Journal of Physics: Conference Series 2259 (2022) 012028 doi:10.1088/1742-6596/2259/1/012028

Data science application in structural integrity analysis of fixed offshore jacket platform

X C Yak[1], E Mat Soom[2*], L K Quen[1], M K Abu Husain[3], M A Azahar[4], N I Mohd Zaki[3], H S Kang[5]

[1] Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, 54100 Kuala Lumpur, Malaysia
[2] Repsol Malaysia Limited, Jalan Ampang, Kuala Lumpur, 50450 Kuala Lumpur, Malaysia
[3] Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, 54100 Kuala Lumpur, Malaysia
[4] Universiti Kebangsaan Malaysia, 43600 UKM, Bangi, Selangor, Malaysia.
[5] School of Mechanical Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor Bahru, Johor, Malaysia.

E-mail: ezanizam@outlook.com

Abstract. Accuracy in analysing the integrity of a structure is critical for determining the structure's fitness for service and reliability status. Today, a variety of techniques and approaches are applied, including the use of data science applications. Data science is a synthesis of computer science, mathematics, and statistics. Meanwhile, the integrity of a structure is susceptible to a mix of statistical and technical design uncertainties that may remain flexible as long as the structure is capable of successfully managing the encountered load. Numerous applications are used in the oil and gas sector to estimate the probability of failure (POF), but they all have a particular restriction. Integral interference equations based on load versus strength are reliable for determining the POF of fixed offshore structures. This study is a quantitative risk assessment, emphasising the Python application, an improved and reliable method for calculating the POF value. A representative sample of the monopod offshore structure was chosen and subjected to global non-linear analysis in this study. The most reliable form of distribution was predetermined, and the algorithm created using Python was used to apply and compute the suitable integral equation depending on the load and strength conditions. The Python method's result demonstrated a high degree of confidence in calculating the new POF in intact condition from a design perspective, inspection interval, and risk to consider.

1. Introduction

Industries have been under continuous pressure to improve the integrity and reliability of their facilities and assets as a result of rising operating expenses and a rise in demand for oil and gas. In structural reliability assessment (SRA), probabilistic models are used to evaluate the reliability of a structure. The purpose of using probabilistic models is to assess the uncertainty associated with statistics and engineering design data [1]. There are many differences between a structure that is being designed and an existing one that is being assessed. The first difference is in the information accessible about the structure, which comprises historical performance data and the structure's current condition. This information may be used to evaluate the structure's continued safety [2]

According to [3], asset integrity is defined as an asset's capacity to fulfil its intended purpose effectively and efficiently while protecting people and the environment. While asset integrity management (AIM) is the process of ensuring that the people, systems, processes, and resources that contribute to the integrity are in existence, in use, and fit for purpose throughout an asset's life. The probability of a device fulfilling its function over a given time period and under specified operating circumstances is called reliability. Time may be quantified in a variety of ways. For instance, operational duration, mileage, shelf life, and cycle number.

To create an algorithm written with Python which will perform the SRA, a few short objectives have been set as the milestone of developing the algorithm. The first objective is to calculate the POF using Python based on the simulation data from pushover analysis of the fixed offshore platform. Once the POF has been calculated, the second objective is to predict the overall POF of the said platform for 12 directions. The last objective is to compare the predicted POF with the international standards.

1.1. Stress-Strain

It is common to pre-determine the loads to be supported and then design the structural members to have enough tolerance to resist the pre-set loads while also preventing the structural parts from failing if the predetermined loads exceed those specified. The safety margin is determined by the level of safety that is needed. The stress-strain diagram for steel is shown in Figure 1 (not to scale). As a result, in order to prevent significant and irreversible strain or deformation as a result of stress, the yield stress is often selected as the failure stress in practice. Engineers or manufacturers often specify maximum stress or load lower than the yield stress to account for the uncertainties in the load distribution. When the material is loaded to its maximum capacity, there is an additional safety margin to ensure that the material does not deform in an unpredictable manner [3].

