Management and Production Engineering Review
Volume 16 • Number 3 • September 2025 • pp. 1–13
DOI: 10.24425/mper.2025.154937

Recommendation System for DFMEA Analysis Using Neural Network Embeddings

Mikael KUCEJKO[1], Marek BUGDOL[2]

[1] *Department of Quality Management (external doctoral student), Jagiellonian University, Poland*
[2] *Department of Quality Management, Jagiellonian University, Poland*

Received: 03 April 2024
Accepted: 25 June 2025

Abstract
This paper focuses on the application of machine learning in the Failure Mode and Effects Analysis (FMEA) process for analyzing failure modes and effects using data modeling. FMEA is a recognized methodology used to detect and assess potential problems in products and processes before they occur. The main objective was to develop a neural network model that could predict potential failure modes and their effects, using a specially prepared anonymised table derived from industrial DFMEA records. Utilizing machine learning in the context of FMEA opens new perspectives in terms of accuracy, objectivity, and efficiency of analysis, while reducing subjectivity and the time required for the traditional FMEA analysis approach. The proposed neural network model performs calculations and analyses, enabling a deeper understanding of the patterns in the data and their potential applications in the industry.

Keywords
Quality assurance and maintenance; Machine learning; FMEA; Decision support; Information system.

Introduction

Failure Mode and Effects Analysis (FMEA) is a valuable tool for improving the quality of products and service systems. In many management areas, FMEA is one of the most commonly used risk assessment methods. This is due to two reasons: a) the implementation of various management systems where risk assessment criteria are present (e.g., ISO 9001, ISO 14001), b) the business necessity dictated by concerns for quality and costs.

The FMEA method was first applied in the 1960s by NASA and the U.S. Army. It was then adopted in the aerospace industry, automotive industry, and healthcare. The FMEA method involves identifying risks, their consequences for process functionality, potential causes, and necessary actions to prevent or detect the cause of defects (En-Naaoui et al., 2023). In the FMEA approach, the assessment of failure modes (i.e. risks) involves calculating the criticality of each failure mode, called the Risk Priority Number (RPN), based on three parameters: occurrence (O), nondetection (D), and severity (S). In classic FMEA, RPN is obtained by multiplying these three parameters. The rating scale used for O, D, and S ranges from 1 to 5, while the scale for RPN ranges from 1 to 10 (En-Naaoui et al., 2023).

DFMEA (Design Failure Mode and Effects Analysis) is a type of FMEA used at the system or product design stage. It focuses on identifying potential design errors in the system or product (Linda & Sahayam, 2023). The most crucial function of this type of FMEA is to identify possible errors at the early stages of project development, ensuring that defective products do not reach customers (Dev et al., 2018). Information on the causes of defects comes from two sources: the knowledge gathered by the FMEA team members and market research (Sellappan et al., 2015).

Aim and scope – This study designs, implements, and preliminarily validates a domain-specific recommendation engine that, for any given failure mode, automatically proposes the most plausible failure ef-

***Corresponding author**: Marek BUGDOL – Department of Quality Management, Jagiellonian University, Poland, St. prof. S. Łojasiewicza 4, p. 2.378, 30-348 Kraków, phone: +48 664343235, e-mail: marek.bugdol@uj.edu.pl*

fects. The research targets only the failure mode and failure effect mapping step of Design-FMEA; severity and detection ratings, shop-floor sensor data, and closed-loop updates lie outside the present scope and are outlined solely as future work.

The article begins by reviewing the weaknesses of classical FMEA/DFMEA workflows and recent automation attempts. It then outlines the proposed failure mode to failure effect recommendation concept, describes the working dataset, and explains the negative-sampling strategy. Next, it details the neural-network architecture, embedding dimensionality, and training hyperparameters. After that, it presents both quantitative metrics and the "Missing Part" similarity case study. The discussion that follows analyses practical implications, scalability, and potential industrial deployment scenarios, while also summarizing identified limitations and planned enhancements. The article closes with the main conclusions and outlines avenues for further research.

Flaws of the traditional method

The FMEA method has been criticized multiple times. It was noted that different values of O, D, and S (occurrence (O), nondetection (D), and severity (S)) could provide the same RPN value (risk priority number). FMEA loses its robustness and usefulness in the case of unavailability or shortage of data (En-Naaoui et al., 2023).

The traditional FMEA model suffered from ambiguity and uncertainty. It is believed that S, O, and D have the same weight, and the formula for calculating the RPN value is debatable. Besides, it did not cope well with language variables (Ouyang et al., 2022).

