

Feed-Forward Neural Network Based Petroleum Wells Equipment Failure Prediction

Agil Yolchuyev

Department Network Systems and Services, Budapest Technology and Economics University, Budapest, Hungary
Email: yolchuyev@hit.bme.hu

How to cite this paper: Yolchuyev, A. (2023) Feed-Forward Neural Network Based Petroleum Wells Equipment Failure Prediction. *Engineering*, 15, 163-175.
<https://doi.org/10.4236/eng.2023.153013>

Received: February 14, 2023

Accepted: March 24, 2023

Published: March 27, 2023

Copyright © 2023 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In the oil industry, the productivity of oil wells depends on the performance of the sub-surface equipment system. These systems often have problems stemming from sand, corrosion, internal pressure variation, or other factors. In order to ensure high equipment performance and avoid high-cost losses, it is essential to identify the source of possible failures in the early stage. However, this requires additional maintenance fees and human power. Moreover, the losses caused by these problems may lead to interruptions in the whole production process. In order to minimize maintenance costs, in this paper, we introduce a model for predicting equipment failure based on processing the historical data collected from multiple sensors. The state of the system is predicted by a Feed-Forward Neural Network (FFNN) with an SGD and Backpropagation algorithm is applied in the training process. Our model's primary goal is to identify potential malfunctions at an early stage to ensure the production process' continued high performance. We also evaluated the effectiveness of our model against other solutions currently available in the industry. The results of our study show that the FFNN can attain an accuracy score of 97% on the given dataset, which exceeds the performance of the models provided.

Keywords

PDM, IoT, Internet of Things, Machine Learning, Sensors, Feed-Forward Neural Networks, FFNN

1. Introduction

Equipment failure can occur one or more times during the operational lifetime of the oil and gas wells [1]. This can happen for multiple reasons, starting from natural disasters such as hurricanes and snowstorms up to harsh environments

or mechanical failures of the drilling components. When equipment functioning is disrupted, it might pose a threat to employees and other components. Therefore, the number of wells that have had some sort of good barrier or integrity failure varies widely (between 1.9% and 75%) [2]. The most important safety precaution in oil facilities is isolating equipment and minimizing the effects of component failure processes [3]. However, due to the lack of knowledge on the equipment's susceptibility to failure and the cause of failure, it might be difficult to determine when and how to isolate sensitive equipment. This reason itself calls for possible early-stage identification of the failure process.

The statistics accumulated in the past decades in the oil- and gas industry have reported severe examples of component losses in wells, with significant consequences, e.g., Phillips Petroleum's failure in 1977 and Saga Petroleum's underground rupture in 1989 [4]. More than 40% [5] of these failures are directly (or indirectly) related to equipment failures during the operational process.

Due to the mission-critical nature of these processes, the oil and gas industry has already implanted thousands of sensors inside and around the physical components of well equipment systems. Raw sensor data are continuously streamed via DCS and SCADA systems measuring temperature, pressure, flow rate, vibration, and depth of drills, turbines, boilers, pumps, compressors, and injectors [6]. As part of the ETL process, extracted data itself needs to be transformed and passed from data quality tests before loading and using in models. Moreover, this process needs to run in real time due to future predictions of possible failures.

These issues triggered researchers to find new solutions to boost the digital transformation process of the oil and gas industry. For this purpose, tools like the Internet of Things (IoT), Big Data, Artificial Intelligence, and Cloud systems proved to be irreplaceable in wells and refineries. Research has been focused on optimizing the use of these technologies to make the system and processes safer.

Dhafer A. Al-Shehr presented a solution by implementing artificial neural networks (ANNs) and adaptive network-based fuzzy inference-based models for corrosion rate prediction. Using artificial intelligence (AI) approaches, he sought to develop an efficient, resilient, and accurate model for estimating the corrosion rate of the metal casing string. The artificial intelligence models were trained using a dataset of 250 data points culled from 218 wells [7].

Anomaly detection based on the sensorial data also was another research objective in [8], where researchers introduced a new combination of one-class support vector machine (SVM) and yet another segmentation algorithm (YASA). They conducted a series of empirical experiments by comparing their methodology to other approaches and applied it to benchmark issues and real-world applications including the identification of anomalies in oil platform turbomachinery. The findings demonstrate that the combination of one-class SVM and YASA outperformed the other industry-standard techniques.