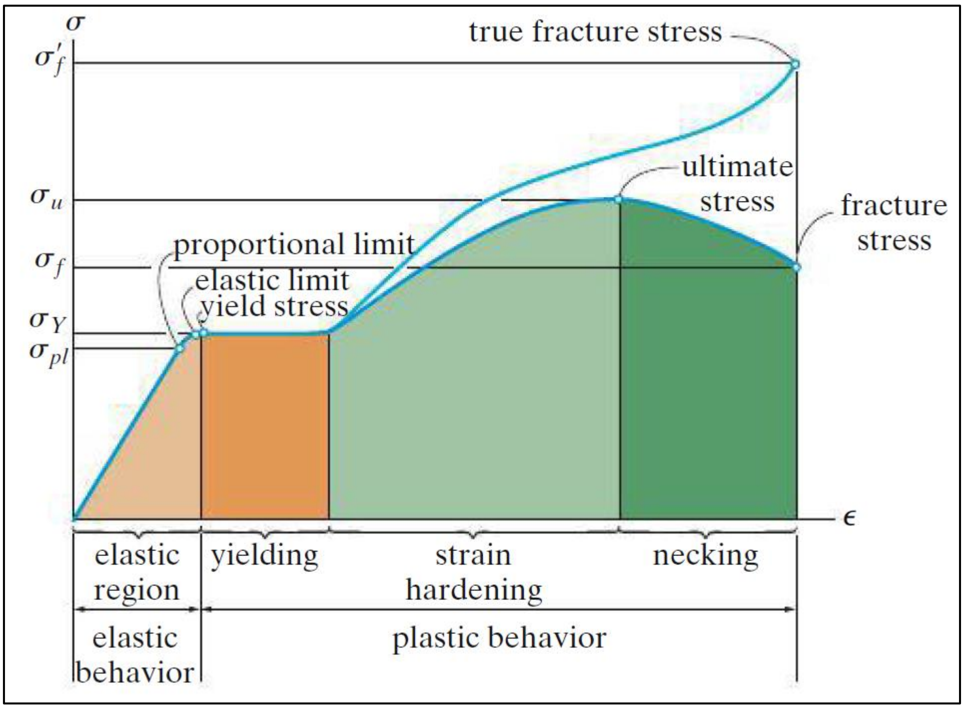

Figure 1. Conventional and true stress-strain diagram for ductile material (steel) [4].

1.2. Probability of Failure (POF)

Probability is a tool to quantify the uncertainty of events and reason in a principled manner. It is essential to know that this integral equation is applicable only if the load and strength models are statically independent for a continuous variable [1]. The process of finding an integral is called integration. In basic reliability engineering, the meaning of reliability is the performance at or above a given standard (i.e. rated from 0 to 1, where 0 is least reliable). On the other hand, the probability of failure is the performance below a given standard (i.e., rated from 0 to 1, where 1 is an absolute failure). Since these events (success and failure) are complements, the following is true where reliability is equivalent to; 1 – the probability of failure (POF). Failures occur where loads and strength models overlap each under [5-8].

Probability of failure (POF) can be expressed as:

$$POF = P(S \leq L) = 1 - P(L \leq S) = 1 - R \tag{1}$$

$$POF = 1 - \int_{-\infty}^{\infty} f_S(s) \left[\int_{-\infty}^{s} f_L(l) dl \right] ds \tag{2}$$

$$POF = \int_{-\infty}^{\infty} [1 - F_L\,(s)\,]\, f_S(s) ds \tag{3}$$

where $f_S(s)$ and $f_L(l)$ are the probability density function (*PDF*) of the strength s and load l, respectively. While $F_L\,(s)$ is probability distribution function (*PDF**) of load in a unit of strength.

Alternately, the probability of failure (POF) can also be expressed as:

$$POF = 1 - R = 1 - (S \geq L) \tag{4}$$

$$POF = 1 - \int_{-\infty}^{\infty} f_L\,(l) \left[\int_{l}^{\infty} f_S(s) d_s \right] dl \tag{5}$$

$$POF = \int_{-\infty}^{\infty} f_L\,(l)\, F_S\,\,(l) dl \tag{6}$$

where $F_S\,(l)$ is the probability distribution function (*PDF**) of strength in the unit of load.

