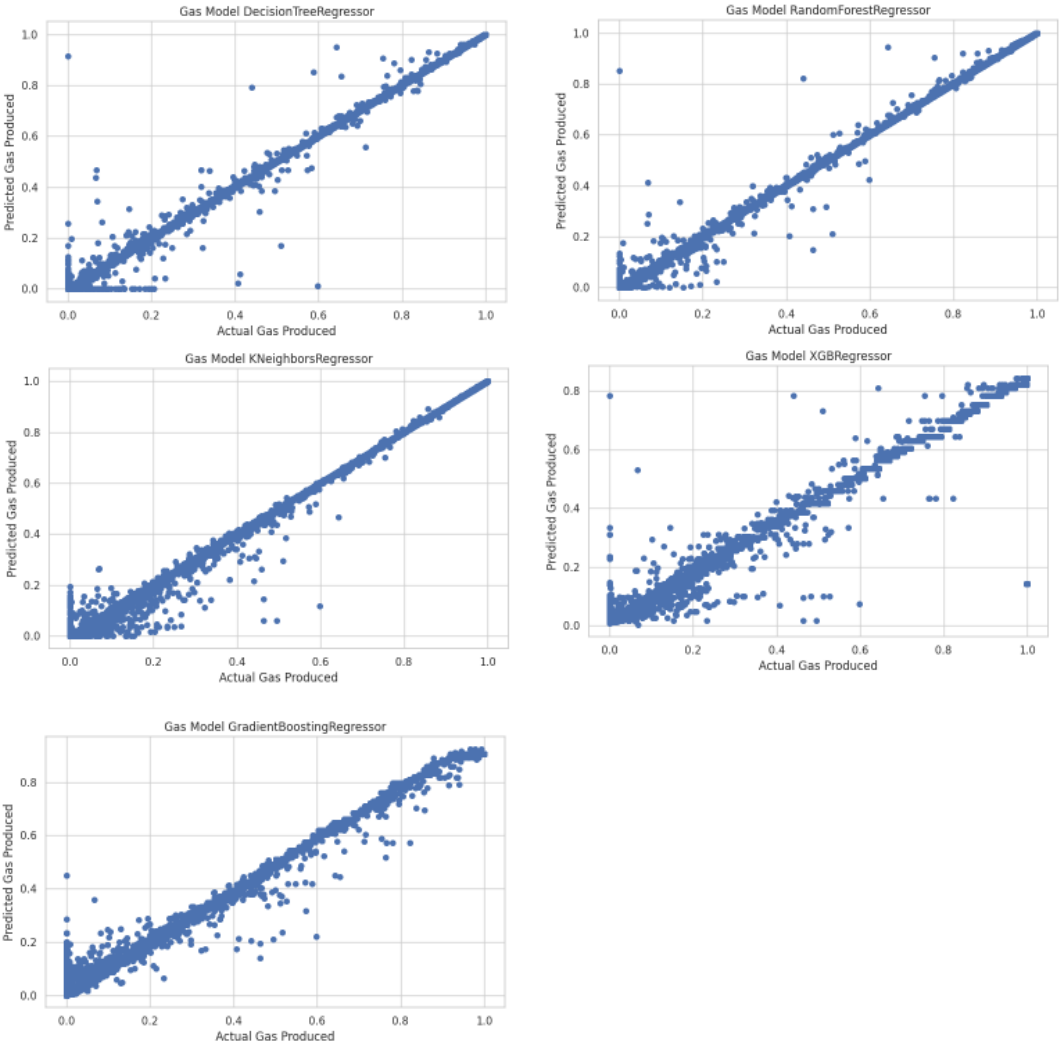

Figure 4.2. Gas production predictions and actual results for five machine learning models (DTR, RFR, KNN, XGBoost, and GBR).

Figure 4.3 shows the comparison of R^2 values for oil and gas in the machine learning models. The Random Forest Regression (RFR) Model's result is higher than that of each model, which means it is more effective for predicting oil and gas production in this thesis.

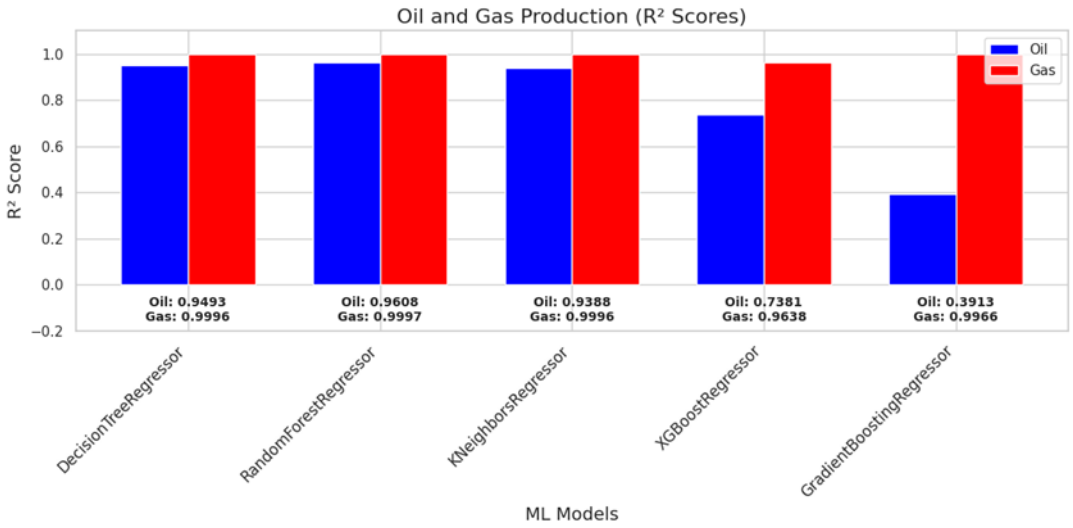

Figure 4.3. Analyzing $\mathbf{R^2}$ values for oil and gas in machine learning models.

4.5. Ensemble Learning Model Results

The results of the models were obtained after combining machine learning models; the bagging model merges two ML techniques: Bagging (Bootstrap Aggregating) and Random Forest; the boosting model merges two ML techniques: GXBoost and GBR; and the stacking model merges three ML techniques: DTR, RFR, and KNN. Table 4.2 shows the results of ensemble learning models and indicates how effectively the models perform using the evaluating matrix MSE,R^2 and MAE.

Table 4.2. Results of ensemble learning models (bagging, boosting and stacking).

Ensemble Learning models	Output	MAE	MSE	R^2
Bagging Model	Oil	0.000276	0.000159	0.964901
	Gas	0.000143	0.000012	0.999724
Boosting Model	Oil	0.001116	0.000047	0.895139
	Gas	0.001283	0.000100	0.997543
Stacking Model	Oil	0.000227	0.000019	0.973602
	Gas	0.000130	0.000007	0.999812

As seen from Table 4.2 above, the stacking model demonstrates the best performance to enhance the results from machine learning models, with very low MAE and MSE values, in addition to R^2 value which is close to one, signifying a high degree of correlation between actual and forecast values. These results indicate that the ensemble learning models successfully enhanced the results for predicting oil and gas production.

Figure 4.4 shows the close match between forecast and actual oil production across ML, including ensemble learning models: Bagging, Boosting, and Stacking.

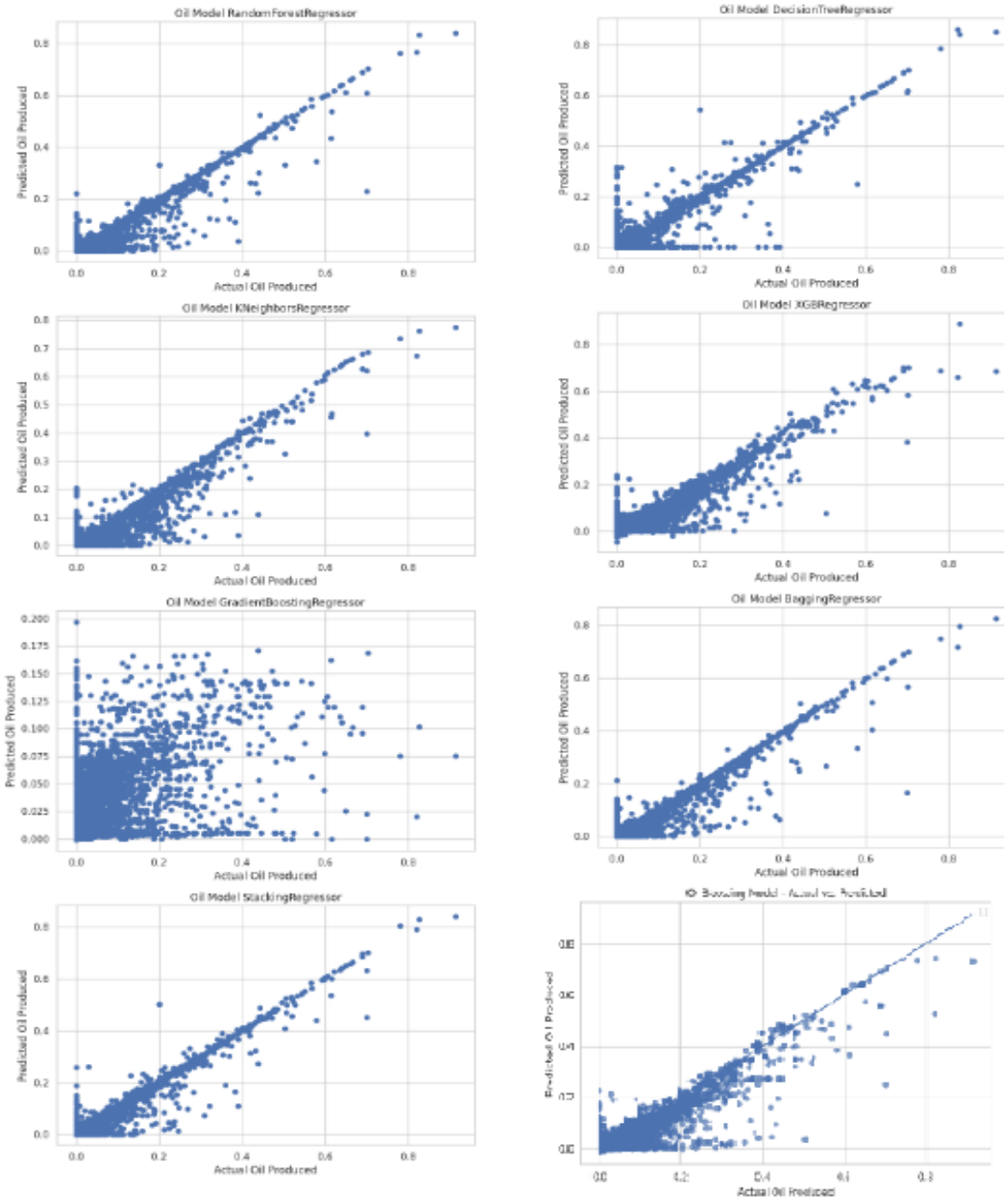

Figure 4.4. Oil production predictions and actual results for machine learning and ensemble learning models.

