Failure prediction with Weibull distribution

Anita Agárdi[1], and Károly Nehéz[1]

Abstract—**In the industrial environment, the reliability of machines and equipment is not only a matter of convenience but a key factor in terms of the productivity and competitiveness of companies. Unexpected breakdowns, shutdowns or malfunctions of machines can cause serious economic damage, not to mention potential workplace accidents or environmental damage. This article presents a failure prediction model, where the probability failure of machines and pieces of equipment is determined using the Weibull distribution. The model can predict the failure of a single machine and determine the failure of the entire system. After the introduction, the most important literature on the subject is presented, followed by a description of the Weibull distribution. The article describes the test datasets and their results. The tests were created for the following data sizes: 2 units, 5 units, 15 units, 40 units, 100 units.**

Index Terms—**failure prediction, machine health, Weibull distribution**

I. Introduction

The reliability and efficiency of machines and pieces of equipment are key factors for the smooth operation of production. Machine breakdowns and unexpected shutdowns can cause significant economic losses and also increase the risk of workplace accidents and environmental pollution. To increase the efficiency of industrial maintenance and operation, failure prediction has become increasingly important. Over the years, researchers have proposed mathematical models and distributions to determine the probability of machine failure, which can be used to optimize maintenance schedules and minimize unexpected downtime. The Weibull distribution is a powerful tool in failure prediction and reliability analyses. The Weibull distribution is a parametric probability distribution. It is well adapted to data that comes from long-lived systems and helps in the analysis of systems where failure changes over time. The rest of the article is structured as follows: Section 2 describes the importance and the scientific background of the topic based on search engine analyses and related publications. Section 3 describes the Weibull distribution. After that, Section 4 presents the test datasets and their results. The last section presents the conclusion and future research direction.

The main contribution of this paper is the application of the Weibull distribution to predict system failures in several datasets of different sizes (2, 5, 15, 40, and 100 units). The research presents two innovative types of diagrams for visual representation of failure probabilities: failure unit and fault tree probability diagrams. These tools enable a more detailed and efficient understanding of the failure behavior of industrial equipment, providing insights to optimize maintenance schedules and reduce unplanned downtime.

II. Literature review

Over the years, many researchers have investigated the topic of failure prediction. The article presents the search results of Google Scholar, where the following keywords were investigated: "Asset lifecycle management", "Condition monitoring" and, "Maintenance strategies". , "Machine health monitoring", "Reliability-centered maintenance", "Equipment reliability", "Lifecycle cost analysis". Figure 1 presents the number of publications per year from 2010 to 2023. Based on the figure it can be seen that the number of publications is increasing over the year.

The article then presents some of the featured articles that investigate the topic of failure probability determination system.

Çınar, Z.M. et al. al [1] investigated predictive maintenance. The paper describes the following types of maintenance: reactive (repairing if it is damaged), planned (scheduled maintenance), proactive (troubleshooting to improve performance), and predictive (reliability is predicted). The article reports on a system where smart sensors monitor machines. This data is transmitted over the network, then the data is monitored by staff to investigate if the machines are well maintained and whether they are in good condition. The software can predict future failures and machine health. The system infers this condition from past data with Machine Learning. The system automatically issues a maintenance ticket to the technician, which the technician approves and performs.

Karuppusamy, D. P. [2] investigates the predictive maintenance scheduling. This maintenance process is based on sensors and their measurements. In the article, the following algorithm was used for predictive maintenance scheduling: decision tree, and random forest.

Luo, M. et al. [3] presented a two-stage maintenance system for equipment prediction and maintenance schedule optimization. A neural network is used for the failure prediction.

Wan, J. et al. [4] implemented a cloud-based big data solution for active preventive maintenance in a production environment. It provides data processing, analysis and forecasting.

[1] Institute of Informatics, Faculty of Mechanical Engineering and Informatics, University of Miskolc, Hungary (E-mail: anita.agardi@uni-miskolc.hu, karoly.nehez@uni-miskolc.hu)