In accordance with the present knowledge of nonlinear analysis study, the Reserve Strength Ratio (RSR) values considered in estimating the potential collapse of offshore fixed structures range between 0.80 and 1.39.

1.3. Data Science and Algorithm

When computer science, mathematics, and statistics are integrated, data science is a multidisciplinary subject. It applies data collection to analyse and identify the pattern of data, which is then used to explain a phenomenon that the researcher is studying. In order to analyse data, machine learning may be developed especially for this purpose [9]. Data Science and Data Analytics have closely linked principles. Analysing data is a science whose objective is to derive a conclusion based on the information included within the data itself [10].

Algorithms, in layman terms, are necessary steps or rules to be followed to accomplish specific tasks. The development of algorithms may be accomplished via the use of a variety of computer languages and tools. In a specific study published in 2014, C++, Java, and Python were all compared. According to the findings of this study, Python is by design a more readable and simpler programming language. There are certain parallels between these three languages in terms of enabling abstraction, which demonstrates their capacity to conduct complex problem-solving. Python's popularity is due in part to its readability and simplicity of code, but it is also due to the variety of Python implementations available. Its flexibility, which includes the ability to utilise Web frameworks (such as Django) and improved integrated development environments (such as PyCharm), provides an infinite number of possibilities for what a programmer may do with it [11].

2. Methodology

2.1. Case Study

The input of metocean (meteorology and oceanography) data for the test structure, including wave height, wave period, winds, currents, tides and water levels, was used and analysed in order to run the test structure's entire modelling and determine the new probability of failure (POF). Then, the appropriate integral equation was used, based on the kind of distribution, probability density function (PDF) or probability distribution function (PDF*) and the condition, either greater load or strength, to multiply both PDF and PDF* according to the selected integral equation (3) or equation (6).

A monopod braced support platform in offshore with a water depth of 54 metres below mean sea level (MSL) was selected as a test structure based on the availability of the data provided by the industry. Considering the fact that it is a monopod or one-legged, the nonlinear analysis took into consideration 12 different directions of environmental impact. The data in Table 1 below includes details on 12 impact directions, base shear at load factor impact and collapse, reserve strength ratio (RSR), and ranking based on the RSR value. It is the lowest-ranked with RSR 1.496, and the direction of 124deg is the most critical one in the ranking.

Table 1. Detail information of monopod braced support platform.

Pushover Direction (°)	Base Shear at Load Factor 1.0 (kN)	Base Shear at Collapse (kN)	RSR	Ranking of Critical Pushover Direction
4	1154.00	5382.00	4.664	10
34	1120.00	6887.00	6.149	12
64	2683.00	5743.00	2.141	5
94	3245.00	5004.00	1.542	2
124	3300.00	4936.00	1.496	1
154	2894.00	4772.00	1.649	3
184	2658.00	5440.00	2.047	4
214	2606.00	6943.00	2.664	7
244	2505.00	5666.00	2.262	6
274	1093.00	4786.00	4.379	9
304	1122.00	4781.00	4.261	8
334	818.40	4604.00	5.626	11

2.2. Method - Python

This study aims to determine the probability of failure (POF) for a case study, which will need the use of Python to write the algorithm. The method begins by extracting data from an input data file (spreadsheet.xlsx) from a pushover nonlinear analysis outcome. The file was tabulated into four (4) columns:

i. Reserve strength ratio (RSR) for a hundred years return period (RSR100)

ii. Base shear (BS = Q)
iii. Probability of base shear (P(BS = Q)
iv. Probability of ultimate strength (P(Ultimate Strength = R)

Environmental stresses, namely waves and sea currents, were simulated from twelve various directions. Environmental loads were calculated using metocean data for a 100-year return period (RP) in Malaysian Water.