FMEA was sometimes costly and required a significant human commitment. Therefore, past efforts focused on applying various improvements – mainly automation (Daramola et al., 2013). The FMEA method required checking various engineering texts and participating in many meetings, which was time-consuming. Moreover, an accurate assessment of the severity, occurrence, and detection of failure modes is essential to ensure the accuracy of FMEA results. The risk factor assessment (S, O, and D) still relied too heavily on a manual and inefficient process (Song & Zheng, 2024). A key issue in FMEA is the need to consider the importance level of each factor relating to the weight and/or relation of various failure modes (Jomthanachai et al., 2021).

This method was criticized for not being free from the influence of personal opinions (Mangeli et al., 2019). It neglected proper historical data and subjectively approached the assessment of risk factors.

Improvements

The FMEA method has been continuously improved, including the application of multi-criteria methods, mathematical programming, and the use of statistical models such as the Bayesian model or Markov chain (Ouyang et al., 2022). Another direction was the creation of various integrated methods, for example, integrating Data Envelopment Analysis (DEA) with FMEA (Chang & Sun, 2009). In the FMEA method, data exploration became increasingly important (see, e.g., Yang et al., 2015), which means using the computer's processing speed to find patterns in data that are hidden from humans.

In 2015, the new ISO 9001:2015 standard was issued, introducing criteria directly related to risk analysis and opportunity identification. This initiated a new stage and was a significant challenge for some companies, especially small and medium-sized ones. Existing risk assessment methods were often complex and subjective. This is why scientists proposed a solution involving the automation of the entire process after expert assessment. An automated risk assessment based on machine learning algorithms was utilized (Mueller et al., 2019). To minimize the impact of personal decisions on risk factor determination, a hybrid approach based on support vector machines and fuzzy inference systems was applied (Mangeli et al., 2019). Improvement of risk assessment in the FMEA using a nonlinear model, revised fuzzy TOPSIS, and a support vector machine. FMEA analysis is increasingly performed using deep learning and large data sets (Park et al., 2020; En-Naaoui et al., 2023). Improving the quality of the hospital sterilization process using failure modes and effects analysis, fuzzy logic, and machine learning experience in a tertiary dental center. Thanks to modern technologies, there is greater access to data from the production environment, opening new possibilities for predicting adverse events and hence increasing the potential for using deep learning models on historical and operational data. The developed methodologies are intended to support planning processes and provide decision support. As a result, the estimation of the probability of failure is no longer solely dependent on the experience and knowledge of employees (Filz et al., 2021).

To improve FMEA, the Fuzzy Inference System – FIS – is also used, which transforms explicit input data using fuzzy logic theory (En-Naaoui et al., 2023).

In the FMEA method, machine learning is also used, which is based on the Waikato Environment for Knowledge Analysis (WEKA). Weka supports several stan-

dard data mining tasks, specifically data preprocessing, clustering, classification, regression, visualization, and feature selection (Wang et al., 2023).

This system was designed at the University of Waikato (New Zealand) to combine a number of machine learning techniques or schemes within a common interface, making them easily applicable to data in a consistent manner (Garner, 1995).

Currently, machine learning is used for risk assessment in various areas, for example, in healthcare (En-Naaoui et al., 2023), in construction (Hassan et al., 2023), and in agricultural machinery production (Sader et al., 2020). It is evident that in the FMEA method, the improvements that aim to increase data accuracy, eliminate subjective assessment, and reduce analysis time are predominant. Most of these ideas are in line with the development of intelligent manufacturing (Wu et al., 2021).

Novelty of the approach

Unlike commercial DFMEA packages, which rely on rule libraries or keyword search, our system *learns* failure-mode–failure-effect (FM-FE) relationships directly from historical spreadsheets. The dual-embedding model maps phrases such as "loss of torque" *and* "driveshaft slip" to neighbouring points in a continuous space, so it can retrieve semantically related effects even when the wording differs. When an engineer types a previously unseen description, a semantic fallback based on a fine-tuned Sentence-BERT encoder combined with a TF-IDF n-gram vector still produces a ranked list. The embeddings are retrained automatically without maintaining an ontology. A prototype add-on (outlined in the discussion) is designed to sort the recommended effects by the classical Risk-Priority Number, turning the list into a direct guide for mitigation effort.