Oil flow rate prediction error analysis was also a research scope of paper [9],

where the researchers evaluated the performance of the different algorithms: The following algorithms have been evaluated: Gene expression programming (GEP), Adaptive Neuro-Fuzzy Inference System (ANFIS), Radial Basis Function (RBF), Least Squares Support Vector Machine (LSSVM) and Multilayer Perceptron (MLP) on the dataset of 1037 data records. The investigation of prediction performance demonstrates that all applied algorithms obtain acceptable levels of accuracy in their forecasts; however, the MLP algorithm generates the most accurate predictions.

IoT was also implemented for pipeline leakage detection, as a real-time alerting system [10]. To compare current solutions where automation of the leakage detection was performed by using PLCs (which communicated through SCADA), in this research, a new system was introduced in order to achieve the same functionality via the real-time monitoring of the pipelines. The research was performed in the lab environment by measuring of flow rate (rate of flow of the liquid) and by using a flow sensor. The main advantage of this system it can detect even small leakage over a remote distance, which is hard to achieve through the PLCs due to the complexity of the system. The system also can use real-time in order to make a decision for critical conditions, which makes it unique to comparing other existing systems. The implementation of this in oil and gas industries will prevent accidents and due to real-time analysis of the data, the decision-making period will be significantly reduced.

It is impracticable and inefficient to analyze and process all the raw data remotely on the cloud server due to network latency and limited cloud computing capabilities. Major production safety issues may arise if abnormal data is not detected. To address this issue, a machine-learning-based edge-cloud system was presented by Feng Shi, Liping Yan, Xiang Zhao, and Richard Xian-Ke Gao [11]. For anomaly identification, the framework uses isolation forest and robust random-cut forest algorithms. The preprocessed time-series data are transmitted to cloud services for data trend prediction and missing data completion using the long short-term memory recurrent neural network technique feed along with the original sequence of historical data combined with the first-order forward difference data.

“Petroleum Analytics Learning Machine” (PALM) is a “brutally empirical” analytical system for controlling upstream and midstream oil and gas operations via IoT devices [12]. It was designed for the emerging unconventional shale oil and gas plays, in which simultaneous analysis of hundreds of IoT attributes from hundreds of horizontal wells with thousands of hydraulic fracture phases must be performed in near real-time. PALM’s predictive and prescriptive solutions combine Support Vector Machine learning, signatures, and real-time Random Forest and decision trees to drive hydraulic fractures toward becoming high rather than low oil and gas producers during the completions of horizontal shale wells. It utilizes hundreds to thousands of geological, geophysical, and engineering variables recorded in the field by the IoT and analyses their significance using a variety of ensemble learning algorithms like Support Vector Regression,

logistic regression, Bayesian models, nearest neighbors, neural networks, and deep learning networks. Companies actively use it in the petroleum industry for various tasks, including 4D seismic monitoring of production changes over time and reservoir simulation models.

To develop a monitoring system for the specific domain of the petroleum industry was the scope of many research works [13]. It was more attractive to introduce new solutions to a particular area of the industry than a full-scale solution itself. Such as one of the recent research works a new Q-learning-based pipeline monitoring system was introduced [14] to determine the activity time of sensor nodes based on their overlapping, energy, and distance to the base station. Same as the previous works, current research also concentrated on the particular issues related to sensor nodes, to predict the death time of sensor nodes and replace them at the right time.

Predictive maintenance (PdM) together with the Internet of Things (IoT) is widely used in the industry (especially in the manufacturing industry). It usually uses sensor data to optimize maintenance activities. While the topic of predictive maintenance (PdM) itself as well as machine learning (ML) for industrial systems have both been covered in various, separate papers, there is a research gap in petroleum industry.

In contrast with previous research works, in this paper, a new prediction model is introduced to determine possible failures in petroleum wells' surface and downhole equipment during the operation. This new model proves to provide better prediction performance than the traditional algorithms based on minimizing the mean square error. Our approach is mainly focused on historical data (time series dataset) which was collected from sensors around the equipment on both sides (downhole and surface) and based on this data predict that in the next steps (step represents time period) the equipment will fail or not. The prediction is given on the basis of estimating the probability of the corresponding events by using an FFNN where the training set is appropriately encoded to solve the problem.

2. Problem Formalizing

PDM is basically concerned with collecting data and estimating the operability of the system under observation. By PDM the system lifecycle can be maximized, and a significant reduction of maintenance cost can be achieved. To attain these objectives, it is essential to develop efficient methods to predict when a failure will happen. This prediction is based on estimating the probability that failure will not yet occur after M steps. The discussion below provides such an estimation method by using the predictive power of feed-forward neural networks.