Figure 2 (i) shows the write of lines that imports python libraries and plotting. The libraries must be imported in order to use functions such as *pandas.read_excel()* and *numpy.where()*. Figure 2 (ii) shows on how to read an input file with pandas, the file's path must be provided. In this case, the whole path to the file is given as a parameter in *pandas.ExcelFile()* (since pandas may be invoked through pd, hence *pd.ExcelFile())*.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as ss

plt.plot
```

(i)

```
file = pd.ExcelFile(r"C:\\Users\Mriya\Desktop\FYP\SRA_Algorithm\InputData.xlsx")
```

(ii)

Figure 2. (i) Python libraries and plotting and (ii) code to read from input file.

The column names must be provided in Figure 3 (i) on how to read particular columns from a file, as shown below. This creates a list called ‘col’ with the columns’ names (square brackets indicate that the variable is a list).

- Select column(s) to read (RSR100, BS = Q, P(BS = Q), P(Ultimate = R))
  (Note: the column name needs to be the same as in Excel file)

In addition, for Figure 3 (ii), due to the fact that the input file contains 12 separate worksheets, and to facilitate the naming of the output file, the names for each of the 12 directions are placed in a list called ‘direction’.

```
col = ['RSR100', 'BS = Q', 'P(BS =Q)', 'P(Ultimate = R)']
```

(i)

```
direction = ['4 deg', '34 deg', '64 deg', '94 deg', '124 deg', '154 deg',
'184 deg', '214 deg', '244 deg', '274 deg', '304 deg', '334 deg']
```

(ii)

Figure 3. (i) columns name and (ii) 12 directions wave impact.

In Figure 4 (i), a ‘*for*’ loop is used to go through all 12 directions. The *len()* function returns the list's length, or the number of entries, while the *range()* function defines the integer range for the loop, which in this event is zero (0) to eleven (11). Continue from Figure 4 (ii), any repeated actions using the ‘*for*’ loop will have a two-space indent before the line. This indicates to Python that the line is included in the loop. The first action in the loop is to read the file and save the contents to the variable called 'data' using *pandas.read_excel().*

```
for i in range(len(direction)):
```

(i)

```
data = pd.read_excel(file, sheet_name=direction[i],usecols=col, engine='openpyxl')
```

(ii)

Figure 4. (i) loop operation and (ii) read excel and store content in data.

The ultimate strength, R, could be determined by multiplying the value in column 'RSR100' with the value in column 'BS = Q' included within the data. The result is then stored in a new column inside data called 'Ultimate = R', which was formed using the *insert()* function in between columns 'BS = Q' and 'P(BS = Q)'. This can be accomplished in the manner shown in Figure 5 (i)

- Calculate 'Ultimate = R' by multiplying 'RSR100' with 'BS = Q'
- Insert the result into the data frame 'data' after column 'BS = Q'

From Figure 5 (ii), Python uses *numpy.where()* to pick a particular equation based on a condition (because numpy is imported as np, thus *np.where()).* This function is comparable to Microsoft Excel's IF function. The condition is configured to check if the value in 'P(BS = Q)' is higher than the value in 'P(Ultimate = R)' The second line is executed if the condition evaluates to TRUE. If the condition evaluates to FALSE, the third line is executed

- Calculate POF for each row using one of the equations
- Equation to be used is based on which probability is larger than the other

```
R = data['RSR100'] * data['BS = Q']
data.insert(2, 'Ultimate = R', R)
```

(i)

```
data['Q X R'] = np.where(data['P(BS = Q)'].gt(data['P(Ultimate = R)']),
(1 -data['P(BS = Q)']) * data['P(Ultimate = R)'],
data['P(BS = Q)'] * data['P(Ultimate = R)'])
```

(ii)

Figure 5. (i) multiplying and (ii) conditions evaluation.

To filter away rows of data that include unusable data (rows where 'BS = Q' equals 0) and are beyond the range of $0.8 \leq$ RSR100 < 1.4, the *np.where()* function is used once again. This time, the value in 'Q X R' is set to zero if the same row's value of 'BS = Q' is equal to zero. The value in 'Q X R' at the same row called 'RSR100' is maintained as computed within the specified range, while the remainder of the value is set to 0. This procedure may be performed in the manner shown Figure 6 (i).

In addition, to save the value of POF and RP computed in the following operation for later use, the first loop initialises the two variables called POF and RP as empty lists as Figure 6 (ii).