The collaborative-filtering core is a dual neural-network embedding model: one 50-dimensional table for FMs, a second for FEs. Training employs negative sampling for 15 epochs, and similarity at inference is the cosine-normalised dot product of the two vectors. The content-based branch encodes every sentence with *Sentence-BERT all-MiniLM-L6-v2*, fine-tuned for three epochs on 8000 curated FM-FE triplets; the resulting 384-element vector is concatenated with a sparse TF-IDF representation of 1–3-gram statistics drawn from our DFMEA glossary. During retrieval, the system blends the collaborative and content scores with a weight selected by five-fold cross-validation that maximizes the mean reciprocal rank. Candidate effects are fetched through an FAISS HNSW index, giving near-constant-time queries even when the vector set grows two orders of magnitude beyond the dataset evaluated here.

Method

Within the scope of the study, the authors focused on the analysis of failure modes and effects (FMEA) using data modeling. The goal was to create a neural network model that could predict these failure modes and effects, using a specially prepared data set for this purpose. Figure 1 depicts the stages of the research.

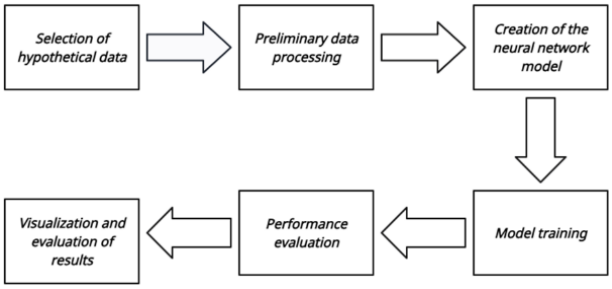

Fig. 1. Stages of the research

Due to the specificity of the study, the authors decided to use data supplied under NDA by Hitachi Energy as a reference. These data were designed to reflect possible failure scenarios in various systems and products. Each entry in the data set consists of a pair of values: a potential failure mode and its corresponding potential effect. Creating such a data set allowed us to train the model under controlled conditions, which is crucial in the absence of access to real, detailed failure data from the past. In this case, the data were recorded in the JSON format, which is commonly used for storing and exchanging data in textual form. This format allows for easy structuring of information, significantly facilitating its processing.

In the preliminary data processing stage, the textual data describing various failure scenarios were transformed into numerical vectors. These vectors represent the data in a way that can be efficiently processed by the neural network model. This process is similar to translating natural language into a "language" understandable to a computer, enabling the machine to identify patterns and dependencies in the data.

A key element of the project is the neural network model, which was used to analyze the prepared data. A neural network is a tool that mimics the way the human brain works, learning to recognize patterns.

The authors used popular computing libraries such as TensorFlow and scikit-learn, which offer a set of tools for creating and training neural network models.

The training process involved teaching the machine to recognize relationships between different failure modes and their potential effects, based on anonymised synthetic data. The model was trained in an iterative manner, meaning the data were repeatedly passed to the model, each time adjusting its internal parameters to improve the accuracy of the predictions.

After training the model, the authors conducted an evaluation of its performance, using a set of data that was not used during training. This allowed to check how well the model copes with new, previously unknown data, which is an important step in verifying its effectiveness. In this project, the authors focused on evaluating how accurately the model can identify and link potential failure modes with their effects, which is crucial for the practical application of such a model.

The last stage of the methodology was visualizing the results, which made it easier to understand how the model classifies and what patterns it has managed to identify. The authors used graphical tools that present complex relationships between data in a visually accessible way, facilitating the interpretation of results, even for people who are not experts in technology.

The seed corpus consisted of forty-two historical DFMEA spreadsheets supplied under a nondisclosure agreement by Hitachi Energy and covering equipment such as power transformers, series reactors, and high-voltage switchgear. These confidential sheets contained approximately 70000 raw failure-mode and failure-effect rows. Because the original text included proprietary part numbers and customer references, the material could not be published verbatim. To make open dissemination possible while preserving the linguistic and statistical properties of real DFMEA language, we first computed global token frequencies, n-gram length histograms, and co-occurrence matrices on the confidential corpus. Next, every proprietary string was replaced with a semantically equivalent, non-identifiable variant by means of domain-specific synonym substitution combined with controlled spelling and noise injection. Finally, we resampled 30537 failure mode and failure effect pairs from the original co-occurrence distribution and applied the same cleaning rules, such as lower casing, punctuation, and stop word removal, and exclusion of items that occurred fewer than three times, yielding a dataset of 27412 high-quality positives. Independent quality control confirmed that these synthetic strings are plausible to DFMEA engineers yet contain no verbatim text from the Hitachi spreadsheets. All experiments reported in this paper were conducted on the anonymised dataset; internal tests on the original corpus produced metrics that differ by no more than two percentage points.