Definition of the Model

In order to formalize our task, let us assume that $x(t) \in \mathbb{R}^k$ is a time series of k observations and any time series $x \in X$, where X (physical information both on

the surface and below the ground for each failure event) is a large set of N time series of the length k . Based on the observations at the dataset:

$x(t-1), x(t-2), \dots, x(t-L+1)$ the underlying challenge is to estimate the probability that the system is still fully operational in the next M steps, where is the probability is:

$$P(x(t+M) \leq a_1, x(t+K-1) \leq a_2, \dots, x(t) \leq a_n | x(t-1) = i, \dots, x(t-L+1) = j) \tag{1}$$

or

$$M : P(x(t+M) \leq a_1, x(t+K-1) \leq a_2, \dots, x(t) \leq a_n | x(t-1) = i, \dots, x(t-L+1) = j) \geq 1 - \varepsilon \tag{2}$$

Let us classify our data in two groups: $x^-(t)$ represents the training data, $x^+(t)$ validation data:

$$x^+(t) := (x(t+M), x(t+M-1), \dots, x(t)) \tag{3}$$

and

$$x^-(t) := (x(t-1), \dots, x(t-L+1)) \tag{4}$$

By introducing these two notations (2) can be formalized as

$$M : P(x^+(t) \in A | x^-(t) = (i, \dots, j)) \geq 1 - \varepsilon$$

Encoding the outcomes by the following two vectors:

$$s^{(1)} = (s_1^{(1)}, s_2^{(1)}) = (1, 0) \rightarrow ha \mathbf{x}^+ \in A$$

$$s^{(2)} = (s_1^{(2)}, s_2^{(2)}) = (0, 1) \rightarrow ha \mathbf{x}^+ \notin A$$

The labelled time series $\{x(t), s(t)\}$, $s \in \{s^{(1)}, s^{(2)}\}$ is referred to as an instance, and the ordered set of (X, S) as data set:

$$\tau^k = \{(x^-(t), s(t)), 1, \dots, N\}, s(t) \in \{s^{(1)}, s^{(2)}\}$$

The task described above is a multi-label binary classification problem that maps the time series $s(t)$ to a probability of a class $p(s = s_i), i = 0, 1$ based on the training data τ^k . As $p(s = s_i) = 1 - p(s = s_1)$ we simply write $p(s)$ instead of $p(s = s_1)$ for simplicity. If we will accept that A is a set of the thresholds, where $A = \{a_1, a_2, \dots, a_n\}$, then for the sake of an example, the evaluation of the system state can be summarized as follows: if $p(s) > a_{\zeta, \zeta=1, \dots, n}$ then the system is malfunctioning, and urgent maintenance action is required while if $p(s) \leq a_{\zeta, \zeta=1, \dots, n}$ then the system operates normally. We use a Feed Forward Neural Network (FFNN) to predict the probabilities given above. In order to achieve this, we need a special encoding technique to obtain the serried probabilities at the output of the network after learning. To optimal weights can be learned by minimizing the objective function

$$\mathbf{w}_{opt} : \min_w \frac{1}{N} \sum_{i=1}^N l(s_i, Net(\mathbf{x}_i, \mathbf{w})) \tag{5}$$

where loss function $l(s_i, Net(\mathbf{x}_i, \mathbf{w}))$ will be

$$l(s_i, Net(\mathbf{x}_i, \mathbf{w})) = \|s_i - Net(\mathbf{x}_i, \mathbf{w})\|^2 \tag{6}$$

As part of the SGD optimization back-propagation can be used in order to find the minimum of (6) [15] [16]. With the implementation, BP algorithm w_{opt} will yield

$$\mathbf{w}_{opt} : \frac{1}{N} \sum_{i=1}^N l(s_i, Net(\mathbf{x}_i, \mathbf{w})) \rightarrow E \|x^- - Net(\mathbf{x}_i, \mathbf{w})\|^2 \tag{7}$$

where

$$\begin{aligned} \mathbf{w}_{opt} : \min_w E \|x^- - Net(\mathbf{x}_i, \mathbf{w})\|^2 &\rightarrow Net(x^-, w) = E(s, x^-) \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} P(x_i^+ \in X | x^-) \\ P(x_i^+ \in X^c | x^-) \end{pmatrix} = P(x_i^+ \in X | x^-) + P(x_i^+ \in X^c | x^-) \end{aligned} \tag{8}$$

and

$$E_1 = P(x_i^+ \in X | x^-); E_2 = P(x_i^+ \in X^c | x^-)$$

To summarize (8) can be written as

$$\mathbf{w}_{opt} : \min_w E \|x^- - Net(\mathbf{x}_i, \mathbf{w})\|^2 \rightarrow E_1 + E_2 \tag{9}$$

As a result, after learning, at the output of the FFNN one can obtain the estimated conditioned probabilities once the past observations are given in the input. The system is regarded to be reliable if there are at least M steps until the failure with the probability of

$$P(x_i^+ \in X | x^-) > 1 - p \tag{10}$$

Algorithm 1 is a pseudocode¹ of our model. It also describes how the cost function search was defined for our model.