DOI: 10.36244/ICJ.2025.5.5

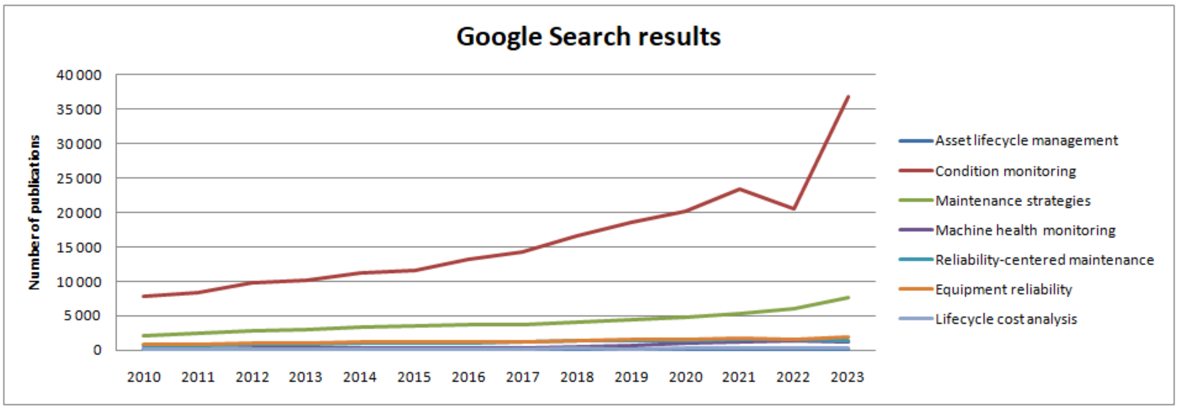

Fig. 1. Google Search result for “Condition monitoring” keyword

TABLE I
COMPARISON OF MAINTENANCE APPROACHES IN THE LITERATURE

Article	Objective	Methods Used	Data Source	System Examined
Our research	Failure prediction using Weibull distribution	Weibull distribution, tests for different equipment sizes	Simulated data (2, 5, 15, 40, 100 units)	Individual machines and full system
Çınar et al. [1]	Predictive maintenance	Machine Learning, analysis of data from smart sensors	Sensor data, transmitted over a network	General machines
Karuppusamy [2]	Predictive maintenance scheduling	Decision tree, Random Forest	Sensor data	Manufacturing equipment
Luo et al. [3]	Two-stage prediction and scheduling	Neural network	Sensor data	Industrial equipment
Wan et al. [4]	Big data analysis for preventive maintenance	Big data-based analysis and forecasting	Cloud-based system	Manufacturing environment
Bastos et al. [5]	Testing predictive algorithms	Rapid Miner, data mining algorithms (alarm, maintenance actions)	Single machine data	Single machine, multi-machine planned
Dangut et al. [6]	Aircraft maintenance and scheduling	Deep Learning, Markov Decision Process	Aircraft data	Aircraft maintenance
Irinyi & Cselkó [7]	Effectiveness of maintenance strategies	Testing constant and variable interval strategies	Artificially created faulty data	Simulated fault-repair system
Shimada & Sakajo [8]	Building maintenance	Time series analysis, statistical approach	Building condition data	Buildings
Wang et al. [9]	ATM error prediction	Classification algorithms (XGBoost, Random Forest, Ada Boost M1, LibSVM)	ATM data	Automated Teller Machines (ATMs)
Martins et al. [10]	Analyzing the condition of a paper industry press	K-Means, Hidden Markov Model (HMM)	Manufacturing data	Paper industry drying press

Bastos, P. et al. [5] investigated maintenance forecasting, during which data analysis algorithms are used. It utilizes the various data mining predictive algorithms found in Rapid Miner. The logical structure of the developed maintenance system consists of three parts: alarm, maintenance measures and predictive maintenance. The authors used the developed prototype system on a single machine, and they plan to apply it to several machines.

Dangut, M.D. et al. [6] investigated aircraft maintenance and maintenance scheduling. Deep learning techniques are used for the problem. In the system, the data is first subjected to a pre-processing step. Then the data is divided into two parts (training and test) and classification is used, and the classification algorithms are evaluated according to efficiency. The developed system was tested by the authors on datasets of real airplanes. Markov Decision Process (MDP) was used as failure prediction algorithm.

D. Irinyi and R. Cselkó [7] examined the effects of constant and variable interval maintenance strategies and compared their effectiveness. Tasks are examined through artificially created faulty data. In the first method, a certain number of errors are corrected, and in the second method, the authors prevent a certain number of errors.

Shimada, J., and Sakajo, S. [8] examine the maintenance of building facilities prone to collapse. The condition of the buildings is diagnosed with time series data. A statistical approach is used to determine when maintenance is required.

Wang, J. et al. [9] investigated predictive maintenance and present a classification-based error prediction method. The authors give an example of ATM maintenance in their article. The authors used the following classification algorithms in the system: XGBoost, Random Forest, Ada Boost M1, and LibSVM (which are available in Weka).

Martins, A. et al. [10] presented a maintenance task of a paper industry drying press as an example. The data was collected every minute over three years and ten months. Based on the data, the model classifies the status of the device into the following categories: "Proper operation", "Warning status" and "Device error". Data cleaning and normalization are performed by the authors. Then the following methods are used to determine the states: K-Means and Hidden Markov Models (HMM).

III. Weibull-distribution

The Weibull distribution [11] is one of the probability distributions widely used in statistical modelling and data analysis. It is particularly popular in industrial reliability analysis and failure prediction. The Weibull distribution is often used to model events over time, such as machine failures, deaths, or product lifetimes. The advantage of the Weibull distribution is that it flexibly adapts to different data sets and reliably describes changes over time that often occur in real life. The following notations are used:

- η – life expectancy
- $F(x) = P(\eta < x)$ – distribution function

There are several common notations for the parameters of the Weibull distribution. The use of different notations is clearly explained by the fact that the Weibull distribution has been used very widely in a wide variety of scientific fields, as well as the fact that many different ways of determining the parameters are known, and the rewriting of the variables for each solution results in significant simplifications.

$$F_c(x) = \{1 - exp(-x^c),\ if\ x \geq 0,\ 0,\ if\ x < 0 \quad (1)$$

The paper uses the above notation to denote the standard Weibull distribution. From this, the distribution of linear transforms results in the following formula:

$$F_c(\tfrac{x-a}{b}). \quad (2)$$

This family of distributions is also a three-parameter, from which c is the so-called shape parameter (type parameter). However, it must be an asymmetric distribution.