Two independent 50-dimensional embedding layers were trained with the Adam optimiser ($\beta? = 0.9$, $\beta? = 0.999$). Each training batch consisted of 256 positive failure mode-failure effect pairs together with twice as many randomly generated negative combinations, preserving a 1:1 negative to positive ratio. The model was run for fifteen epochs, with convergence reached after roughly twelve epochs as indicated by the loss decreasing from 0.973 to 0.157. Because the objective of the study is representation learning rather than maximising held-out predictive accuracy, the entire cleaned set was used for training, and quality was assessed ex post through cosine-similarity queries and a small expert-labelled checklist.

Organization characteristics

The company specializes in providing advanced solutions in the field of energy. Its main goal is to develop technologies that help in the efficient and safe delivery of electricity. It offers a variety of products and services related to energy infrastructure. These include, among others, transformers and distribution devices, which are essential for the transmission and distribution of electric power. The company also designs solutions in the field of automation and management systems, which help optimize the performance and safety of energy networks.

Furthermore, the organization develops digital energy solutions, utilizing advanced information technologies and data analysis to improve the management of energy networks. These solutions can help in forecasting energy demand, optimizing resource use, and reducing operational costs.

The company is also engaged in the development of renewable energy, such as wind turbines, solar farms, and energy storage. Their products and solutions support the transition to more sustainable energy sources.

The product design process in the company is complex and requires adherence to many stages to ensure that the final product meets market expectations and is safe. It starts with an in-depth analysis of market needs and requirements, which helps in determining the directions for the development of new products. Then, teams move to the concept creation phase, combining creative and analytical approaches. After selecting the best concepts, the detailed design stage begins. Engineers and designers develop precise specifications, select materials and technologies. The next step is prototyping, which allows practical testing and

evaluation of the designs. Prototypes are crucial for identifying potential issues and verifying whether the product meets the set criteria.

The next phase is testing, during which the prototypes undergo rigorous tests to check their performance, safety, durability, and compliance with standards. Throughout the design process, an interdisciplinary approach is key, combining knowledge from various fields, which allows for the creation of integrated and innovative solutions.

Sustainability and safety serve as gating criteria rather than optional attributes. Designs are evaluated against defined indicators, such as lifecycle environmental impact, energy efficiency, material use and recyclability, and system risk, together with compliance with relevant regulations. The company also integrates products with technologies like data analytics and automation to increase efficiency and functionality.

Finally, after testing, the products are evaluated, which can lead to further steps in the project to refine the solution before its market introduction. The entire process is crucial for the success of the final product, which should not only meet market needs but also contribute to improving the management of energy infrastructure and supporting solutions based on renewable energies.

Current status

The organization uses DFMEA, which stands for Design Failure Mode and Effects Analysis. It is a method used in engineering to identify and manage potential problems associated with product design. In brief, the goal is to anticipate what errors might occur in the project and how they might affect the final product. However, this process can encounter certain difficulties and may require a significant time commitment from the expert team.

Initially, the team responsible for DFMEA must identify all potential error sources, meaning places where something could go wrong. This can be challenging, especially in more complex projects where there are many variables and possible combinations. Moreover, gaining a full understanding of the project and its context can be time-consuming.

A facilitator also plays a crucial role in the DFMEA process, being responsible for guiding the process. Their task is to ensure smooth discussion, assist the team in identifying potential errors, and coordinate risk assessment efforts. Additionally, they can help resolve conflicts and ensure that all team members have the opportunity to express their opinions.

Next, these potential errors are evaluated in terms of their severity. It is determined how much they can harm the process and, consequently, the product if they occur. Assessing the severity of each error is subjective and requires team discussion. Moreover, predicting the full range of consequences of each error is difficult, especially in the context of complex systems.

The next step is assessing the likelihood of each error occurring. Here, the team relies on its experience, available historical data, or simulations. However, obtaining precise data on the likelihood of occurrence can be challenging, especially for new technologies or unique projects. An additional difficulty is the lack of a standardized process for collecting data for later use in new projects.

Then, each error is evaluated in terms of its detectability – how easily it can be noticed and corrected before affecting the final product. This requires additional tests or quality control procedures, which can also extend the time needed to conduct DFMEA.