For the given multi-label binary classification task, the architecture of FFNN was described in **Figure 1**, where can be seen Input, Output and Hidden layers.

3. Experiments

This section will describe the experimental design and the training details. The results of the experiments were also presented in this section.

3.1. Training Data

In order to train our model, we used the ‘‘ConocoPhillips’’ data set from Kaggle [17], where a total of 172 features are given in this dataset, which consists of Id, target, and sensor data. Total of 100 sensors have single readings, and the remaining 7 sensors have time-based readings (each sensor among these 7 has 10 time-based readings). The actual dataset was introduced for the binary classification task where the ‘‘target’’ column has a value of 0 or 1 (‘‘target’’ value 0 indicates surface failures and value 1 indicates downhole failure). We took this dataset

¹Actual code of the model: https://github.com/agilyol/ffnn_kagle.git.

Algorithm 1. Pseudocode for FFNN algorithm.

Procedure FFNN:

Input Data:

- $x(t)$ is a time series of k observations
- $\gamma(t)$ input-output mapping value

Define Hyper-Parameters:

- Epoch
- Learning rate (η)
- Batch Size
- Class weight
- Callbacks
- Dense layer
- Dropout layer

Initialize Weight

- w_i : Initial input weight to random value

Define the Cost Function:

- Same as in Equation (9)

For i in Range(Epoch):

- Calculate E for all inputs. If E is smaller than tolerated value then exit from the loop with raising an exception
- For each input, calculate gradients for all weights, included bias weights
 - **If** length of gradient vector is smaller than given minimum border value, then raise an exception and exit from the loop.
 - **Else** modify all weights by adding a negative multiple of the gradient to the weights, calculate accuracy and loss for the given epoch.

End if

End loop

End Procedure

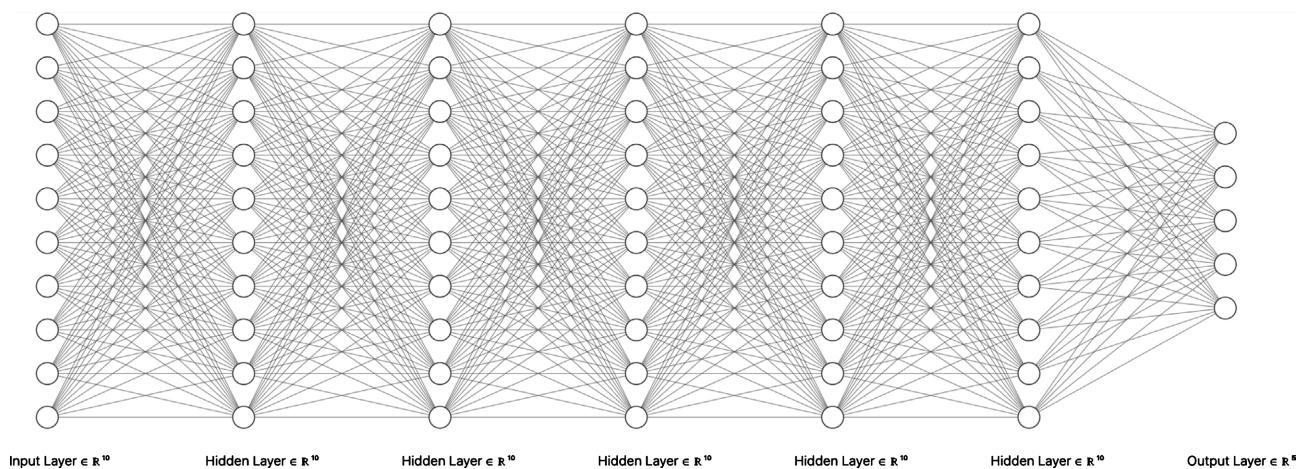


Figure 1. FFNN architecture for multi-label binary classification.

as a base for our task and eliminated from it, single reading metrics in order to predict possible failure date/time. After the cleaning process, we had a time-based historical dataset from the sensors: [“sensor7”, “sensor24”, “sensor25”, “sensor26”, “sensor64”, “sensor69”, “sensor105”].