Figure 4.5 shows the close match between forecast and actual gas production across ML, including ensemble learning models: Bagging, Boosting, and Stacking. The ability of ensemble learning models to accurately predict gas production indicates their effectiveness in models.

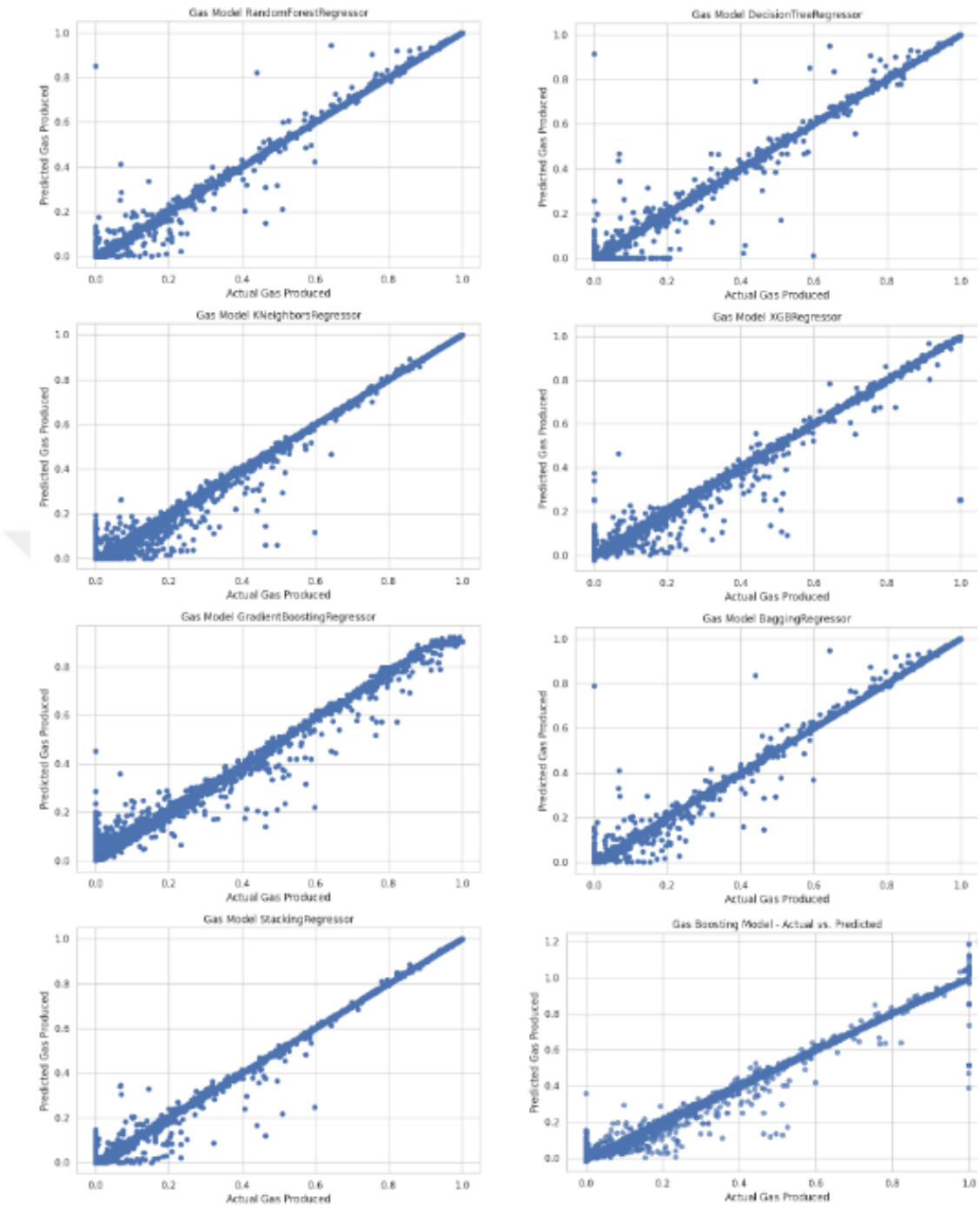

Figure 4.5. Gas production predictions and actual results for machine learning and ensemble learning models.

Figure 4.6 shows the R^2 value results of ML including ensemble learning models. The results of ensemble learning models are higher than those of machine learning, which means that they are effective for predicting oil and gas production.

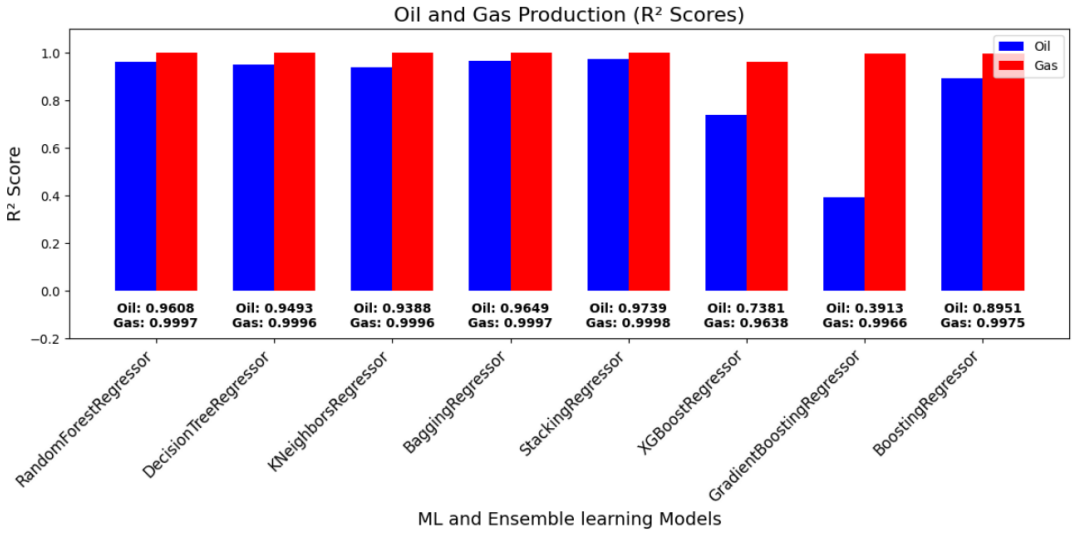

Figure 4.6. Analyzing $\mathbf{R^2}$ values for oil and gas in machine learning and ensemble learning models.

4.6. Deep Learning Model Results

This part illustrates the results of deep learning models which are ANN, RNN, and LSTM.

4.6.1. Result of artificial neural network model

This part illustrates and discusses the results of the ANN model. The experiment began by testing various parameters and hidden layers structures for more accurate results. Table 4.3 shows the results of the ANN model including metrics MAE, MRE, and R^2. Additionally, the results indicate that ANN models perform better with larger batch sizes and less effective with smaller ones.

Table 4.3. Result of artificial neural network mode.

ANN Model	MAE	MSE	R^2
Oil	0.003129	0.000124	0.724388
Gas	0.002746	0.000109	0.997309

These results suggest that the ANN model performs well in predicting oil and gas data, with good accuracy for oil and high accuracy for gas predictive capability. Figure 4.7a illustrates the oil production training and validation loss. Figure 4.7b shows the gas production training and validation loss.