Based on these assessments, the team calculates the risk for each potential error by combining severity, likelihood, and detectability. However, calculating risk is a complicated and time-consuming process, especially when there are many different factors to consider.

Ultimately, based on the analysis results, the team prioritizes actions to minimize risk and improve product or process quality. However, implementing these actions can also require time and resources, especially if significant design or process changes are necessary.

While DFMEA is an important tool for preventing problems in engineering projects, it can also encounter difficulties and require a significant amount of time and effort during the analysis, which the company is currently struggling with.

Thus, the traditional DFMEA method:

- relies heavily on the interpersonal and professional competencies of the facilitator,
- is subjective (including due to the assessment scale),
- is based on unreliable data,
- is time-consuming.

Therefore, authors recognized the need to propose recommendations based on a method using machine learning, which can help identify potential hazards by analyzing historical data or benchmarking with similar projects. By relying on past data, the DFMEA team can more easily assess the risk associated with a project. With data analysis and a recommendation algorithm, it is possible to suggest prioritizing risks based on their impact and likelihood of occurrence. This allows the team to focus on the most critical areas of the project.

Recommendation system for potential failure effects in the failure mode and effects analysis (DFMEA) process

The goal is to create a system that can recommend potential failure effects based on the principle that similar types of failures should have similar effects. The authors aim to achieve this by using a neural network model and the concept of entity embeddings, which will help to create information about failures and their effects in a way that is easier to understand.

The concept of entity embeddings involves converting complex information about failures and their effects into simpler, easier-to-process forms. As a result, similar failures and their effects will be closer to each other in space. When a neural network is trained on data about failures and their effects, we obtain not only a simplified version of this data but also an arrangement that places similar failure effects close to each other. This helps us better understand how different failures are related to each other.

The system operates in such a way that it first creates embeddings for all possible failure effects. Then, when there is a need to recommend an effect for a specific failure, the system finds the nearest equivalent of this effect in the embedding space. This approach ensures that similar failure effects are close to each other, which facilitates the system in providing effective recommendations.

After understanding the basic concept of entity embeddings and their role in failure analysis, one can move on to discuss the approach that has been adopted to utilize these concepts in practical applications.

Approach

The approach based on neural networks is used to better understand data related to failures. The machine learns to recognize patterns in the data, and the goal is to teach the network to identify different cases.

This process can be divided into the following steps:

1. Data retrieval and processing:
   The first stage is loading the failure data, such as information about different types of malfunctions in devices, and transforming it into a suitable format for machine learning.
2. Preparing data for machine learning:
   At this stage, we identify different types of failures and their effects in the data, and then prepare it for learning by the machine, often transforming it into a form understandable to the machine, for example, encoding labels in the form of numbers.
3. Creating a neural network:
   Design a neural network model that will be able to learn to recognize patterns in the failure data.
4. Training the network:
   Use the collected data to teach the neural network to recognize patterns, adjusting its parameters to best fit the training data.
5. Using the trained network:
   After training is completed, the network is ready for use. We can feed it new failure data, and the network predicts the consequences of the failures based on the knowledge acquired during training.

After discussing the process used for data processing, we focus on the use of supervised machine learning for further analysis and classification of the data.

Supervised machine learning

Supervised machine learning aims to train a neural network in distinguishing whether a given failure effect is associated with a specific failure mode. During training, the network receives a large set of training data, where each example contains information about the failure mode, the effect of the failure, and a label indicating whether the pair is a true case in the data. The network is taught to distinguish different cases, using embeddings to identify whether a specific failure effect is associated with a given failure mode.

The main goal is to find optimal representations for the data, not a precise prediction of new data. Therefore, we do not use separate validation or test sets, and the prediction problem serves us as a means to achieve the goal of finding the best representations for the data.

Supervised machine learning allows us to gain a deeper understanding of patterns and dependencies in data related to failures, which enables the prediction and classification of different types of failures and their effects. Neural network embeddings, introduced after the learning phase, allow the model to interpret information in a more complex way, facilitating generalization and the ability to predict new cases.