The FFNN model is trained for a maximum of 50 epochs, and we stop the training if the validation loss does not improve after ten consecutive epochs. Given the imbalanced nature of the data, we utilize an imbalanced dataset sampler to re-balance the training class distributions. The model has been trained with the SGD [18] optimizer (initial learning rate = 10^{-2}) with a momentum rate 0 based on the validation loss. We use the default parameters for the SGD optimizer. The batch size is 16. The presented models were implemented in Keras [19] deep learning framework with TensorFlow backend in Python programming language. We use a validation dataset (20% source data) for hyper-parameter selection and early stopping. The hyper-parameters that we use for the FFNN include: the dense layer unit number is {32}, and the dropout rate {0.3}.

To obtain our results we used two workstations which have an i7 processor, 16 GB of RAM and 12 GB of NVidia Titan XP and Titan X GPUs. Both of the workstations use Ubuntu 20.04 operation systems with Cuda 10.1 and cuDNN 8.0.

3.2. Experiment Metrics

For evaluating the performance of our model, we used below metrics:

Accuracy score: is the metric which represents the fraction of correctly classified data instances over the total number of data instances:

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

Precision score: is a metric which has a positive predictive value and calculates fraction of relevant instances among the retrieved instances:

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

Recall: represents percentage of actual positives which are correctly identified:

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

F1 score: is a metric which is the harmonic mean of both: precision and recall

$$\text{F1 score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

Reliability probability: represents estimated prediction result based on the experiments.

Actual probability: represents exact probability from actual dataset.

3.3. Results of the Experiments

During our experiments with multiple epochs, we found that FFNN yielded the best results after 50 epochs. As the result of the tests, we determined that after 50 epochs accuracy [20] of the FFNN reached 0.970 (the good accuracy score for

the models is at least 0.90 out of 1 [2]) where the loss was 0.030 only. **Figure 2** and **Figure 3** shows how the accuracy and loss changed over 50 epochs on the train and validation dataset.

When the performance of the FFNN was evaluated, our approach was to compare the results of the model with similar architectural concepts. To accomplish this, we opted to utilize the “Decision Tree Classifier” (DTC) [21] and “Random Forest Classifier” (RFC) [22] algorithms. Both FFNN and DTC methods can model data that have nonlinear relationships between variables, and both can handle interactions between variables. On the other hand, RFC combines the predictions of many decision trees into a single model similarly.

As a first step, we chose to evaluate the accuracy and F1 scores of our model in comparison to the aforementioned models. **Table 1** displays the results of the compared accuracy and F1 scores for the FFNN, DTC, and RFC algorithms.

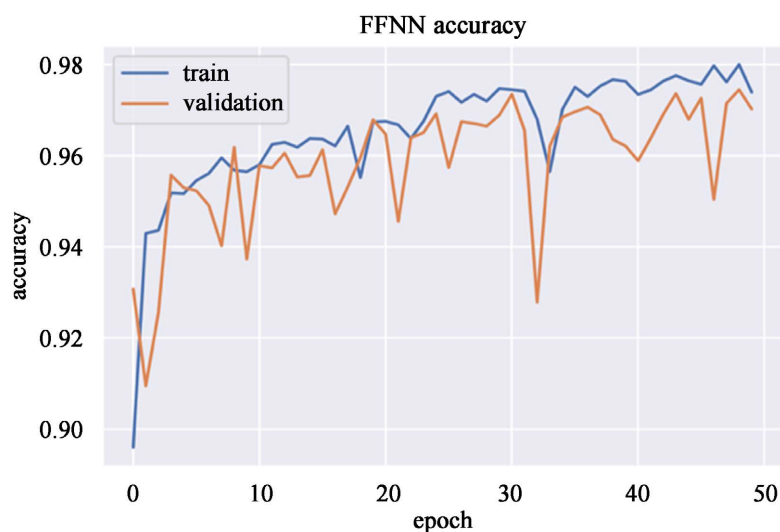


Figure 2. FFNN accuracy after 50 epochs.