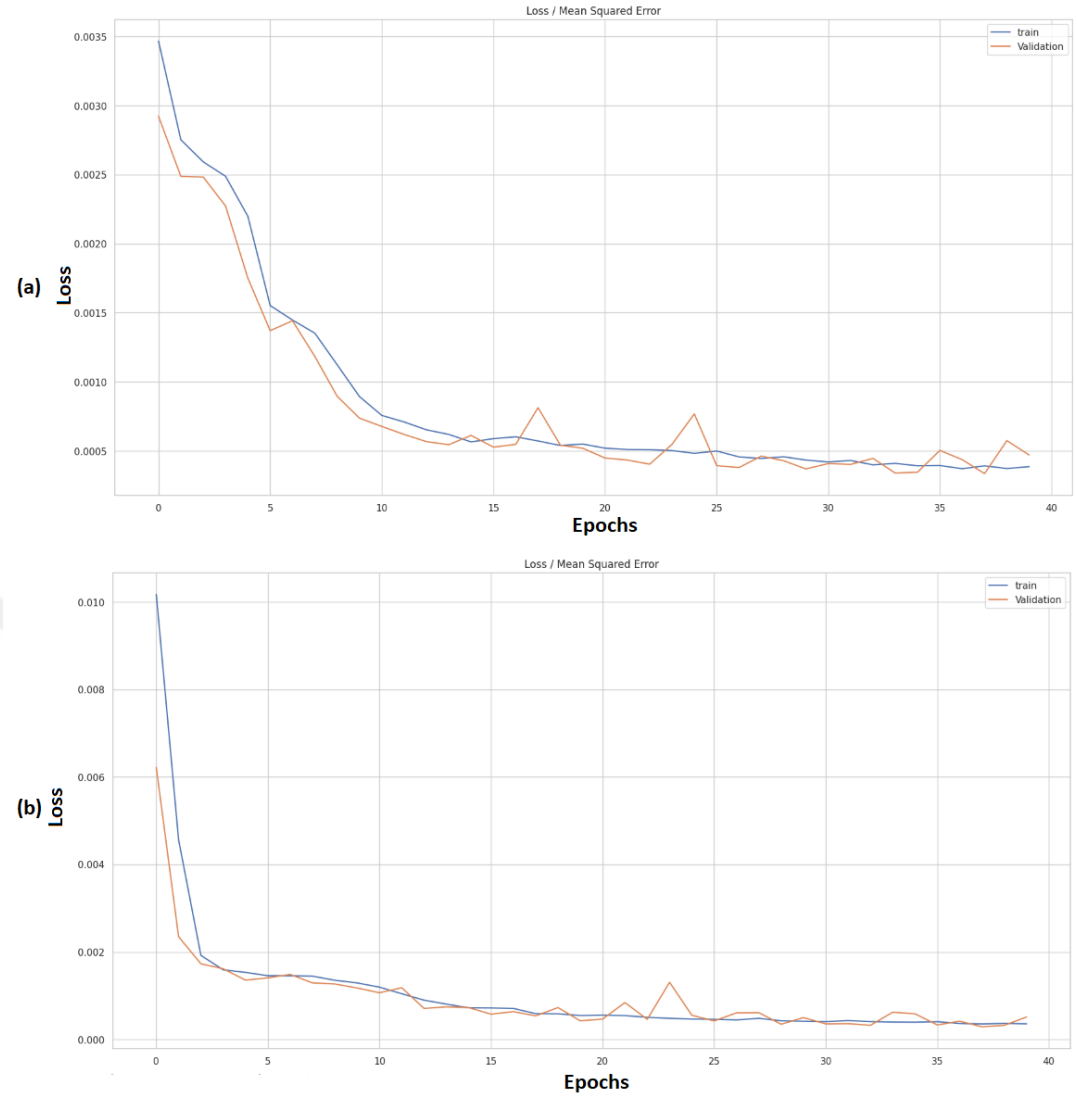

Figure 4.7. (a) ANN model training and validation loss for oil production. (b) ANN model training and validation loss for gas production.

Figure 4.8a shows the matching between predicted oil produced and actual oil produced in the ANN model while Figure 4.8b shows the matching between predicted gas produced and actual gas produced in the ANN model.

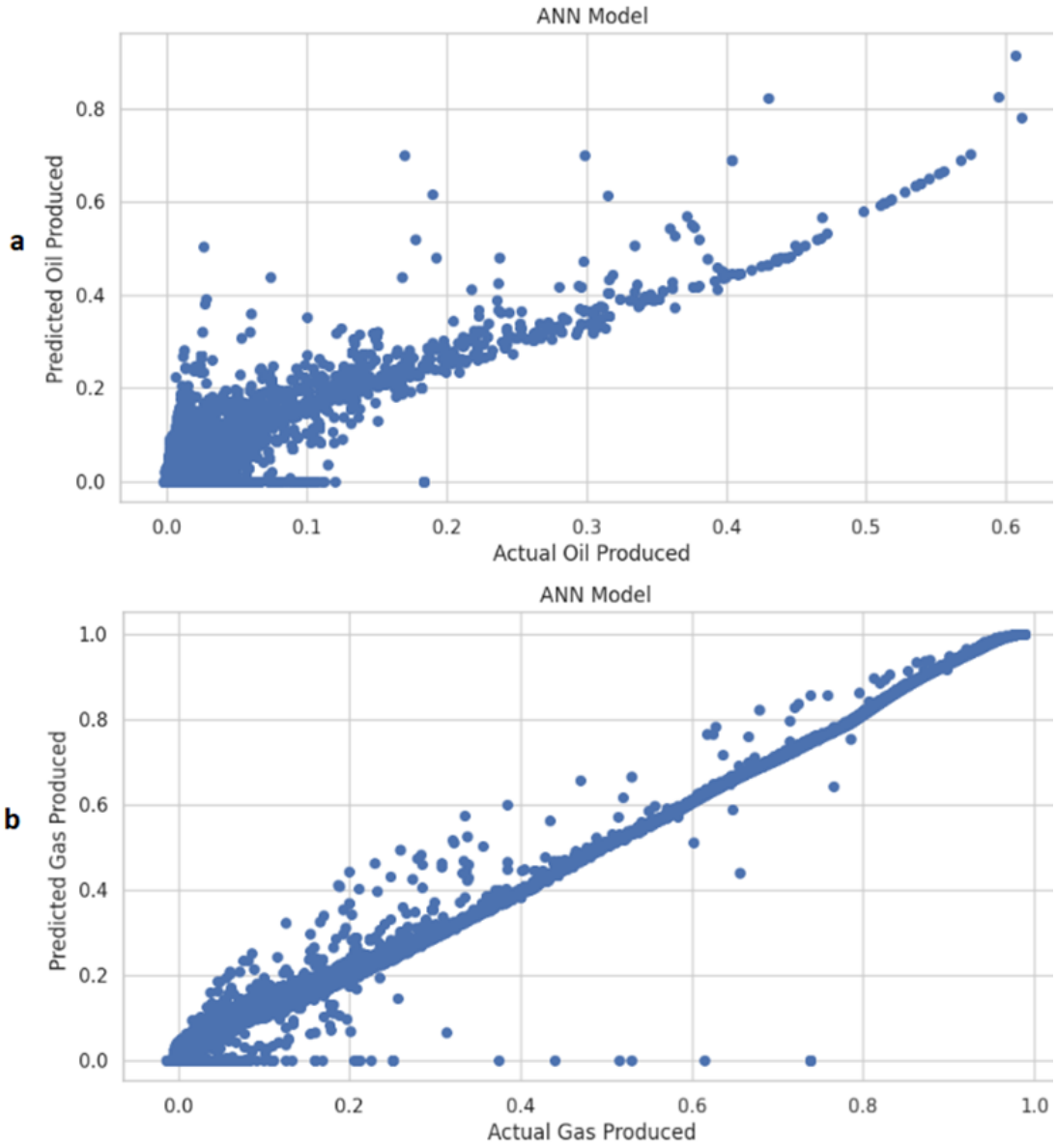

Figure 4.8. (a) The ANN model's predicted and actual oil production. (b) The ANN model's predicted and actual gas production.

4.6.2. Recurrent neural networks model result

Table 4.4 shows the RNN performance for oil and gas is evaluated by MAE**,** MSE and R^2. The experimentation started with exploring different parameters and hidden layer configurations to enhance result accuracy.

Table 4.4. Result of recurrent neural network mode.

RNN Model	MAE	MSE	R^2
Oil	0.002320	0.00045	0.898655
Gas	0.001073	0.00026	0.940912

Figures 4.9 and 4.10 show the oil and gas production training and validation loss across epochs in the RNN model. These measurements provide important information about the accuracy and performance of the RNN model.

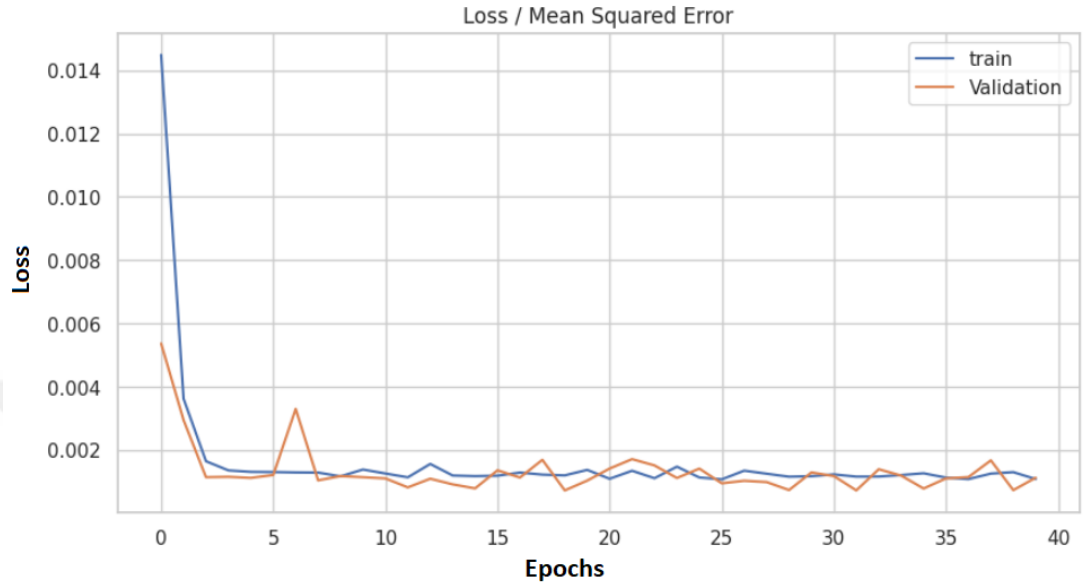

Figure 4.9. The RNN model's training and validation losses for oil production.

Figure 4.10. The RNN model's training and validation losses for gas production.

Figure 4.11a shows the relationship between the forecast oil produced and the actual oil in the RNN model, and Figure 4.11b shows the relationship between the forecast gas produced and the actual gas produced in the RNN model.