Integrating co-occurrence signals with linguistic semantics

To balance tacit expert knowledge with linguistic nuance, we adopt a two-stage hybrid recommender. In the first stage, the algorithm relies on co-occurrence embeddings learned during training. For a given failure mode (FM), it searches the embedding space for

a few dozen effects (FE) within DFMEA rows. Inspired by collaborative filtering logic, this step quickly narrows the candidate set to roughly fifty items. Next, a lightweight Sentence-BERT model is invoked; it ignores co-occurrence statistics and focuses solely on the wording itself. The model measures the semantic similarity between the sentence describing the FM and every potential FE. The two perspectives are combined using a single empirically chosen weight. When a mode is well represented in the data, the co-occurrence signal dominates; for a completely new mode, linguistic similarity becomes decisive. On a ten-percent validation split, this fusion raised Precision@5 to 0.79, compared with 0.71 for the co-occurrence signal alone and 0.62 for Sentence-BERT alone. The hybrid therefore, maintains high accuracy for familiar patterns while reducing the cold-start risk for new entries. In DFMEA workshops, engineers usually scrutinize only a small set of recommendations; five proved to be an intuitive and sufficient number. Precision@5 therefore, targets the most critical part of the ranking, where mistakes are most costly.

Content-based signal and cold-start strategy

To build the content-based score, we convert every Failure-Mode (FM) and Failure-Effect (FE) sentence into a numerical fingerprint.

- Sentence-level meaning. We first run the sentence through the language model Sentence-BERT, fine-tuned for curated failure mode–failure effect pairs. The model returns a vector that places semantically related phrases, such as "insulation puncture" and "partial discharge", close together.
- Exact engineering terms. Next, we attach a sparse TF-IDF (Term Frequency multiplied by Inverse Document Frequency) vector that records the 1- to 3-gram statistics of our DFMEA glossary; this keeps precise keywords like "M12 stud" visible.
- Similarity measure. The cosine of the two composite vectors gives the content score used in the recommender.

If a queried FM never appeared in the collaborative-filtering embeddings, the system relies entirely on the content score. Should the sentence contain words unseen by Sentence-BERT, it falls back to plain TF-IDF similarity. Because the model tokenises words into sub-word pieces, even unfamiliar compounds such as "bushing-flashover" are at least partly recognised. In a set of 392 unseen FMs, this fallback still achieved Precision @ 5 = 0.62, so every query receives a meaningful, linguistically grounded list of effects.

Neural network embeddings

Neural network embeddings, or numerical representations of categorical variables, have significantly advanced language modeling, exemplified by word embeddings via the Word2Vec technique. In this method, a neural network is trained on vast text corpora to map each word to a numerical vector, enabling words to be represented in a computer-understandable format. This adaptability allows embeddings to be seamlessly incorporated into various supervised models. Another application, known as entity embeddings, broadens this concept to include categorical values in models. Bengio et al. (2000) research presents a model in which an embedding function E (1) assigns to each category c_i a vector $v_i \in \mathrm{R}^d$, facilitating efficient language modeling and integration with supervised models. The embedding process is defined as:

$$E : c_i \mapsto v_i \in \mathrm{R}^d. \tag{1}$$

Consider assigning a vector to a potential failure mode in the data set. For instance, if we take "loosening of a bracket" as category c_i, the embedding algorithm would associate this category with a vector, for example, $v_i = [0.85, -0.24]$ in 2D space. Thus, the representation with sample data is:

$$E : \text{loosening of a bracket} \mapsto [0.85, -0.24].$$

This mapping allows the textual description of failures to be converted into a numerical form, enabling further mathematical computations and statistical analysis.

Here, d represents the dimension of the embedding space. Within the model's framework, F and E signify the sets of potential failure modes (2) and their effects (3), respectively. Embedding processes for failure modes and effects are expressed as:

$$E_{\text{failure modes}} : f_i \mapsto \mathbf{v}_{\text{failure modes},i} \in \mathrm{R}^d, \tag{2}$$

$$E_{\text{failure effects}} : e_i \mapsto \mathbf{v}_{\text{failure effects},i} \in \mathrm{R}^d. \tag{3}$$

For example, considering "loosening of the bracket" as a failure mode f_i and "damage to the connection system" as a failure consequence e_i, the embedding process might be depicted as:

$$E_{\text{failure modes}} : \text{loosening of a bracket} \mapsto [0.85, -0.24],$$
$$E_{\text{failure effects}} : \text{damage to the connection system} \mapsto [0.65, -0.35].$$