```
# Filter out rows with BS = Q = 0
data['Q X R'] = np.where(data['BS = Q'] == 0, 0, data['Q X R'])

# Filter out the value which is outside of the range of 0.8 <= x <= 1.4
data['Q X R'] = np.where(data['RSR100'] < 0.8, 0, data['Q X R'])
data['Q X R'] = np.where(data['RSR100'] > 1.4, 0, data['Q X R'])
```

(i)

```
#Initialise an empty list for Probability of Failure (POF) and #ReturnPeriod (R
P)

if i == 0:
   POF = []
   RP = []
```

(ii)

Figure 6. (i) filter process and (ii) loop initialises.

In Figure 7 (i), the POF for the range 0.8 ≤ RSR100 <1.4 is computed by totalling the whole column of 'Q X R' using the *sum()* function. After that, the computed value is added to the list using the *append()* function. Thus, the whole procedure may be carried out in the following manner. Note that the last line is used to show the result in the integrated terminal in Visual Studio Code, which has been commented out by appending a "#" before the line.

The same process is used to compute RP, which may be accomplished in the following manner as shown in Figure 7 (ii). The last two lines, which have been commented out, are used to output the result of RP to the terminal.

```
# Calculate and store result of Probability of Failure (POF) into Python
# list named 'POF'

POF.append(data['Q X R'].sum())

# print("Probability Of Failure for", direction[i], "=", POF[i])
```

(i)

```
# Calculate and store result of Return Period (RP) into
Python list
# named 'RP'

RP.append(round(1 / POF[i], 2))