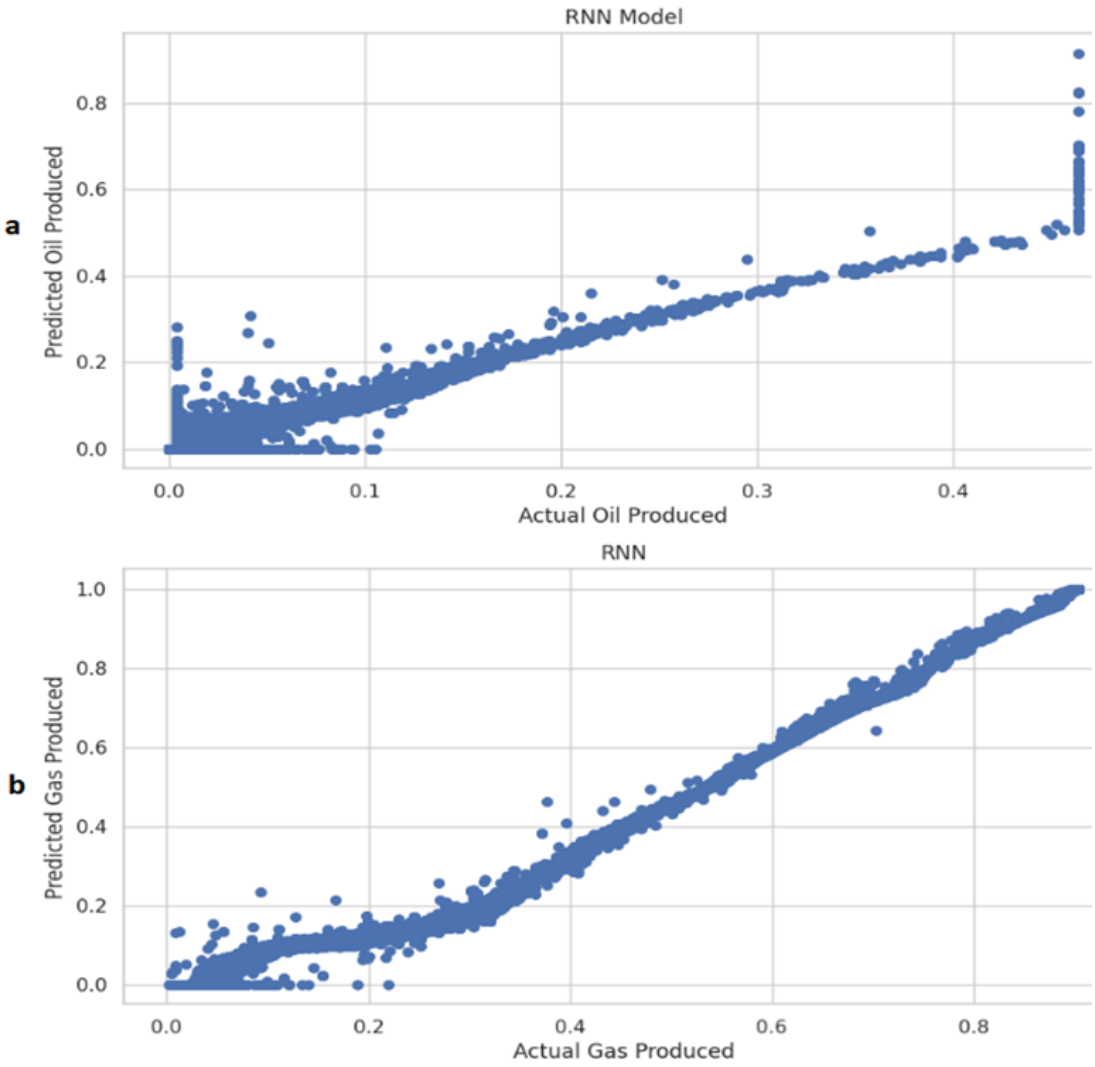

Figure 4.11. (a) The RNN model's predicted and actual oil production. (b) The RNN model's predicted and actual gas production.

4.6.3. Result of long short-term memory model

Table 4.5 shows the LSTM model's oil and gas performance metrics using R^2, MAE and MSE.

Table 4.5. Result of Long Short-Term Memory model.

LSTM Model	MAE	MSE	R^2
Oil	0.002210	0.000686	0.933324
Gas	0.001854	0.000031	0.983207

The LSTM model demonstrates excellent performance in predicting oil and gas. The high R^2 values indicate that the model successfully accounts for a significant portion

of the data variability. Figure 4.12a and Figure 4.12b show the oil and gas production validation loss and training across epochs in LSTM model.

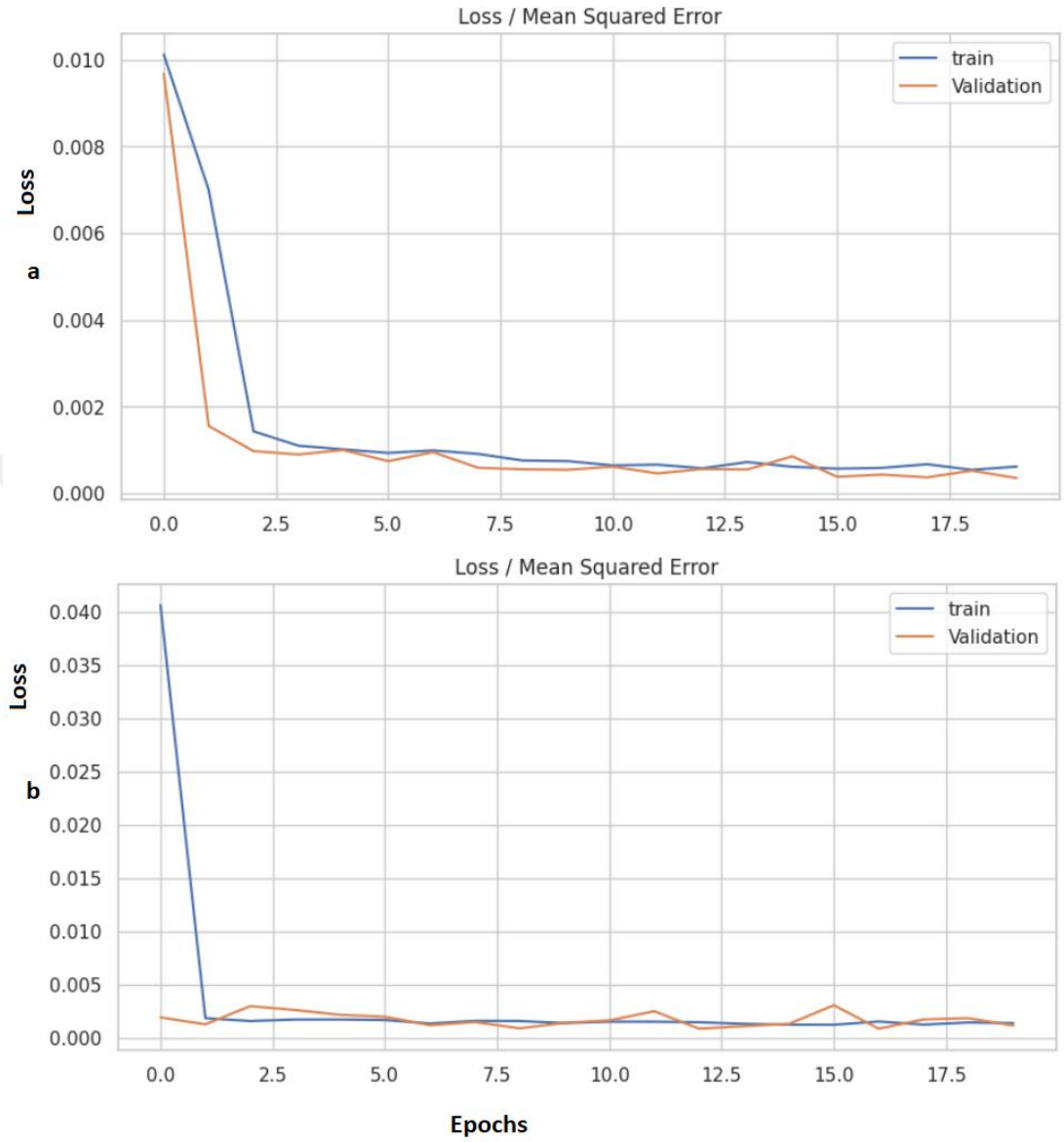

Figure 4.12. (a) The LSTM model's training and validation losses for oil production. (b) The LSTM model's training and validation losses for gas production.

Figure 4.13a shows the similarity between the forecast oil produced and actual oil in the LSTM model, while Figure 4.13b shows the similarity between the forecast gas produced and the actual gas produced in the LSTM model.

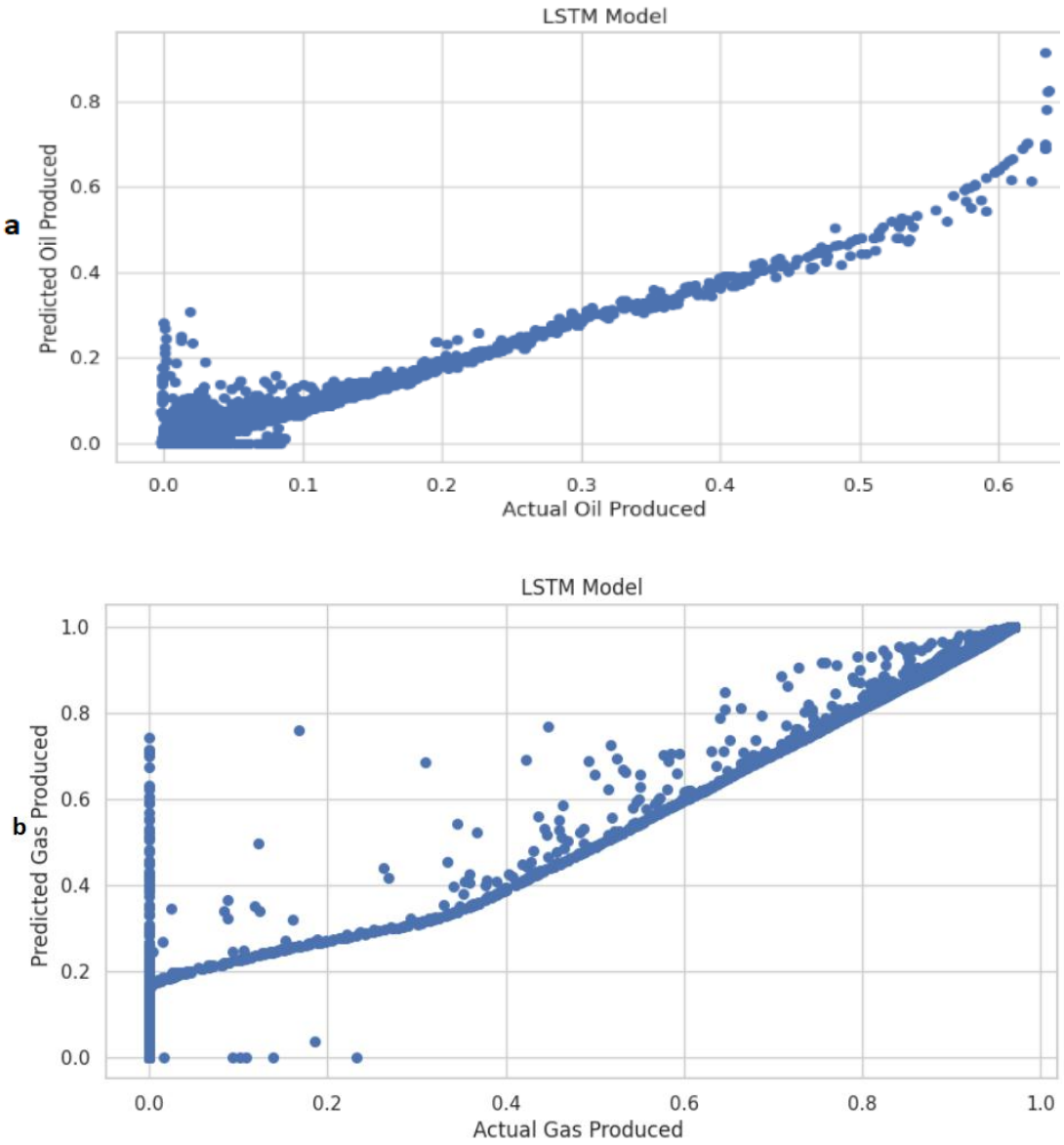

Figure 4.13. (a) The LSTM model's predicted and actual oil production. (b) The LSTM model's predicted and actual gas production.