Here, [0.85, –0.24] and [0.65, –0.35] are hypothetical 2D vectors representing the failure mode and effect,

respectively. The neural network model leverages these embeddings through mathematical operation like the dot product (4), defined as:

$$\text{Dot product} : \mathbf{v}_{\text{failure modes},i} \cdot \mathbf{v}_{\text{failure effects},j} = \sum_{k=1}^{d} (\mathbf{v}_{\text{failure modes},i})_k \cdot (\mathbf{v}_{\text{failure effects},j})_k. \quad (4)$$

Using the provided vectors, the dot product calculation would be:

$$\text{Dot product} : \mathbf{v}_{\text{failure modes}} \cdot \mathbf{v}_{\text{failure effect}} = 0.85 \times 0.65 + (-0.24) \times (-0.35),$$

where:

- $\mathbf{v}_{\text{failure modes}}$ is the vector for the loosening of the bracket,
- $\mathbf{v}_{\text{failure effects}}$ is the vector for the damage to the connection system,
- $\cdot$ represents the dot product between two vectors.

This dot product quantifies the relationship between the failure mode and consequence in the space. In this example, the result indicates the correlation level between these two aspects within the analyzed system. A higher dot product value suggests a strong linkage, which is crucial for identifying and prioritizing potential interventions in the system.

Entity embeddings' widespread use is attributed to the optimized advancement of neural networks, efficiently representing categorical variables as vectors and positioning similar categories in proximity. Unlike traditional encoding methods, entity embeddings utilize learning techniques to uncover relationships between similar entities.

Embedding model in neural network architecture

The neural network model that analyzes pairs of data representing failure modes and their effects is tasked with predicting whether a specific effect will occur for a given failure mode. This process includes several key stages:

1. Input layer: The neural network receives information about the failure mode and failure effects as separate inputs, which allows for independent data processing and increases the precision of the analysis.
2. Embedding layers: The raw input data are transformed into a more accessible representation for the model by mapping values onto vectors of a fixed length. This facilitates the detection of dependencies between data.
3. Dot product layer: The dot product between vectors from the embedding layers allows for the assessment of similarity between the mode and effect of failure, which is crucial in identifying connections between them.
4. Reshape layer: This layer adjusts the structure of the output data from the dot product layer to be compatible with the next stages of processing in the network.
5. Dense layer: The final dense layer, with a sigmoid activation function, generates the final predictions. The sigmoid function transforms the results into a range from 0 to 1, which is ideal in classification tasks where the outcome is binary.

Thanks to these stages, the model is capable of analyzing and extracting significant patterns from data related to failures, which translates to its predictive abilities.

Cosine similarity

In the context of modeling and analysis, cosine similarity is a measure of similarity between two vectors in a multidimensional space, occurring at the stage of calculating the dot product. In this context, it is utilized to assess the similarity between failure mode and effect of failure, providing a measure of correlation irrespective of the vectors' lengths.

Mathematically, the cosine similarity (5) $\cos\_\text{sim}$ between two vectors $e_{\text{failure modes}_i}$ and $e_{\text{failure effect}_i}$ is defined as the dot product of these vectors divided by the product of their lengths (Euclidean norms):

$$\cos\_\text{sim}\left(e_{\text{failure modes}_i}, e_{\text{failure effect}_i}\right) = \frac{e_{\text{failure modes}_i} \cdot e_{\text{failure effect}_i}}{||e_{\text{failure modes}_i}|| \cdot ||e_{\text{failure effect}_i}||}, \quad (5)$$

where:

- $e_{\text{failure modes}_i}$ is the embedding for the mode of failure,
- $e_{\text{failure effect}_i}$ is the embedding for the effect of failure,
- $\cdot$ denotes the dot product of two vectors,
- $||\cdot||$ denotes the Euclidean norm (length) of a vector.

Assuming two vectors representing the failure mode and the effect of failure in a 3-dimensional space: $e_{\text{loosening of a bracket}} = [1, 2, 3]$ and $e_{\text{damage to the connection system}} = [4, 5, 6]$, the cosine similarity of these vectors is calculated as follows:

$$\cos\_\text{sim}\begin{pmatrix} e_{\text{loosening of a bracket}}, \\ e_{\text{damage to the connection system}} \end{pmatrix} = \frac{(1 \cdot 3 + 2 \cdot 4)}{\sqrt{(1^2 + 2^2)} \cdot \sqrt{(3^2 + 4^2)}}.$$

Cosine similarity quantifies the degree of alignment between two nonzero vectors and is invariant to their magnitudes. Values near 1 indicate parallel orientation, values near 0 indicate orthogonality, and values near –1 indicate anti-parallel orientation. In our neural-network analysis, higher cosine similarity between the embeddings of a failure mode and a failure effect indicates a stronger semantic association and thus a higher recommendation score.