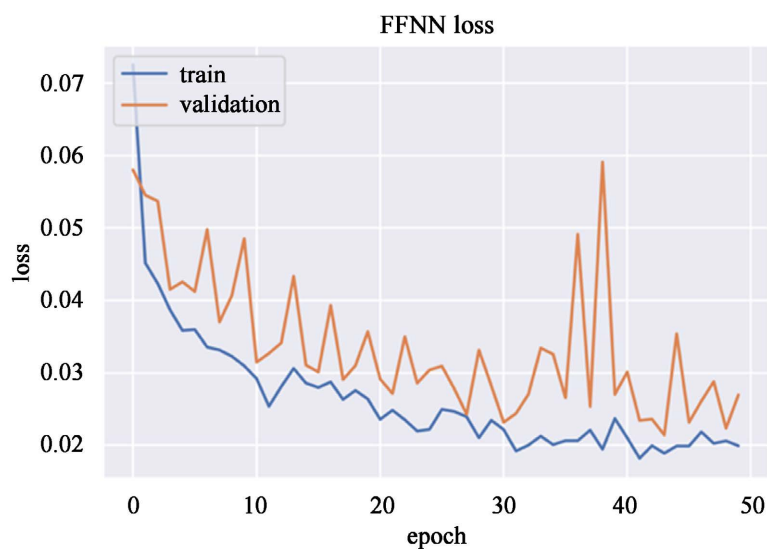


Figure 3. FFNN loss after 50 epochs.

For evaluating the effectiveness of FFNN, we also examined both precisions and recall our model for the provided data set. Evaluation results are summarized in **Table 2** included macro average and weight overage of the given model.

During the second stage of testing, we aimed to assess the actual and reliability probability of our model by comparing the performance to other earlier mentioned models. Our test results revealed that all models performed similarly in terms of fluctuations in reliability probabilities, and all three models performed well. **Figure 4** and **Figure 5** illustrate the changes in both actual and reliability probabilities, as well as the evolution of reliability probabilities across the nine stages for all proposed models.

Table 1. FFNN, RFC, DTC F1 and accuracy score.

Model Name	Metrics	
	ACCURACY SCORE	F1 SCORE
FFNN	0.970	0.520
RFC	0.949	0.474
DTC	0.932	0.612

Table 2. FFNN precision, recall^a.

Indicator	Predictions			
	0	1	Mavg	Wavg
Precision	1.00	0.36	0.68	0.99
Recall	0.97	0.92	0.94	0.97
Support	11788	212	12000	12000

^aMavg = Macro Average; Wavg = Weighted Average.

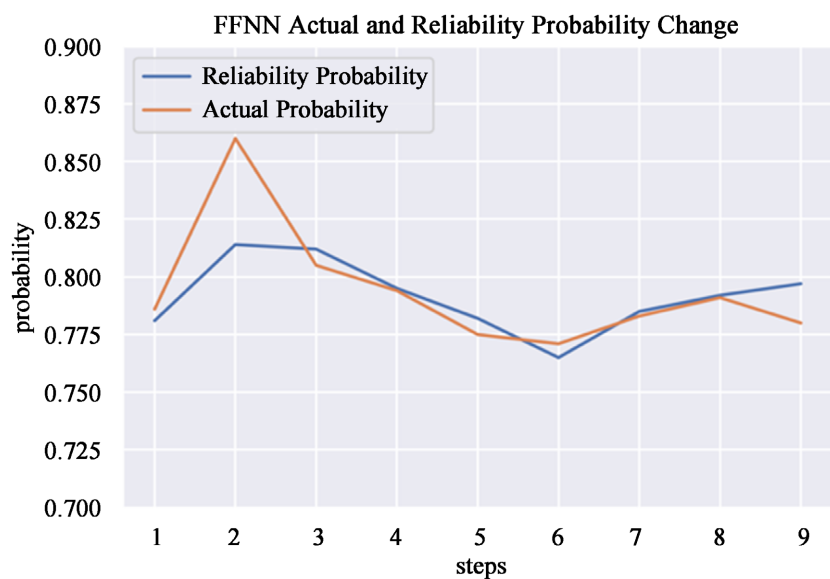


Figure 4. FFNN actual and reliability probability change after 9 steps.