1. In the case of the distribution, if $c = 1$ then the exponential distribution, if $c = 2$ the Rayleigh distribution, while $c = 3.57$ the distribution becomes nearly symmetrical and closely approximates the normal distribution. With a suitable parameter choice, it is also possible for the Weibull distribution to closely approximate the lognormal and Γ -distributions [12].

2. There are many methods for determining the parameters, but they are not robust on the one hand, and are difficult to handle on the other. A good example of difficult handling is parameter determination using the momemtum method. If ξ a, b, c are a random variable with parameter Weibull distribution, then

$$E\left((\xi - a)^k\right) = b^k\Gamma\left(1 + \frac{k}{c}\right), \quad (3)$$

i.e.

$$\mu = E(\xi) = a + b\Gamma\left(1 + \frac{1}{c}\right)\ \sigma^2 = E(\xi - \mu)^2 = b^2\left(\Gamma\left(1 + \frac{2}{c}\right) - \Gamma\left(1 + \frac{1}{c}\right)^2\right)\ \alpha_3 = E\left(\left(\frac{\xi-\mu}{\sigma}\right)^3\right) = \frac{\Gamma\left(1+\frac{3}{c}\right)-3\Gamma\left(1+\frac{2}{c}\right)\Gamma\left(1+\frac{1}{c}\right)+2\Gamma\left(1+\frac{1}{c}\right)^3}{\left(\Gamma\left(1+\frac{2}{c}\right)-\Gamma\left(1+\frac{1}{c}\right)^2\right)^{3/2}} \quad (4)$$

The μ, σ^2, α_3 can be easily determined from a given sample, the system of equations is not easy to solve, and it does not always have a solution [13].

Investigating with partial cases first, i.e. we assume that either the value of c or a is known.

1. If the value of c is known, then the value of a and b can be easily estimated from the previous system of equations, but the estimate will not be robust, because the average used to approach the expected value is not robust. On the other hand, the method of robust moments based on the distribution function gives a good estimate, since the distribution function is easy to handle and the type parameter is known. And so we already have a robust method, e.g. to the exponential distribution, to the Rayleigh distribution.

2. If a is known, then our sample or the distribution function can be transformed into another type of distribution with the following transformation [14,15]:

$$\eta = (\xi - a) \quad (5)$$

then the distribution function is the following

$$G(y) = 1 - exp\left(-exp\left(\frac{y-T}{s}\right)\right), \quad (6)$$

where $T = b$ is the location parameter and $s = \frac{1}{c}$ is the scale parameter. With this, we not only got a robust estimation option for the scale and shape parameters of the Weibull distribution, but also a simple option compared to the relatively complicated methods.

For the case when all three parameters are unknown, it was not possible to develop a truly robust method, because it was not possible to provide a really good robust estimate for the location parameter a or the shape parameter c separately. However, the following simple procedure can be used well.

The value of the parameter a from the sorted sample can be estimated with the following equation:

$$a_n^{(0)} = \xi_1^{\star} - 2(\xi_2^{\star} - \xi_1^{\star}) \quad (7)$$

This can be further accelerated by not using the median and the median absolute deviation as a starting solution, but the

$$1 - e^{-1} \quad and \quad 1 - e^{-0.5} \quad (8)$$

using quantiles (ordered sample elements) belonging to probabilities

$$T_n^{(0)} = (\xi_k^{\star} - a) \quad and \quad s_n^{(0)} = \frac{(\xi_k^{\star}-a)-(\xi_l^{\star}-a)}{(2)} \quad (9)$$

using starting estimates where

$$k = \left[n\left(1 - e^{-1}\right) + 0.5\right]$$

$$and \qquad l = \left[n\left(1 - e^{-0.5}\right) + 0.5\right]. \quad (10)$$

Computer experience shows that if the number of sample elements is greater than 100, then the values of the parameters can be easily recovered from the pseudo-random numbers. The statistic value of ω_n^2 is usually less than 0.2. This shows that the fit is acceptable at almost all significance levels.

For example, the computer runs to return the theoretical result that a sample with a normal distribution can be well approximated by a Weibull distribution.

If we use the suggested quantile estimates instead of the median and median absolute deviation, the number of iteration steps is approx. halved.

Inference Process

The Weibull distribution allows for flexible modeling of failure probabilities for individual components. During inference:

- Input Data: The model starts with time-to-failure data for each unit or system component. This data is either collected empirically or generated synthetically, as in your test cases.
- Parameter Estimation: The Weibull parameters (a, b, c) are estimated using robust methods when possible. Depending on the available information:
    - If c (shape parameter) is known, robust estimation techniques can calculate a (location) and b (scale) efficiently.
    - If a (location parameter) is known, transformations simplify the estimation of b and c.
    - If all parameters are unknown, initial estimates are derived using quantiles ($1 - e^{-1}$ and $1 - e^{-0.5}$) to accelerate computation and reduce iteration steps.

The outcome of this process is a probability distribution for each component's time-to-failure.

Aggregation Using Dependency Graphs

To determine the system-level failure probability, the model aggregates individual unit distributions based on their relationships within a dependency graph:

Graph Representation: Nodes represent components (units), and edges define dependencies (e.g., whether one component's failure directly impacts another).