Figure 4.14 shows the R^2 value results of DL models. This evaluation compares the R^2 values for oil and gas in deep learning models. Additionally, the LSTM model indicates higher accuracy compared to other deep learning models in forecasting oil and gas production.

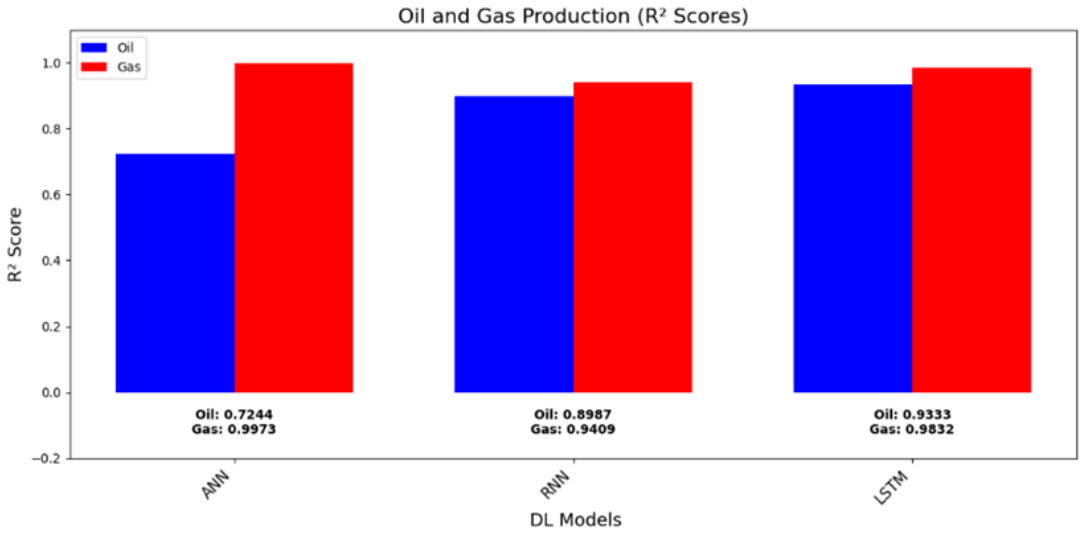

Figure 4.14. Analyzing $\mathbf{R^2}$ values for oil and gas in deep learning models.

4.7. Comparison Between the Results of ML, Ensemble Learning and DL Models

This experiment focuses on predicting future oil and gas using ML, ensemble learning, and DL models. This section compares the performance of eleven models: DTR, RFR, KNN, XGBoost , GBR, Bagging, Boosting, Stacking, ANN, RNN, and LSTM based on their training performance. Analyzing the related training performance is essential to ascertain the potential for great prediction performance in these models. We evaluate training performance of the models using the R^2 metric to provide a more precise analysis.

4.7.1. The result of oil production prediction

- The stacked model obtained the highest score of 0.973943, indicating a very good correlation between the model predictions and the real oil production data, indicating that ensemble learning can increase accuracy.
- Bagging and RFR models also demonstrate high R^2 values of 0.964901 and 0.960828, respectively, suggesting strong predictive capabilities.

4.7.2. The result of gas production prediction

- The Stacked model outperformed all other models, achieving a remarkable R^2 value of 0.999808, indicating an extremely strong correlation between the actual gas production data and the model's predictions.

- All models have slightly high R^2 values with scores above 0.94, suggesting good predictive performance.
- Across all models, the gas data slightly better predictive accuracy compared to the oil data, highlighting its superior performance for forecasting in this thesis.

Overall, these results highlight the effectiveness of various ML, ensemble learning, and DL models in forecasting both oil and gas production.

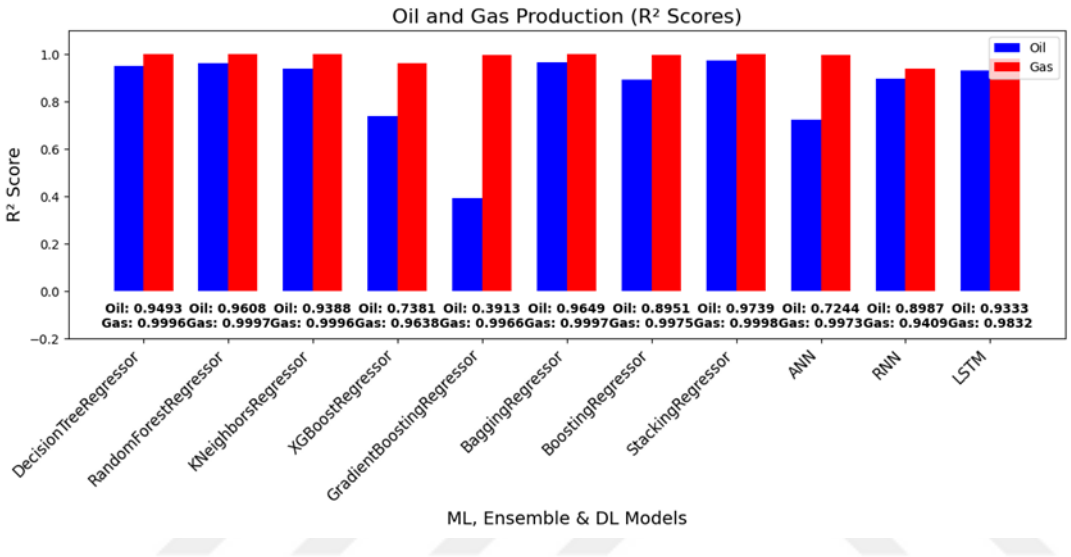

Figure 4.15. ML, ensemble learning and, DL model results.

According to the results shown in Figure 4.15 above, the stacked model is the most effective model in this thesis for oil and gas production prediction.

4.8. A List of Previous Studies with Our Proposed Techniques

Table 4.6 presents the results of the proposed methods 2024 and previous studies. The accuracy summary of the suggested methods utilizing machine learning, ensemble learning, and deep learning models for feature extraction reveals higher accuracy than the previous studies.

The results show that our suggested models achieves a high accuracy level of 99% with a value, demonstrating superior accuracy compared to previous studies. Our suggested methods use the same techniques as in previous studies, but they use a different dataset, and the characteristics of the dataset may influence the choice of a particular technique.

Table 4.6. A comparison list of the final results of proposed methods with previous studies.

Reference & year	Dataset	Models	Output	R^2
Kim, M. et al. (2020) [24]	1,000 dry gas wells in the Montney in Canada	ANN	Gas	0.9121
		1D-CNN	Gas	0.9070
		LSTM	Gas	0.9102
		1D-CNN + LSTM	Gas	0.9147
M.S.Zanjani et al (2020) [25]	The well NO159-F-1C from Norwegian	ANN	Oil	0.98
		LR	Oil	0.92
		SVR	Oil	0.89
C. Tan et al. (2021) [26]	137 fractured wells in WY shale gas block in Sichuan, China	RFR	Gas	0.72
		BP	Gas	0.82
		SVR	Gas	0.74
		XGBoost	Gas	0.87
		Light GBM	Gas	0.83
		LR	Gas	0.78
G. Hui et al. (2021) [27]	Fox Creek area of Alberta	LR	Gas	0.653
		Neural Network	Gas	0.729
		XGBoost	Gas	0.794
		Extra Trees	Gas	0.809
N. M. Ibrahim et al (2022) [28]	Saudi Aramco	MLR	Oil	0.834
			Gas	0.7684
		PLR	Oil	0.966
			Gas	0.9185
		SVR	Oil	0.9659
			Gas	0.9185
		DTR	Oil	0.9225
			Gas	0.9236
		RFR	Oil	0.9355
			Gas	0.9247
		XGBoost	Oil	0.9561
			Gas	0.9336
		ANN	Oil	0.9697
			Gas	0.9185
		RNN	Oil	0.9785
			Gas	0.8787
L. Mai-Cao et.al (2022) [29]	In the oilfield X, Southern Vietnam	RFR	Oil	0.87
		GBR	Oil	0.85
		KNN	Oil	0.91
		SVR	Oil	0.96
		MLP	Oil	0.93
		CNN	Oil	0.88
		GRU	Oil	0.90
		RNN	Oil	0.92
A. E. Al-Aghbari et.al(2022) [30]	The data from the Volve oil field was released by Equinor in 2018.	GA-TCN-LSTM	Oil	0.93
		TCN	Oil	0.92
		LSTM	Oil	0.91
		GRU	Oil	0.92
		RNN	Oil	0.92
W. Liu, Z. Chen et.al (2023) [31]	Data collected from CNPC	ANN	Oil	0.795
		XGB	Oil	0.857
S. Hosseini et.al (2023) [32]	The Volve oil field from Norwegian	LSTM	Oil	0.99
		CNN	Oil	0.98
		LRM	Oil	0.94
L. Song et al (2023) [33]	The data from 394 oil wells collected from offshore oilfields in China	LR	Oil	0.39
		XGBOOST	Oil	0.95
		LightGBM	Oil	0.83
		BP	Oil	0.44
		LSTM	Oil	0.85

Table 4.6. (Continued) A comparison list of the final results of proposed methods with previous studies.