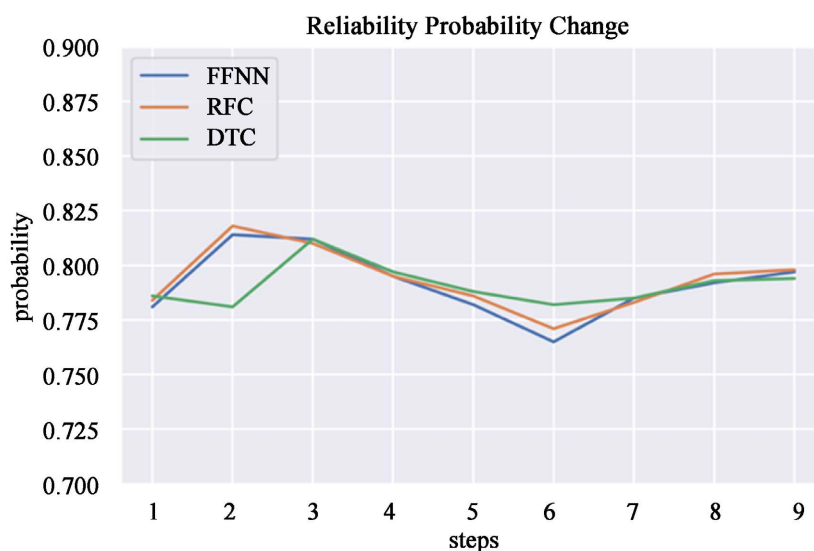


Figure 5. Reliability probability change after 9 steps.

The outcomes revealed that similar to other models FFNN had good performance on the provided time series data. This could be attributed to the dataset's dimensionality, as Neural Networks and Tree-based algorithms perform well on one-dimensional datasets. However, this may not hold for mixed multi-dimensional datasets. In the future, we plan to conduct additional research by applying our model to complex multi-dimensional datasets. The results and data presented in the tests indicate that FFNN can be used in various industries to detect potential malfunctions at an early stage.

4. Conclusions

In this paper, we introduced an FFNN-based prediction model for petroleum wells equipment failure. The proposed model can predict possible failure based on historical data. The results were compared with two different models which use “Random Forest” and “Decision Tree” classifiers. The evaluated accuracy and F1 scores, and the outcome of the proposed model are very competitive with previous state-art results. We also provided compared results of actual and reliability probability change over steps, which shows that for the given multi-label binary classification task, FFNN performed well. Our solution achieved these results without explicit alignments.

For the evaluation of the model's accuracy, we used the actual “ConocoPhillips” data set from the industry. To clean from this set of the data single reading metrics and created a new time-based series data set. We grouped this dataset into two subgroups (training and validation data sets) and validated the performance of all three models on a validation data set.

The results presented in this paper can be applied in real-time IoT systems; however, because of the rapid development of Machine Learning and IoT technologies, further aspects will need to be investigated.

In the next stage of our research, we are planning to extend our prediction

model for multiclass classification tasks, where we are going to predict when the possible equipment failure will occur on specific parts of the petroleum wells, e.g. based on the data set from the pipeline pressure when the next incident can happen. Considering the importance of the industry, predicting possible failures in the early stages, can be a vital solution for avoiding incidents.

We will also evaluate the performance of the different prediction methods as part of our future research. Moreover, we are going to investigate the performance of the different models on different real-world tasks.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

- [1] Brissaud, F., Varela, H., Declerck, B. and Bouvier, N. (2012) Production Availability Analysis for Oil and Gas Facilities: Concepts and Procedure. *11th International Probabilistic Safety Assessment and Management Conference and the Annual European Safety and Reliability Conference*, Helsinki, June 2012, 4760-4769.
- [2] Tribedi, U. (2020) What Is the Maximum Accuracy That a Machine Learning Model Can Achieve?
<https://medium.com/think-ai/what-is-the-maximum-accuracy-that-a-machine-learning-model-can-achieve-e43dba772080>
- [3] Deyab, S.M., Taleb-berrouane, M., Khan, F. and Yang, M. (2018) Failure Analysis of the Offshore Process Component Considering Causation Dependence. *Process Safety and Environmental Protection*, **113**, 220-232.
<https://doi.org/10.1016/j.psep.2017.10.010>
- [4] U.S. Fire Administration (1989) Technical Report Series Phillips Petroleum Chemical Plant Explosion and Fire Pasadena, Texas. USFA-TR-035.
<https://ncsp.tamu.edu/reports/USFA/pasadena.pdf>
- [5] Jackson, R.B. (2014) The Integrity of Oil and Gas Wells. *Proceedings of the National Academy of Sciences of the United States of America*, **111**, 10902-10903.
<https://www.pnas.org/content/111/30/10902>
<https://doi.org/10.1073/pnas.1410786111>
- [6] Burt, J. (2018) Unifying Oil and Gas Data at Scale. The Next Platform.
<https://www.nextplatform.com/2017/05/30/unifying-oil-gas-data-scale/>
- [7] Al-Shehri, D.A. (2019) Oil and Gas Wells: Enhanced Wellbore Casing Integrity Management through Corrosion Rate Prediction Using an Augmented Intelligent Approach. *Sustainability*, **11**, Article 818. <https://doi.org/10.3390/su11030818>
- [8] Martí, L., Sanchez-Pi, N., Molina, J.M. and Bicharra Garcia, A.C. (2014) YASA: Yet Another Time Series Segmentation Algorithm for Anomaly Detection in Big Data Problems. In: Polycarpou, M., *et al.*, Eds., *Hybrid Artificial Intelligence Systems. HAIS 2014. Lecture Notes in Computer Science*, Springer, Cham, 697-708.
https://doi.org/10.1007/978-3-319-07617-1_61
- [9] Syah, R., Ahmadian, N., Elveny, M., Alizadeh, S.M., Hosseini, M. and Khan, A. (2021) Implementation of Artificial Intelligence and Support Vector Machine Learning to Estimate the Drilling Fluid Density in High-Pressure High-Temperature Wells. *Energy Reports*, **7**, 4106-4113. <https://doi.org/10.1016/j.egyrs.2021.06.092>