System Failure Rules:

- Series Configuration: The system fails if any component fails.

$$P_{system} = 1 - \prod_{i=1}^{n} (1 - F_i(t)) \quad (11)$$

where $F_i(t)$ is the failure probability of the i-th component

- Parallel Configuration: The system fails only if all components fail.

$$P_{system} = \prod_{i=1}^{n} (1 - F_i(t)) \prod_{i=1}^{n} (1 - F_i(t)) \quad (12)$$

IV. Test tesults

This section presents the test results. During the research, test runs were conducted using the following datasets: 2 units, 5 units, 15 units, 40 units, and 100 units. The datasets consist of randomly generated test data, designed to include parameters relevant to building operations.

In this context, a 'unit' refers to individual components in the case of a single machine or, for a system, different machines that collectively form the system. Units represent elements that function together within a system. A failure in a single unit can potentially impact the entire system's operation.

The program is self-developed, with a Python backend and Angular frontend. In Python, we used the Reliability library for the operations in connection with distribution. The frontend was coded in Angular, and the Chart.js library was applied for the charts.

The test results is represented on two types of diagrams. The first is the failure-unit probabilities diagram, where the units are listed as individual objects when determining the failure probabilities, and the failure-tree probabilities diagram, where the failure probability of individual units is already determined by the failure probability of those units determines what it depends on.

A. 2 units

In this test case, the system contains 2 units. The start date of the measurements is: 28.01.2014, and the end date is: 28.01.2024. 10 years have passed between the start date and the end date of the measurements. And approx. 90-120 days passed between the two measurements.

The forecast starts from 28.01.2024 and does it every 10 days, and it makes a forecast for a total of 100 days from the start day.

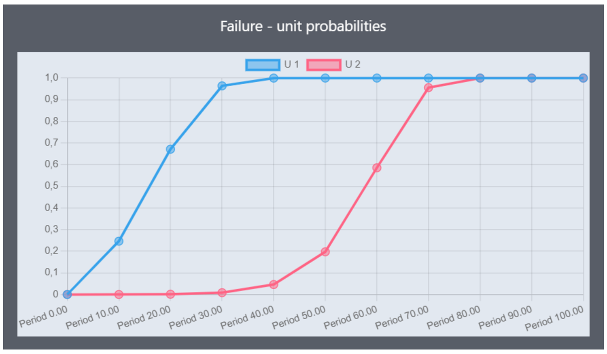

Fig. 3. Failure-unit probabilities diagram for 2 units

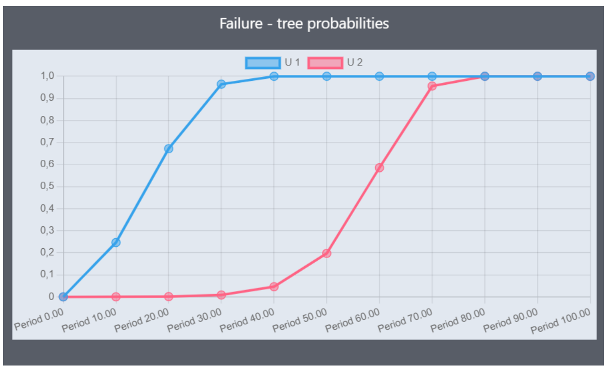

Fig. 4. Failure-tree probabilities diagram for 2 units

In the figure above, we can see that the failure-unit probabilities and the failure-tree probabilities diagrams are the same. Even at the starting time, both units have a 0 probability of failure. U1 fails at day 40 with probability 1, and U2 fails at day 80 with probability 1.

B. 5 units

The measurements for 5 units are also started on 28.01.2014. and lasted for 10 years. 90-120 days also passed between each measurement.

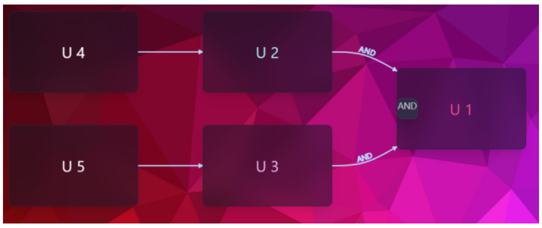

Fig. 5. Dependency graph for 5 units

It can be seen that U1 has an AND relationship with U2 and U3. This means that if U2 or U3 fails, then U1 will fail as well. U2 depends on U4, while U5 depends on U3, which means that if U4 fails, so does U2, and if U5 fails, U3 also fails.
The forecasts are begin from 28.01.2024.

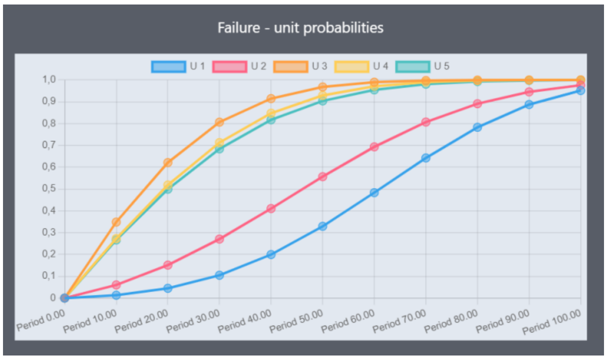

Fig. 6. Failure-unit probabilities diagram for 5 units

The Figure 6 is the figure of failure-unit probabilities, which shows that U3 has the earliest probability of failure, and U1 has the lowest probability of failure. As the period increases, the probability of failure increases. While the failure rate is 0 at the start time, this value is often 100 or close to 100 days after the start time, so it is almost certain that the individual units will break down.