Reference & year	Dataset	Models	Output	R^2
N. Liu et al. (2022) [34]	The data from a shale gas survey well in Xinjiang, north-western China	SVR	Gas	0.5574
		RFR	Gas	0.8718
		ETR	Gas	0.9268
		GBR	Gas	0.8663
		LightGBM	Gas	0.8777
		XGB	Gas	0.8059
		Stacking Model	Gas	0.9568
F. Ye et al. (2024) [35]	The data from China over 200 wells in Gas Field A and 207 wells in Shale Gas Field B	RFR	Gas	0.8898
		ETR	Gas	0.9468
		LightGBM	Gas	0.9045
		GBR	Gas	0.9109
		LR	Gas	0.8486
		Blending Model	Gas	0.9524
Y. Hou. et al. (2024) [36]	Data from 242 shale gas wells in China	RFR	Gas	0.39
		SVM	Gas	0.86
		LightGBM	Gas	0.89
		Stacking Model	Gas	0.92
Proposed methods in 2024	The dataset contains annual production for more than 75,000 oil and gas wells in New York State, USA	DTR	Oil	0.949276
			Gas	0.999598
		RFR	Oil	0.960828
			Gas	0.999733
		KNN	Oil	0.938847
			Gas	0.999580
		XGBRegressor	Oil	0.738127
			Gas	0.963784
		GBR	Oil	0.391319
			Gas	0.996638
		Bagging Model	Oil	0.964901
			Gas	0.999724
		Boosting Model	Oil	0.895139
			Gas	0.997543
		Stacking Model	Oil	0.973943
			Gas	0.999808
		ANN	Oil	0.724388
			Gas	0.997309
		RNN	Oil	0.898655
			Gas	0.940912
		LSTM	Oil	0.933324
			Gas	0.983207

As seen from the Table 4.6 above, several studies have explored various ML, ensemble learning and DL models for forecasting oil and gas production, demonstrating the evolution of techniques and their effectiveness. For instance, L. Song et al. (2023) utilized data from 394 offshore oil wells in China, achieving R^2 scores ranging from 0.39 to 0.95 for oil production using models like LR, XGBoost, LightGBM, BP, and LSTM.

N. Liu et al. (2022) analyzed shale gas well data from Xinjiang, China, achieving R^2 scores between 0.5574 and 0.9568 using models such as SVR, RFR, ETR, GBR, LightGBM, XGB, and a stacking model. F. Ye et al. (2024) applied similar models to over 400 wells in two different gas fields (A and B) in China, reaching R^2 scores up

to 0.9524, particularly with their blending model. Y. Hou et al. (2024) focused on 242 shale gas wells, with their stacking model achieving an R^2 of 0.92.

In comparison, the proposed methods in this thesis (2024) utilized an extensive dataset of over 75,000 oil and gas wells from New York State, USA, incorporating a diverse set of machine learning, ensemble learning, and deep learning models, including DTR, RFR, KNN, XGBRegressor, GBR, Bagging, Boosting, Stacking, ANN, RNN, and LSTM. The proposed approach achieved significantly higher R^2 scores, particularly for gas production, with values as high as 0.999808 using a stacking model. For oil production, R^2 scores reached up to 0.973943, outperforming previous studies. This comprehensive evaluation underscores the superiority of the proposed methods, especially in handling large datasets and achieving near-perfect prediction accuracy for both oil and gas production, highlighting their potential for advancing production forecasting.

5. CONCLUSION AND FUTURE WORK

5.1. Conclusion

This thesis introduced machine learning, ensemble learning and deep learning models for the petroleum industry's oil and gas production.

- The goal of this thesis is to develop a more accurate and efficient production prediction model using ML and DL techniques compared to traditional decline curve analysis and numerical reservoir simulation models. The thesis emphasizes the significance of predicting oil and gas production for a number of reasons, including making informed decisions, estimating reserves, optimizing production, assessing market dynamics, predicting future trends, project commercial viability analysis, and optimizing recovery rates, costs, and operational efficiency.
- The fundamental concept is to create a straight forward and readily applicable ML and DL models with the goal of facilitating fast and well- informed decision-making.
- This thesis developed and evaluated eleven artificial intelligence techniques for production prediction using machine learning, ensemble learning, and deep learning models, including three ensemble learning models: Bagging, Boosting, and Stacking models and five machine learning models: DTR, RFR, KNN, XGboost, and GBR. Three deep learning models: ANN, RNN, and LSTM were used to evaluate the prediction models' performance on the testing data would be assessed with metrics such as MSE, MAE, and R^2.
- The results indicated that ensemble learning techniques achieved higher on the dataset with the most effective performance.
- The stacking model gets high results with R^2 values of 99% for both outputs oil and gas. The production indicates a strong correlation between actual data and model predictions.

- It is essential to recognize that the best model choice may differ based on the distinct characteristics of the data being examined. Hence, conducting experiments with several models and evaluating their efficacy is crucial to determine the most appropriate choice.

5.2. Future Work

One of our goals is to create a system that automatically chooses ML, ensemble learning, or DL models according to the properties and kind of the dataset, the algorithm will be chosen depending on the data that is available. In future works:

- Aims to utilize more specific datasets, focusing on individual fields or wells and the factors influencing their production.
- Applying the same methodologies to oil resources and fields in Iraq or Turkey and training them on verification and production operations for oil wells and well head pressure processing.
- Develop and assess hybrid models that amalgamate various ML, ensemble, and DL methodologies to capitalize on their advantages and enhance forecasting precision.
- Consolidate data from diverse sources, including geographical data, seismic data, and production records, to improve model efficacy.
- Incorporating Time-Series Analysis: Utilize advanced time-series models such as Temporal Convolutional Networks (TCN) or (ARIMA) AutoRegressive Integrated Moving Average in conjunction with deep learning techniques to effectively capture temporal dependencies in production data.
- Training the models on images instead of numerical data.

REFERENCES

[1] British Petroleum. (2021). Statistical Review of World Energy. BP Global. https://www.bp.com.

[2] Speight, J. G. (2014). Handbook of Petroleum Refining.https://www.academia.edu/63659108/Handbook_of_Petroleum_Refining

[3] NEO Jan 29, 2020 " OIL & GAS: IS UPSTREAM OR DOWNSTREAM RIGHT FOR ME? " https://www.texvyn.in/post/career-in-oil-and-gas.

[4] R. McClay September 25, 2024 " How the Oil and Gas Industry Works"https://www.investopedia.com/investing/oil-gas-industry-overview/

[5] C. LU, H, Jiang, and J. Yang, " Shale oil production prediction and fracturing optimization based on machine learning" October 2022 https://doi.org/10.1016/j.petrol.2022.110900.

[6] Fetkovich, M. J. (1980). "Decline Curve Analysis Using Type Curves." Journal of Petroleum Technology, https://doi.org/10.2118/4629-PA.

[7] Shahab Kareem, " Metaheuristic algorithms in optimization and its application: a review," JAREE (Journal on Advanced Research in Electrical Engineering) 6(1), April 2022, DOI:10.12962/jaree.v6i1.216.

[8] C. S. W. Ng, A. J. Ghahfarokhi, and M. N. Amar, "Well production forecast in Volve field: Application of rigorous machine learning techniques and metaheuristic algorithm," Journal of Petroleum Science and Engineering, vol. 208, p. 109468, 2022, doi: https://doi.org/10.1016/j.petrol.2021.109468.

[9] Mohaghegh, S. D. (2017). "Machine Learning Applications in Reservoir Engineering: Part 1." Journal of Petroleum Technology, 69(6), 70-77. DOI: 10.2118/0617-0070-JPT.

[10] J. X. Chen, H. L. Wang, and K. Zhao, "Comparative evaluation of machine learning techniques for hydrocarbon reservoir prediction," Energies, vol. 14, no. 3, p. 806, 2021, doi: https://doi.org/10.3390/en14030806.

[11] Abou-Sayed, A. S. (2021). "AI in the Petroleum Industry." Society of Petroleum Engineers AI Newsletter. https://www.spe.org.

[12] Russell, S., & Norvig, P. (2021). Artificial Intelligence: A Modern Approach (4th Edition). Pearson.

[13] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. http://www.deeplearningbook.org.

[14] D. Koroteev and Z. Tekic, "Artificial intelligence in oil and gas upstream: Trends, challenges, and scenarios for the future," Energy and AI, vol. 3, p. 100041, 2021, doi: https://doi.org/10.1016/j.egyai.2020.100041.

[15] L. Kuang et al., "Application and development trend of artificial intelligence in petroleum exploration and development," Petroleum Exploration and Development, vol. 48, no. 1, pp. 1-14, 2021, doi: https://doi.org/10.1016/s1876-3804(21)60001-0.

[16] Z. Tariq et al., "A systematic review of data science and machine learning applications to the oil and gas industry," Journal of Petroleum Exploration and Production Technology, pp. 1-36, 2021, doi: https://doi.org/10.1007/s13202-021-01302-2.

[17] M. A. Ruiz-Serna, G. A. Alzate-Espinosa, A. F. Obando-Montoya, and H. D. Álvarez- Zapata, "Combined artificial intelligence modeling for production forecast in a petroleum production field," CT&F-Ciencia, Tecnologia y Futuro, vol. 9, no. 1, pp. 27-35, 2019, doi: https://doi.org/10.29047/01225383.149.

[18] N. A. Sami and D. S. Ibrahim, "Forecasting multiphase flowing bottom-hole pressure of vertical oil wells using three machine learning techniques," Petroleum Research, vol. 6, no. 4, pp. 417-422, 2021, doi: https://doi.org/10.1016/j.ptlrs.2021.05.004.

[19] A. Sircar, K. Yadav, K. Rayavarapu, N. Bist, and H. Oza, "Application of machine learning and artificial intelligence in oil and gas industry," Petroleum Research, vol. 6, no. 4, pp. 379-391, 2021, doi: https://doi.org/10.1016/j.ptlrs.2021.05.009.

[20] Z. Guo, H. Wang, X. Kong, L. Shen, and Y. Jia, "Machine learning-based production prediction model and its application in Duvernay Formation," Energies, vol. 14, no. 17, p. 5509, 2021, doi: https://doi.org/10.3390/en14175509.