# print("Return Period for", direction[i], " =", RP[i],
# "year(s)")
```

(ii)

Figure 7. (i) Calculation on the POF and (ii) Calculation on the return period (RP).

The process of the whole algorithm can be represented with the flowchart in Figure 8.

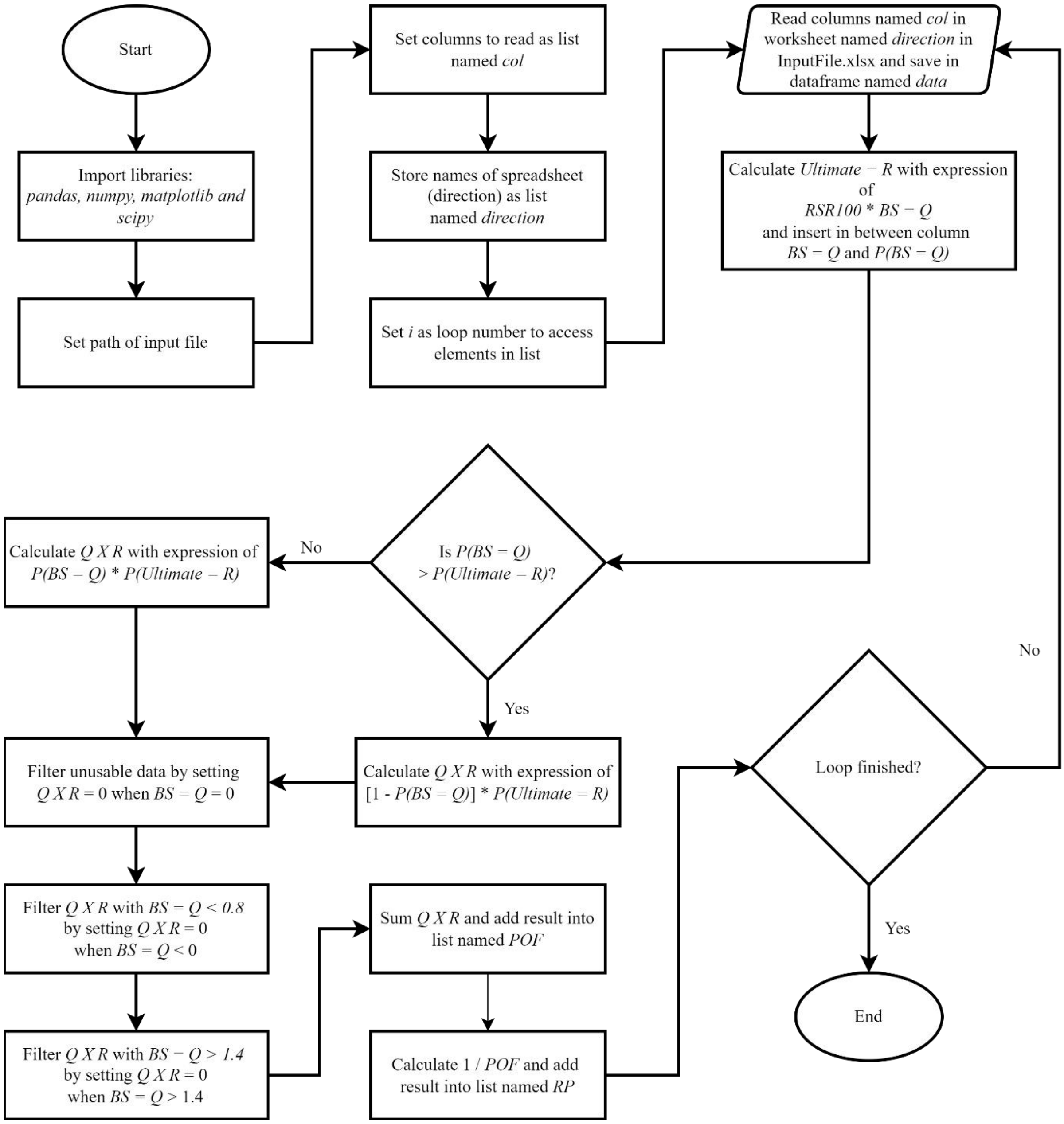

Figure 8. Flowchart of the algorithm.

3. Result and Discussion

According to [12] and [13], this platform is classified as C2 (medium consequence category) and S3 (unmanned, life safety category). As a result, the exposure rating for this monopod braced jacket platform would be L2 (low exposure) by referring to the ISO standard. Table 2 shows the probability of failure specified in the three standards as listed. Based on table 2, the higher the POF, the higher the risk. According to ISO 19902:2007 (Design Target POF) and ISO 19901-9:2019 (Inspection Interval POF), POF of 5.00×10^{-4} is considered as critical condition. Meanwhile, PETRONAS's recommendation is more stringent where at POF of 4.00×10^{-4}, risk mitigation is required, which the POF is much lower compared to another two standards.

3.1. Code and Standard Recommendation

Following the development of the method, conclusions regarding the result of POF obtained using the Python algorithm may be derived by comparing it to the result calculated using Equation (3) and (6) against the [12] target reliability, [13] inspection interval and [14] as authority point of view. It is essential to assess if the POF results are in good condition from a design perspective, inspection interval, and risk to consider.

Table 2. Probability of failure comparison measure (POF)

<table>
<tr><th>Risk Ranking</th><th>Probability of Failure (POF)</th><th colspan="2">ISO 19902:2007 Amendment - 19902:2013 Design Target POF</th><th colspan="2">ISO 19901-9:2019 Inspection Interval POF</th><th colspan="2">PETRONAS Recommendation (Ayob et al, 2014)</th></tr>
<tr><td rowspan="9">Very High

Very low</td><td>1.00</td><td></td><td rowspan="4">L1/L2: Critical Condition</td><td></td><td rowspan="4">L1: 1-3 years
L2: 3-5 years</td><td></td><td rowspan="5">For Risk Mitigation</td></tr>
<tr><td>1.00×10^{-2}</td><td></td><td></td><td></td></tr>
<tr><td>1.00×10^{-3}</td><td></td><td></td><td>Unmanned Design</td></tr>
<tr><td>5.00×10^{-4}</td><td>L2 @ 1/2000</td><td></td><td rowspan="2"></td></tr>
<tr><td>4.00×10^{-4}</td><td></td><td rowspan="2">L1/L2: Good Condition</td><td>Medium</td><td>L1/L2: 3-5 years</td></tr>
<tr><td>1.00×10^{-4}</td><td></td><td>Low</td><td rowspan="4">L1: 3-5 years