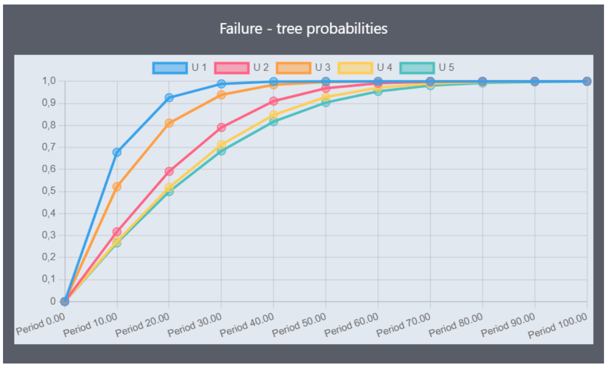

Fig. 7. Failure-tree probabilities diagram for 5 units

The failure-tree probabilities diagram already takes into account the dependence on other units for the failure probabilities of individual units. This type of diagram always gives greater or equal failure values than the failure-unit probability diagram, since it no longer examines the individual units by themselves, but also includes the dependence on other units in the failure probability.
The values of the failure-tree probabilities diagram are already different from the values above. Here, U1 has the highest probability of failure (it also depends on U2 and U3, which also depend on U4 and U5, so U1 actually depends on all other units). U4 and U5 have the lowest failure probabilities, since these units do not depend on the others. Here, U1 already fails with a probability of 1 in the 30th period, while the failure probability of each unit becomes 1 around the 80th period.

C. 15 units

The measurements started on 28.01.2014. The measurements were created every 90-120 days for 10 years.

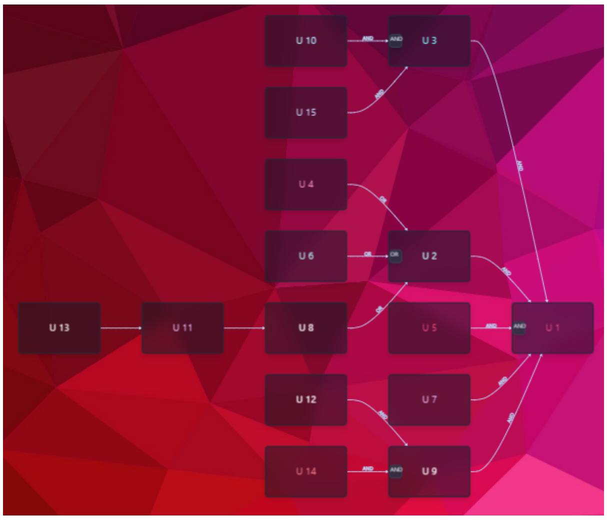

Fig. 8. Dependency graph for 15 units

The diagram above shows the graph representation of the system. The U1 directly or indirectly depends on each unit.

This unit is AND connection with U3, U2, U5, U7 and U9. U3 is also AND connection with U10 and U15. U2 is OR connection with U4, U6 and U8. U9 is AND connection with U12 and U14. In the case of the AND connection, if one unit fails, the given unit will also fail. In the case of an OR relationship, the given unit fails if all the units on which it depends are error.

The forecasts started from 10.01.2023 and forecasts were created every 10 days for 150 days.

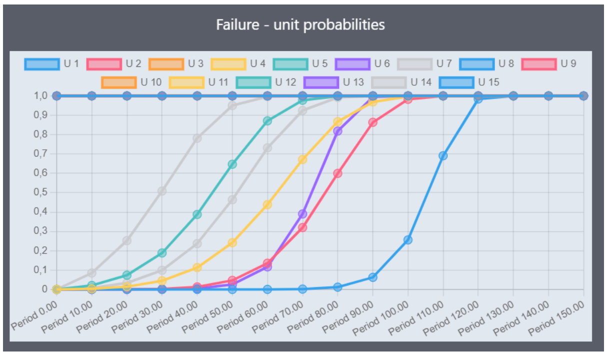

Fig. 9. Failure-unit probabilities diagram for 15 units

The failure-unit probabilities diagram is shown in the figure above. Accordingly, it can be seen that some units will fail from the start date of the forecast. While many units will certainly not fail even when the estimates begin. However, approximately 120 days from the start of the estimate, all units will be defective. It can be seen that some units break down later, while other units break down earlier (on their own, not as a dependency system).

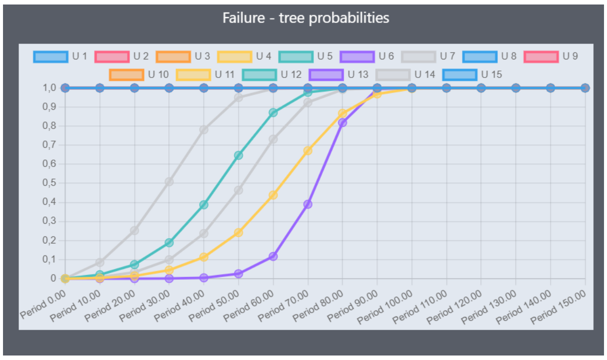

Fig. 10. Failure-tree probabilities diagram for 15 units

The failure-tree probabilities diagram is already narrower, because here we also take into account dependencies. So if a given unit would not be fail in itself at a given time, but the unit (or units) it depends on is fail, then the unit will also fail on this diagram. In the graph above, we can see that the U1 unit is the one that depends on all other units, so it is already fail at the beginning of the forecast. The diagram shows that certain units have a 0-probability failure at the beginning of the forecast, but after 90-100 days, all units will fail. It can also be seen that there are only 6 units that do not fail at the start of the forecast.