- [10] Kumar, A. and Kumari, M. (2020) Design and Analysis of IOT Based Real Time System for Door Locking/Unlocking Using Face Identification. *International Journal of Recent Technology and Engineering*, **8**, 2093-2095. <https://doi.org/10.35940/ijrte.E5794.018520>
- [11] Shi, F., Yan, L., Zhao, X. and Gao, R.X.-K. (2022) Machine Learning-Based Time-Series Data Analysis in Edge-Cloud-Assisted Oil Industrial IoT System. *Mobile Information Systems*, **2022**, Article ID: 5988164. <https://doi.org/10.1155/2022/5988164>
- [12] Anderson, R.N. (2017) 'Petroleum Analytics Learning Machine' for Optimizing the Internet of Things of Today's Digital Oil Field-to-Refinery Petroleum System. 2017 *IEEE International Conference on Big Data (Big Data)*, Boston, MA, 11-14 December 2017, 4542-4545. <https://doi.org/10.1109/BigData.2017.8258496>
- [13] Al-Radhi, M.S., Al-Kamil, S.J. and Tamás, S. (2020) A Model-Based Machine Learning to Develop a PLC Control System for Rumaila Degassing Stations. *Journal of Petroleum Research and Studies*, **10**, 1-18. <https://doi.org/10.52716/jprs.v10i4.364>
- [14] Rahmani, A.M., Ali, S., Malik, M.H., Yousefpoor, E., *et al.* (2022) An Energy-Aware and Q-Learning-Based Area Coverage for Oil Pipeline Monitoring Systems Using Sensors and Internet of Things. *Scientific Reports*, **12**, Article No. 9638. <https://doi.org/10.1038/s41598-022-12181-w>
- [15] Seide, F., Fu, H., Droppo, J., Li, G. and Yu, D. (2014) 1-Bit Stochastic Gradient Descent and Its Application to Data-Parallel Distributed Training of Speech DNNs. *INTERSPEECH 2014*, Singapore, 14-18 September 2014, 1058-1062. <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/IS140694.pdf>
- [16] Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) Learning Representations by Back-Propagating Errors. *Nature*, **323**, 533-536. <https://doi.org/10.1038/323533a0>
- [17] Predictive Equipment Failures. Kaggle. <https://www.kaggle.com/c/equipfails/data>
- [18] Halgamuge, M.N., Daminda, E. and Nirmalathas, A. (2020) Best Optimizer Selection for Predicting Bushfire Occurrences Using Deep Learning. *Natural Hazards*, **103**, 845-860. <https://link.springer.com/article/10.1007/s11069-020-04015-7>
<https://doi.org/10.1007/s11069-020-04015-7>
- [19] Keras (n.d.) Simple. Flexible. Powerful. <https://keras.io/>
- [20] Brownlee, J. (2022) How to Calculate Precision, Recall, F1, and More for Deep Learning Models. <https://machinelearningmastery.com/how-to-calculate-precision-recall-f1-and-more-for-deep-learning-models/>
- [21] Mohitdholi (2019) Predictive Equipment Failures. Kaggle. <https://www.kaggle.com/mohitdholi/conoco-kernal-md>
- [22] Parsadastjerdi (2019) Conoco Phillips Challenge. Kaggle. <https://www.kaggle.com/parsadastjerdi/conoco-phillips-challenge>