[21] C. Gkerekos, I. Lazakis, and G. Theotokatos, "Machine learning models for predicting ship main engine Fuel Oil Consumption: A comparative study," Ocean Engineering, vol. 188, p. 106282, 2019, doi: https://doi.org/10.1016/j.oceaneng.2019.106282.

[22] B. M. Negash and A. D. Yaw, "Artificial neural network based production forecasting for a hydrocarbon reservoir under water injection," Petroleum Exploration and Development, vol. 47, no. 2, pp. 383-392, 2020, doi: https://doi.org/10.1016/s1876- 3804(20)60055-6.

[23] S. Choubey and G. Karmakar, "Artificial intelligence techniques and their application in oil and gas industry," Artificial Intelligence Review, vol. 54, no. 5, pp. 3665-3683, 2021, doi: https://doi.org/10.1007/s10462-020-09935-1.

[24] Kim, M. "Deep Learning-Based Prediction of the Cumulative Gas Production of the Montney Formation, Canada" https://geoconvention.com/wp-content/uploads/abstracts/2020/57980-deep-learning-based-prediction-of-the-cumulative-g.pdf .

[25] M. S. Zanjani, Mohammad Abdus Salam, and Osman Kandara "Data-Driven Hydrocarbon Production Forecasting Using Machine Learning Techniques". International Journal of Computer Science and Information Security (IJCSIS),Vol. 18, No. 6, June 2020.

[26] C. Tan et al., "Fracturing productivity prediction model and optimization of the operation parameters of shale gas well based on machine learning," Lithosphere, vol. 2021, no. Special 4, p. 2884679, 2021, doi: https://doi.org/10.2113/2021/2884679.

[27] G. Hui, S. Chen, Y. He, H. Wang, and F. Gu, "Machine learning-based production forecast for shale gas in unconventional reservoirs via integration of geological and operational factors," Journal of Natural Gas Science and Engineering, vol. 94, p. 104045, 2021, doi: https://doi.org/10.1016/j.jngse.2021.104045.

[28] N. M. Ibrahim et al., "Well Performance Classification and Prediction: Deep Learning and Machine Learning Long Term Regression Experiments on Oil, Gas, and Water Production," Sensors, vol. 22, no. 14, p. 5326, 2022,doi: https://doi.org/10.3390/s22145326. 14

[29] L. Mai-Cao and H. Truong-Khac, "A Comparative Study on Different Machine Learning Algorithms for Petroleum Production Forecasting," Improved Oil and Gas Recovery, vol. 6, 07/25 2022, doi: https://doi.org/10.14800/iogr.1205.

[30] A. E. Al-Aghbari and B. K. B. Lee, "Multivariate Time Series Forecasting of Oil Production Based on Ensemble Deep Learning and Genetic Algorithm," Available at SSRN 4460174, 2023, doi: https://doi.org/10.2139/ssrn.4460174.

[31] Wei Liu , Zhangxin Chen , Yuan Hu , Liuyang Xu " A systematic machine learning method for reservoir identification and production prediction" https://doi.org/10.1016/j.petsci.2022.09.002.

[32] S. Hosseini and T. Akilan, "Advanced Deep Regression Models for Forecasting Time Series Oil Production," arXiv preprint arXiv:2308.16105, 2023.

[33] Laiming Song, Chunqiu Wang, Chuan Lu1, Shuo Yang, Chaodong Tan " Machine Learning Model of Oilfield Productivity Prediction and Performance Evaluation" https://doi:10.1088/1742-6596/2468/1/012084.

[34] Naipeng Liu · Hui Gao · Zhen Zhao · Yule Hu · Longchen Duan" A stacked generalization ensemble model for optimization and prediction of the gas well rate of penetration: a case studyin Xinjiang" https://doi.org/10.1007/s13202-021-01402-z.

[35] Fan Ye, Xiaobo Li, Nan Zhang and Feng Xu" Prediction of Single-Well Production Rate after Hydraulic Fracturing in Unconventional Gas Reservoirs Based on Ensemble Learning Model" https://doi.org/10.3390/pr12061194.

[36] Yiting Hou, Qun Luo and Yixin Zhou (2024) " Application of Stacking Model with Feature Weighting Method in Shale Gas Production Prediction" https://doi:10.1109/ICCEA62105.2024.10603544.

[37] R. Y. Choi, A. S. Coyner, J. Kalpathy-Cramer, M. F. Chiang, and J. P. Campbell, "Introduction to machine learning, neural networks, and deep learning," Translational Vision Science & Technology, vol. 9, no. 2, pp. 14-14, 2020.

[38] B. Mahesh, "Machine learning algorithms-a review," International Journal of Science and Research (IJSR).[Internet], vol. 9, pp. 381-386, 2020.

[39] Y. Başţanlar and M. Özuysal, "Introduction to machine learning," miRNomics: MicroRNA biology and computational analysis, pp. 105-128, 2014.

[40] D. S. W. Ting et al., "Artificial intelligence and deep learning in ophthalmology,"British Journal of Ophthalmology, vol. 103, no. 2, pp. 167-175, 2019.

[41] P. F. Orrù, A. Zoccheddu, L. Sassu, C. Mattia, R. Cozza, and S. Arena, "Machine learning approach using MLP and SVM algorithms for the fault prediction of a centrifugal pump in the oil and gas industry," Sustainability, vol. 12, no. 11, p. 4776, 2020, doi: https://doi.org/10.3390/su12114776.

[42] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "Machine learning and deep learning approaches for cybersecuriy: A review," IEEE Access, 2022.

[43] S. Ray, "A quick review of machine learning algorithms," in 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon), 2019: IEEE, pp. 35-39, doi: https://doi.org/10.1109/comitcon.2019.8862451.

[44] A. Géron, Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. " O'Reilly Media, Inc.", 2022.

[45] K. M. Hanga and Y. Kovalchuk, "Machine learning and multi-agent systems in oil and gas industry applications: A survey," Computer Science Review, vol. 34, p. 100191, 2019, doi: https://doi.org/10.1016/j.cosrev.2019.08.002.

[46] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," SN computer science, vol. 2, no. 3, p. 160, 2021.

[47] G. Carleo et al., "Machine learning and the physical sciences," Reviews of Modern Physics, vol. 91, no. 4, p. 045002, 2019, doi: https://doi.org/10.1103/revmodphys.91.045002.

[48] Y. H. Liu, Python machine learning by example. Packt Publishing Ltd, 2017.

[49] S. Dridi, "Supervised Learning-A Systematic Literature Review," 2021.

[50] V. Nasteski, "An overview of the supervised machine learning methods," Horizons. b,vol. 4, pp. 51-62, 2017.

[51] J. Alzubi, A. Nayyar, and A. Kumar, "Machine learning from theory to algorithms: an overview," in Journal of physics: conference series, 2018, vol. 1142: IOP Publishing, p. 012012.

[52] S. Sah, "Machine learning: a review of learning types," 2020. https://doi:10.20944/preprints202007.0230.v1.

[53] S. S. Dash, S. K. Nayak, and D. Mishra, "A review on machine learning algorithms," Intelligent and Cloud Computing: Proceedings of ICICC 2019, Volume 2, pp. 495- 507, 2020.

[54] M. M. Shawkat, A. R. B. Risal, N. J. Mahdi, Z. Safari, M. H. Naser, and A. W. Al Zand, "Fluid Flow Behavior Prediction in Naturally Fractured Reservoirs Using Machine Learning Models," Complexity, vol. 2023, 2023, doi: https://doi.org/10.1155/2023/7953967.

[55] Gurucharan M K Jul 15, 2020 " Machine Learning Basics: Decision Tree Regression" https://towardsdatascience.com/machine-learning-basics-decision-tree-regression-1d73ea003fda .

[56] A. Al-Fakih, A. F. Ibrahim, S. Elkatatny, and A. Abdulraheem, "Estimating electrical resistivity from logging data for oil wells using machine learning," Journal of Petroleum Exploration and Production Technology, vol. 13, no. 6, pp. 1453-1461, 2023, doi: https://doi.org/10.1007/s13202-023-01617-2.

[57] G. S. Ohannesian and E. J. Harfash, "Epileptic Seizures Detection from EEG Recordings Based on a Hybrid system of Gaussian Mixture Model and Random Forest Classifier," Informatica, vol. 46, no. 6, 2022, doi: https://doi.org/10.31449/inf.v46i6.4203.

[58] A. K. Ali and A. M. Abdullah, "Fake accounts detection on social media using stack ensemble system," International Journal of Electrical & Computer Engineering (2088-8708), vol. 12, no. 3, 2022, doi: https://doi.org/10.11591/ijece.v12i3.pp3013- 3022.

[59] M. Y. Khan, A. Qayoom, M. S. Nizami, M. S. Siddiqui, S. Wasi, and S. M. K.-u.-R. Raazi, "Automated prediction of Good Dictionary EXamples (GDEX): a comprehensive experiment with distant supervision, machine learning, and word embedding-based deep learning techniques," Complexity, vol. 2021, pp. 1-18, 2021, doi: https://doi.org/10.1155/2021/2553199.

[60] R. Zhang, Y. Li, A. T. Goh, W. Zhang, and Z. Chen, "Analysis of ground surface settlement in anisotropic clays using extreme gradient boosting and random forest regression models," Journal of Rock Mechanics and Geotechnical Engineering, vol. 13, no. 6, pp. 1478-1484, 2021.

[61] B. Lu and Y. He, "Evaluating empirical regression, machine learning, and radiative transfer modelling for estimating vegetation chlorophyll content using bi-seasonal hyperspectral images," Remote Sensing, vol. 11, no. 17, p. 1979, 2019, doi: https://doi.org/10.3390/rs11171979.

[62] D. K. Seo, Y. H. Kim, Y. D. Eo, W. Y. Park, and H. C. Park, "Generation of radiometric, phenological normalized image based on random forest regression for change detection," Remote Sensing, vol. 9, no. 11, p. 1163, 2017, doi: https://doi.org/10.3390/rs9111163.

[63] P. Jain, A. Choudhury, P. Dutta, K. Kalita, and P. Barsocchi, "Random forest regression-based machine learning model for accurate estimation of fluid flow in curved pipes," Processes, vol. 9, no. 11, p. 2095, 2021, doi: https://doi.org/10.3390/pr9112095.