D. 40 unit

The previously presented systems were smaller, but the created software is also suitable for analyzing large systems. In this test run, we created a system with 40 units. The beginning of the measurements is 28.01.2014., and the measurements lasted for 10 years, the measurements were made every 90-120 days here as well.

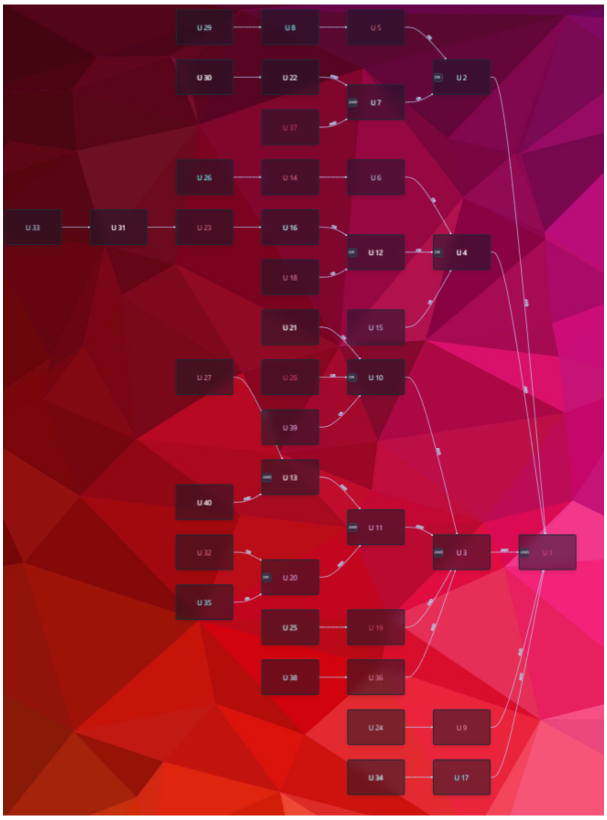

Fig. 11. Dependency graph for 40 units

The figure above presents the system in a graph structure. Unit U1 depends directly or indirectly on all other units.

Here too, the starting date was 10.01.2023, forecasts were created every 10 days for 150 days.

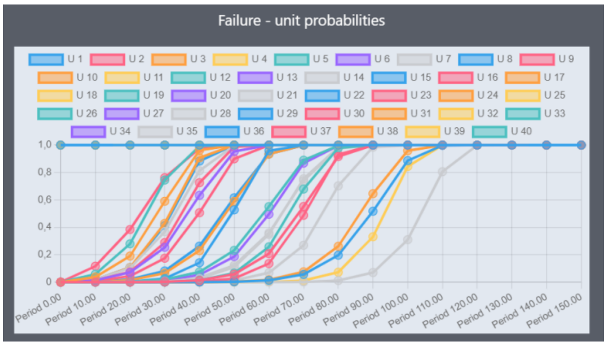

Fig. 12. Failure-unit probabilities diagram for 40 units

The figure above is the failure-unit probabilities diagram. Here we can see the individual failure probabilities of each unit. Some units fail already at the start of the estimate, while other units approx. errors occur only 120 days after the start of the estimate. We can see how different the failure probabilities of individual units are.

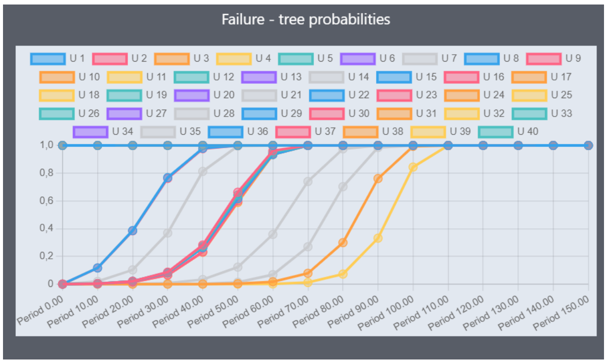

Fig. 13. Failure-tree probabilities diagram for 40 units

In the case of the failure-tree probabilities diagram, many units will already be fail at the start of the measurement, because here the system also includes the dependency relationships in the evaluation. It can be seen that individual units fail only 90-110 days from the start of the estimate. This is possible for units that do not depend on other units (they may already depend on units, but the number of dependencies is not large).

E. 100 unit

The last data line shows a system of 100 units. The measurements started on 28.01.2014, lasted for 10 years and the measurement was made every 90-120 days.

For the forecasting the 2023.10.01 was chosen as the starting date, and forecasts were created every 10 days for 150 days from the starting date.

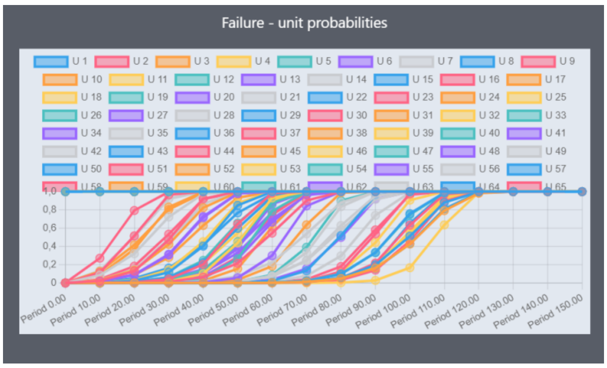

Fig. 14. Failure-unit probabilities diagram for 100 units

The figure above is the diagram of failure-unit probabilities, where the failure probabilities of the individual units were also illustrated. Here we can see that most of the units do not fail even at the start time of the forecast (0 probability of failure). However, by the 120th day from the forecast start time, almost all units will be faulty.