L2: 6-10 years</td><td>Manned Design and Inspection</td><td rowspan="4">For Monitoring</td></tr>
<tr><td>3.00×10^{-5}</td><td>L1 @ 1/33000</td><td rowspan="3">L1/L2: Best Condition</td><td></td><td></td></tr>
<tr><td>1.81×10^{-6}</td><td></td><td></td><td></td></tr>
<tr><td>0.00</td><td></td><td></td><td></td></tr>
</table>

3.2. Comparison Result to Standard

The results of the POF calculation for 12 directions of wave impact of overlap between probability density function (PDF) and probability distribution function (PDF*) using the Python approach are shown in Figure 9, which is based on the sum of load and strength distributions using the applied integral equation.

It is compared to [12-13] and [14] recommendation as to the Malaysian authority. The reference line by ISO 19902 in Figure 9 is the outcome from a design perspective is L2, which is under the best condition. The reference by ISO 19901, on the other hand, indicates the POF of the inspection interval is between six (6) and ten (10) years. Meanwhile, the reference line by PETRONAS utilized in Figure 9 indicates the requirement of monitoring only, without the need of risk mitigation. Based on the calculated POF in the present study, the estimated POFs remain within the allowed values for the standard mentioned above, which is a platform braced monopod.

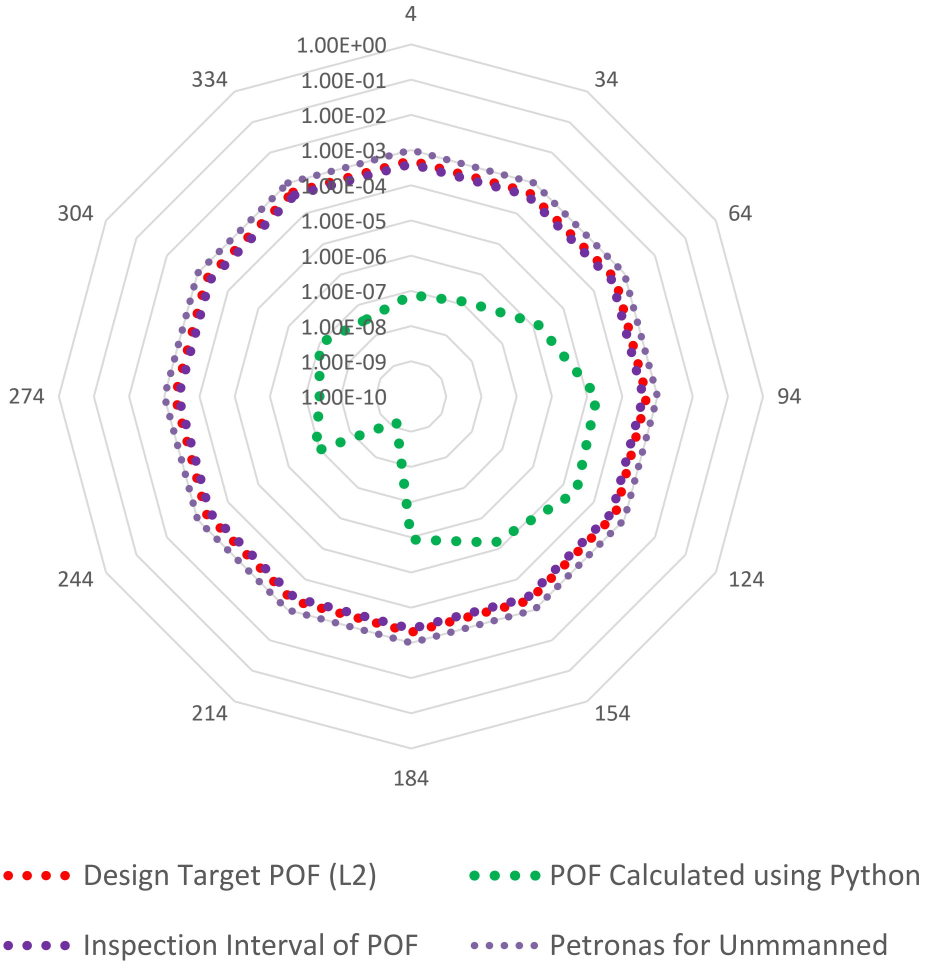

Figure 9. Comparison result for Probability of failure (POF)

4. Conclusion

Safety is determined statistically, while structural reliability is determined via engineering methodology. The principle of demand and supply, or load and strength, was used in this simplified reliable analytical method to estimate the probability of failure of test structures exposed to severe storm conditions.

Algorithms are often used to refer to the rules or steps that must be followed to achieve specific tasks, when the rules or steps are established by humans. The rules or procedures may be as straightforward as those we follow in our everyday lives. Simple addition and subtraction, multiplication, and division may all be considered algorithms, as can more complicated mathematical operations such as long division, integration, and differentiation, which need the application of processes or rules to get the final results of the computations.

The probabilistic model utilized in the present study is a combination of the load and strength models. All results from analyses of the wave impact of overlap between the probability density function (PDF) and the probability distribution function (PDF*) are acceptable and conform to the platform operator's criteria for value delivery and benefit categorization. It is economically advantageous in terms of resource optimization and platform re-evaluation.