[64] C. Hu, G. Jain, P. Zhang, C. Schmidt, P. Gomadam, and T. Gorka, "Data-driven method based on particle swarm optimization and k-nearest neighbor regression for estimating capacity of lithium-ion battery," Applied Energy, vol. 129, pp. 49-55, 2014.

[65] M. E. Javanmard and S. Ghaderi, "A hybrid model with applying machine learning algorithms and optimization model to forecast greenhouse gas emissions with energy market data," Sustainable Cities and Society, vol. 82, p. 103886, 2022.

[66] S. Islamov, A. Grigoriev, I. Beloglazov, S. Savchenkov, and O. T. Gudmestad, "Research risk factors in monitoring well drilling—A case study using machine learning methods," Symmetry, vol. 13, no. 7, p. 1293, 2021.

[67] S. B. Imandoust and M. Bolandraftar, "Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background," International journal of engineering research and applications, vol. 3, no. 5, pp. 605-610, 2013.

[68] H. Singh, Y. Seol, and E. M. Myshakin, "Prediction of gas hydrate saturation using machine learning and optimal set of well-logs," Computational Geosciences, vol. 25, pp. 267-283, 2021.

[69] N. U. Moroff, E. Kurt, and J. Kamphues, "Machine Learning and statistics: A Study for assessing innovative demand forecasting models," Procedia Computer Science, vol. 180, pp. 40-49, 2021, doi: https://doi.org/10.1016/j.procs.2021.01.127.

[70] J. Avanijaa, "Prediction of house price using xgboost regression algorithm," Turkish Journal of Computer and Mathematics Education (TURCOMAT), vol. 12, no. 2, pp. 2151–2155-2151–2155, 2021.

[71] T. Bikmukhametov and J. Jäschke, "Oil production monitoring using gradient boosting machine learning algorithm," Ifac-Papersonline, vol. 52, no. 1, pp. 514-519, 2019, doi: https://doi.org/10.1016/j.ifacol.2019.06.114.

[72] C. Cao, P. Jia, L. Cheng, Q. Jin, and S. Qi, "A review on application of data-driven models in hydrocarbon production forecast," Journal of Petroleum Science and Engineering, vol. 212, p. 110296, 2022, doi: https://doi.org/10.1016/j.petrol.2022.110296.

[73] K. Tissaoui, T. Zaghdoudi, A. Hakimi, and M. Nsaibi, "Do Gas Price and Uncertainty Indices Forecast Crude Oil Prices? Fresh Evidence Through XGBoost Modeling," Computational Economics, pp. 1-25, 2022.

[74] M. Zou, W.-G. Jiang, Q.-H. Qin, Y.-C. Liu, and M.-L. Li, "Optimized XGBoost model with small dataset for predicting relative density of Ti-6Al-4V parts manufactured by selective laser melting," Materials, vol. 15, no. 15, p. 5298, 2022, doi: https://doi.org/10.3390/ma15155298.

[75] David Cournapeau and Matthieu Brucher 2007 "Gradient Boosting regression" https://scikit-learn.org/stable/auto_examples/ensemble/plot_gradient_boosting_regression.html.

[76] A. Ali December 2023 “Gradient Boosting Machine Learning Algorithm” https://DOI: 10.13140/RG.2.2.31609.65123.

[77] Friedman, J. H. (2001). "Greedy Function Approximation: A Gradient Boosting Machine." Annals of Statistics, 29(5), 1189–1232. DOI: 10.1214/aos/1013203451.

[78] E. Budu June 5, 2024 " Bagging, Boosting, and Stacking in Machine Learning" https://www.baeldung.com/cs/bagging-boosting-stacking-ml-ensemble-models.

[79] Brijesh Soni Apr 28, 2023 "Understanding Boosting in Machine Learning: A Comprehensive Guide "https://medium.com/@brijesh_soni/understanding-boosting-in-machine-learning-a-comprehensive-guide-bdeaa1167a6.

[80] Zixuan Zhang Jun 26, 2019 Published in Towards Data Science " Boosting Algorithms Explained" https://towardsdatascience.com/boosting-algorithms-explained-d38f56ef3f30.

[81] M. Kalirane10 Oct, 2024 "Ensemble Learning in Machine Learning: Bagging, Boosting and Stacking" https://www.analyticsvidhya.com/blog/2023/01/ensemble-learning-methods-bagging-boosting-and-stacking/.

[82] j. T. point. "Difference between Machine Learning and Deep Learning." https://www.javatpoint.com/machine-learning-vs-deep-learning (accessed.

[83] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning,"Electronic Markets, vol. 31, no. 3, pp. 685-695, 2021.

[84] F. Chollet, Deep learning with Python. Simon and Schuster, 2021.

[85] A. I. Georgevici and M. Terblanche, "Neural networks and deep learning: a brief introduction," Intensive Care Medicine, vol. 45, no. 5, pp. 712-714, 2019.

[86] S. Sharma, and A. Athaiya, "Activation functions in neural networks,"Towards Data Sci, vol. 6, no. 12, pp. 310-316, 2017.

[87] J. Feng and S. Lu, "Performance analysis of various activation functions in artificial neural networks," in Journal of physics: conference series, 2019, vol. 1237, no. 2: IOP Publishing, p. 022030.

[88] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete, "A survey on modern trainable activation functions," Neural Networks, vol. 138, pp. 14-32, 2021.

[89] Y. Yu, K. Adu, N. Tashi, P. Anokye, X. Wang, and M. A. Ayidzoe, "Rmaf: Relu- memristor-like activation function for deep learning," IEEE Access, vol. 8, pp. 72727- 72741, 2020.

[90] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures,"IEEE access, vol. 7, pp. 53040-53065, 2019.

[91] H. Kukreja, N. Bharath, C. Siddesh, and S. Kuldeep, "An introduction to artificial neural network," Int J Adv Res Innov Ideas Educ, vol. 1, pp. 27-30, 2016.

[92] R. Al-Shabandar, A. Jaddoa, P. Liatsis, and A. J. Hussain, "A deep gated recurrent neural network for petroleum production forecasting," Machine Learning with Applications, vol. 3, p. 100013, 2021.

[93] M. L. A. I. Recipes. "Recurrent Neural Networks for Sentiment Analysis." https://www.gosmar.eu/machinelearning/2020/08/23/recurrent-neural-networks-for- sentiment-analysis/ (accessed.

[94] A. Mathew, P. Amudha, and S. Sivakumari, "Deep learning techniques: an overview," Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020, pp. 599-608, 2021.

[95] c. s. blog. "Understanding LSTM Networks." https://colah.github.io/posts/2015-08- Understanding-LSTMs/ (accessed.

[96] W. Zha et al., "Forecasting monthly gas field production based on the CNN-LSTM model," Energy, p. 124889, 2022.

[97] U. P. Iskandar and M. Kurihara, "Long Short-term Memory (LSTM) Networks for Forecasting Reservoir Performances in Carbon Capture, Utilisation, and Storage (CCUS) Operations," Scientific Contributions Oil and Gas, vol. 45, no. 1, pp. 35-51, 2022.

[98] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," PeerJ Computer Science, vol. 7, p. e623, 2021, doi: https://doi.org/10.7717/peerj-cs.623.

[99] D. Zhang, "A coefficient of determination for generalized linear models," The American Statistician, vol. 71, no. 4, pp. 310-316, 2017,doi: https://doi.org/10.1080/00031305.2016.1256839.

[100] D. world Publisher State of New York May 3, 2024 "Oil and Gas Annual Production: Beginning 2001" https://catalog.data.gov/dataset/oil-and-gas-annual-production-beginning-2001.

[101] Y.-J. Hu, T.-H. Ku, R.-H. Jan, K. Wang, Y.-C. Tseng, and S.-F. Yang, "Decision tree- based learning to predict patient controlled analgesia consumption and readjustment," BMC medical informatics and decision making, vol. 12, pp. 1-15, 2012.

[102] A. M. Melesse et al., "River water salinity prediction using hybrid machine learning models," Water, vol. 12, no. 10, p. 2951, 2020.

[103] S. Dridi, "Supervised learning-a systematic literature review," preprint, Dec, 2021.

[104] S. Olszewski et al., "Regression Modeling for Monitoring Organochlorine Pesticide Residues," in MoMLeT+ DS, 2023, pp. 204-218.

[105] Parth Shukla 28 Nov, 2022 " Stacking Algorithms in Machine Learning " https://www.analyticsvidhya.com/blog/2022/11/stacking-algorithms-in-machine-learning/ .

[106] I.Goodfellow and Y. Bengio and A. Courville 2016 part 3/20 Deep Generative Models " Deep Learning " https://www.deeplearningbook.org/.

[107] M. S. Alhajeri, J. Luo, Z. Wu, F. Albalawi, and P. D. Christofides, "Process structure- based recurrent neural network modeling for predictive control: A comparative study," Chemical Engineering Research and Design, vol. 179, pp. 77-89, 2022.

[108] S. A. Al-hilfi, "Forecasting crude oil Prices in iraq using neural networks," 2022.

CURRICULUM VITAE

Name Surname : Azhar Naji Muhajir ALYAHYA

EDUCATION:

- **Undergraduate :** 2007, Baghdad University, Computer Engineering Department.
- **Graduate :** 2024, Sakarya University, Software Engineering Department, Software Engineering.

PROFESSIONAL EXPERIENCE AND AWARDS:

- 2008-2017 Midland Refineries Company MRC / Daura Refineries / Baghdad worked as a desktop applications programmer.
- 2018 Oil Exploration Company OEC / Baghdad worked as a desktop applications programmer.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- Azhar Alyahya, Gülüzar ÇİT (2024, November). Advanced Oil and Gas Production Forecasting Using Deep Learning Techniques. 2nd International Conference on Engineering, Natural and Social Sciences ICIAS 2024 on November 22-23, 2024 in Konya, Turkey.
- Azhar Alyahya, Gülüzar ÇİT (2025, May). Enhanced Oil and Gas Production Forecasting Through Stacked generalization Ensemble Learning Technique. Sakarya University Journal of Computer and Information Sciences.