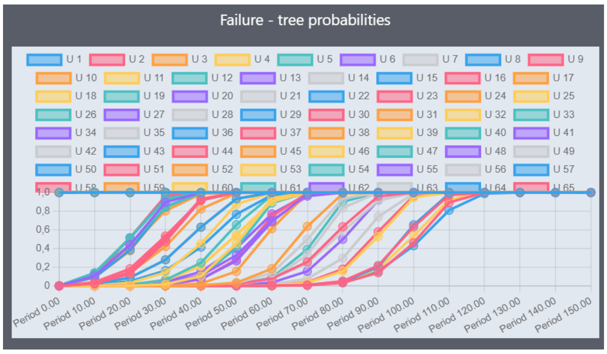

Fig. 15. Failure-tree probabilities diagram for 100 units

With the created software, we can create a diagram for a specific unit, then only the unit and its dependencies (direct and indirect dependencies) will be visible. This is also good because in practice it is possible that we only want to know about the failures of a few units. Furthermore, even in the case of this large example, it is easier to view the entire system in its small details, because the diagrams projected on a part of the system are much more readable than the above-mentioned complete system diagram.

For example, for the U60 we get the diagrams below.

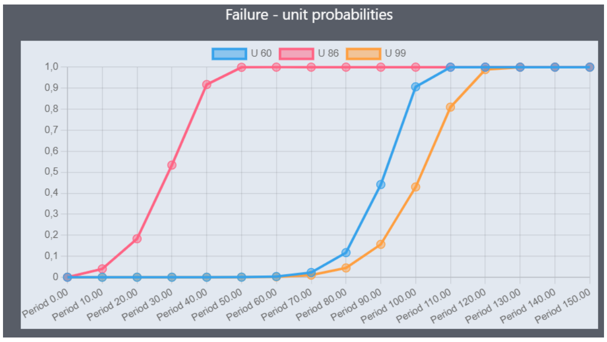

Fig. 16. Failure-unit probabilities diagram for U60

The diagram above shows the failure-unit probabilities diagram. U60, U86 and U99 are also represented because U60 depends on these units. It can be seen that U60 alone begins to fail drastically from the 70th day from the start of the forecast, and will definitely fail by the 110th day. U86 starts to fail drastically from the 10th day from the beginning of the forecast and the glaze deteriorates by the 50th day. U99 starts to fail from the 80th day and will definitely fail by the 120th day.

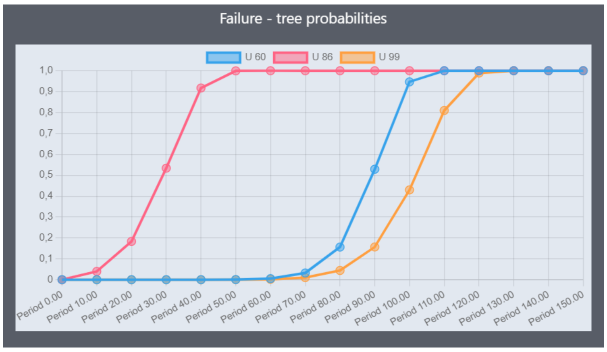

Fig. 17. Failure-tree probabilities diagram for U60

The failure-tree probability diagram is completely similar to the failure-unit diagram. This is possible because U60 is AND connected to the other two units, and while U86 will fail before U60, U99 will fail later, and U60 will still be functional (because of the OR connection) if the one of the units it depends on is still working.
After that, the paper presents another example: the following diagrams show the results for U6.

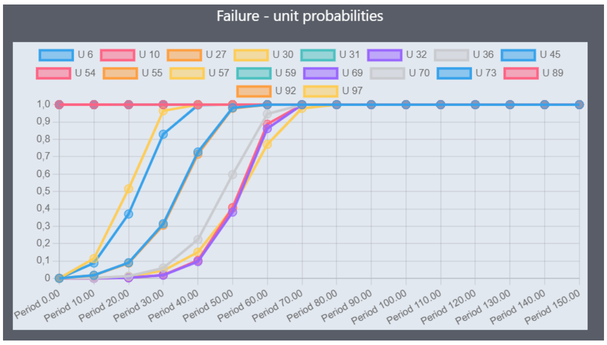

Fig. 18. Failure-unit probabilities diagram for U6

It can also be seen from the figure above that U6 has many dependencies If we compare it with the failure-tree probabilities diagram below, individual units fail sooner (due to dependencies) than in the diagram above.