This is critical to the oil and gas industry's standard since it relates to quality, safety, and cost, particularly when the collapse of the platform may result in fatalities, asset damage, and environmental issue.

Acknowledgement
The authors would like to acknowledge the support of the academic and industrial establishments they present. The leading author has undertaken this study for the final year project (FYP) subject in his bachelor’s degree with supported by industry co-authors. This research was financially supported by Universiti Teknologi Malaysia under Industry-International Incentive grant (Q.K130000.3643.03M14).

References
[1] C. Sundararajan and F. J. Witt, Stress-Strength Interference Method, Probabilistic Structural Mechanics Handbook, 1st Edition, Springer, Boston, MA, 1995.
[2] E. Gerhard et al, Ageing and Life Extension of Offshore Structures, John Wiley & Sons Ltd: UK, University of Stavanger, Oxfordshire, 2019, ppl 134.
[3] R. M. Chandima, Best Practices for Topside Facilities Inspection, Capgrow, Singapore, 2019, ppl 5
[4] R. C. Hibbeler, Mechanics of Materials, 9th Edition, Pearson Prentice Hall, 2014
[5] C. S. Naresh, Using @Risk in Load and Resistance factor Design (LRFD), Approach for Civil Engineering, Consultant Palisade Websat, USA: NCS, 2012
[6] C. Srinivasan, Offshore Structural Engineering, Reliability and Risk Assessment, Taylor and Francis Group, 2016
[7] A. Younan, Basic Principle of Structural Reliability, Exxon Mobil Upstream Research Company (2014)
[8] S. S. Rao, Reliability Engineering, 1st Edition., Pearson, 2016.
[9] J. Grus, Data Science from Scratch, 2nd Edition, O'Reilly Media, Inc., 2019
[10] A. Monnapa, Data Science vs. Big Data vs. Data Analytics, Simplilearn Americas, Inc. 2020
[11] S. J. Humer and E. C. Foster, A Comparitive Analysis Of The C++, Java, And Python Languages, 2014
[12] ISO 19902, Petroleum and Natural Gas Industries - Fixed Offshore Structures, First Edition 2007-12-01, International Standards Organization, Switzerland, 2007
[13] ISO 19901, Petroleum and Natural Gas Industries – Specific Requirements for Offshore Structure, Part 9: Structural Integrity Management, First Edition 2019-07, International Standards Organization, Switzerland, 2019
[14] Ayob et al, Requalification of Offshore Jacket Structures in Malaysia Waters, OTC-25021-MS, PETRONAS, 2014