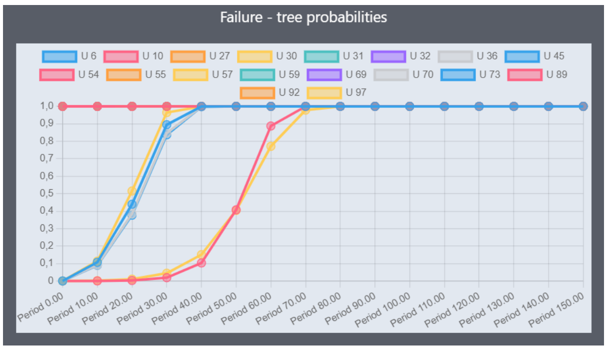

Fig. 19. Failure-tree probabilities diagram for U6

V. Conclusions

This paper presents the failure estimation of complex systems with Weibull distribution. During the analysis of the examined data series, it can be concluded that the Weibull distribution can be effectively used to determine the failure probabilities. In the data sets examined in the paper, it can be observed varying failure times for different units. This data allows us to determine the probability of machine failure over time and optimize maintenance schedules to ensure the uninterrupted operation of industrial processes. the article presented tests for the following data sizes: 2 units, 5 units, 15 units, 40 units, 100 units.
Overall, the Weibull distribution is efficient in industrial reliability analysis and failure prediction, which can help industrial companies optimize their operations and increase their competitiveness in the dynamic business environment.
Future research direction is the introduction of the developed system in a production environment and the comparison of the results given by the Weibull distribution with the results given by other algorithms.

Acknowledgement

This research was partially supported by National Research Development and Innovation Office, Hungary, grant number 2020-1.1.2-PIACI-KFI-2020-00147.

References

[1] Z. M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei, “Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0.” *Sustainability*, vol. 12, no. 19, 8211, 2020, doi: 10.3390/su12198211

[2] D. P. Karuppusamy, “Machine learning approach to predictive maintenance in manufacturing industry-a comparative study.” *Journal of Soft Computing Paradigm (JSCP)*, 2(04), pp. 246–255, 2020, doi: 10.36548/jscp.2020.4.006

[3] M. Luo, H. C. Yan, B. Hu, J. H. Zhou, and C. K. Pang, “A data-driven two-stage maintenance framework for degradation prediction in semiconductor manufacturing industries.” *Computers & Industrial Engineering*, vol. 85, pp. 414–422, 2015, doi: 10.1016/j.cie.2015.04.008

[4] J. Wan, S. Tang, D. Li, S. Wang, C. Liu, H. Abbas, and A. V. Vasilakos, “A manufacturing big data solution for active preventive maintenance.” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2039–2047, 2017, doi: 10.1109/TII.2017.2670505

[5] P. Bastos, I. Lopes, and L. Pires, “Application of data mining in a maintenance system for failure prediction.” *Safety, Reliability and Risk Analysis: Beyond the Horizon: 22nd European Safety and Reliability*, pp. 933–940, 2014

[6] M. D. Dangut, I. K. Jennions, S. King, and Z. Skaf, “Application of deep reinforcement learning for extremely rare failure prediction in aircraft maintenance.” *Mechanical Systems and Signal Processing*, 171, 108873, 2022, doi: 10.1016/j.ymssp.2022.108873

[7] D. Irinyi and R. Cselkó, "Analysis of failure prediction methods," *6th International Youth Conference on Energy (IYCE)*, Budapest, Hungary, 2017, pp. 1–5, 2017, doi: 10.1109/IYCE.2017.8003692.

[8] J. Shimada and S. Sakajo, “A statistical approach to reduce failure facilities based on predictive maintenance.” In *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 5156–5160, 2016, doi: 10.1109/IJCNN.2016.7727880

[9] J. Wang, C. Li, S. Han, S. Sarkar, and X. Zhou, "Predictive maintenance based on event-log analysis: A case study." *IBM Journal of Research and Development*, vol. 61, no. 1, 11–121, 2017, DOI: 10.1147/JRD.2017.2648298

[10] A. Martins, I. Fonseca, J. T. Farinha, J. Reis, and A. J. M. Cardoso, "Maintenance prediction through sensing using hidden markov models—A case study." *Applied Sciences*, vol. 11, no. 16, 7685, 2021, DOI: 10.3390/app11167685

[11] J. Wang, and H. Yin, "Failure rate prediction model of substation equipment based on Weibull distribution and time series analysis." *IEEE access*, 7, 85 298–85 309, 2019, DOI: 10.1109/ACCESS.2019.2926159

[12] E. Dékány, J. Gedeon, L. Gillemot, "A Weibull-eloszlás néhány műszaki alkalmazása I.", *GÉP,* pp. 12-15, 1981

[13] I. Deák, "Véletlenszámgenerátorok és alkalmazásaik," *Akadémiai Kiadó,* Budapest, 1986.

[14] S. S. Shapiro, C. W. Brain, "Some new tests for the Weibull and extreme value distributions" *Coll. Math. Soc. J. Bólyai, Goodness-of-fit, Debrecen*, pp. 511–527, 1984

[15] V. M. Zolotarev, "Független valószínűségi változók összegeinek modern elmélete" *Nauka, Moszkva*, 1986

Anita Agárdi is senior lecturer at the Institute of Informatics, Faculty of Mechanical Engineering and Informatics, University of Miskolc, Hungary. She received a Ph.D. degree in Computer Science from the University of Miskolc in 2023. Her research interest include optimization algorithms, data mining and logistics.

Károly Nehéz got his MSc in mechanical engineering at the University of Miskolc, Hungary, in 1997 and PhD degree in software engineering in 2003. He currently works as an associate professor at the Institute of Computer Science, head of the institute since 2019. His primary research interest is Software Engineering, and AI algorithms.