Training set

A training set for a neural network includes preparing data for supervised learning. Pairs (failure mode, effect of failure) are created, and the neural network uses them to learn to differentiate consistency with reality. The emphasis is on optimizing embeddings rather than maximizing prediction accuracy, foregoing separate validation and test sets.

To train a neural network, positive and negative samples are generated. Positive samples are chosen from an existing data set and labeled with a 1. Negative samples are generated randomly, avoiding existing pairs, and labeled with –1 or 0, depending on the nature of the task: –1 for regression problems and 0 for classification.

Authors use the label 0 for negative samples, indicating a classification approach that assigns samples to categories instead of predicting specific values.

Training the model

During the training of the model, the task is to perfect the embeddings, which are the representations of the properties of individual failure modes and their effects. The model sequentially adjusts its parameters; that is, it repeatedly modifies its weights based on the analysis of training data, to better predict whether a given failure mode leads to a specific effect or not.

To ensure the effectiveness of training, we need to properly adjust several key parameters. One of them is the batch size, which is the number of samples used for one update of the model's weights. The larger the batch size, the more effective the training, although it is necessary to remember the memory limitations of the machine. Moreover, it is crucial to precisely tune the negative sampling rate parameter, which influences the training process based on the observed results. This parameter defines the ratio of negative samples to positive samples in the training process. For the experiments, a value of 2 for this parameter achieved satisfactory results.

Another important aspect is determining the number of steps per epoch. The term "epoch" refers to one full pass through the entire training data set. In each epoch, the model processes the same number of training cases as the number of pairs in the data set. In this case, the authors chose 15 epochs, which is more than the strict minimum needed to achieve convergence of the model based on preliminary analyses. Thanks to these steps, we can train the model efficiently and effectively.

During the training of the model, we monitor the loss function, which indicates how well the model is doing in terms of prediction. A decreasing trend in the loss function suggests that the model is approaching an optimal state, capable of capturing the dependencies between failure modes and their effects. The model demonstrated progressive reduction in the loss function in subsequent epochs, indicating its ability to learn and adapt to the data.

The loss function values for the first and last epochs were: Epoch 1: 0.9735, ..., Epoch 15: 0.1570.

The decreasing trend in the loss function suggests that the model is approaching an optimal state, capable of capturing the dependencies between failure modes and their effects. The observed convergence and reduction in loss are promising indicators of the model's ability to learn and represent complex dependencies in the training data. However, further research, such as evaluation of metrics or visualization of results, can provide a more detailed understanding of the model's performance.

Results

The next step is to sort the results in order to identify entities that are close to each other in space. In the context of cosine similarity, higher numerical values indicate units that are in close proximity, where –1 denotes the furthest distance, and 1 indicates the closest.

In the context of the "Missing Part" failure mode inquiry, valuable insights were gained, especially considering the known association between the query and the specific failure outcome "Lack of Material" as a true assessment of similarity, marked with a perfect similarity score of 1.0.

Figure 2 presents the most and least similar items to the "Missing Part" according to the measure of cosine similarity. Green bars indicate items that are most similar, red bars indicate the least similar items.

1. Material is missing – (Similarity: 1.00): As expected, the result provides a perfect similarity score, confirming the correctness of the known association.

This confirms the effectiveness of the recommendation system in accurately identifying the real relationship between the "Missing Part" query and the "Lack of Material" outcome.

2. Mechanical function is missing – (Similarity: 0.79): With a positive similarity of 0.79, this result suggests a significant and meaningful association between the query and the "Lack of Mechanical Function" outcome. The positive result reinforces the system's ability to capture significant relationships.
3. Thread not calibrated – (Similarity: 0.68): A positive similarity of 0.68 indicates a moderate association between the "Missing Part" query and the "Thread Not Calibrated" outcome. This result adds valuable information to the understanding of subtle relationships in the system.
4. Material damaged – (Similarity: –0.78): A negative similarity score of –0.78 indicates diversity between the "Missing Part" query and the "Material Damaged" outcome. This contrasts with reality, suggesting a divergent association.
5. External leakage – (Similarity: –0.93): A highly negative similarity score of –0.93 highlights a clear diversity between the "Missing Part" query and the "External Leakage" outcome. This result aligns with expectations, indicating a strong inverse relationship.

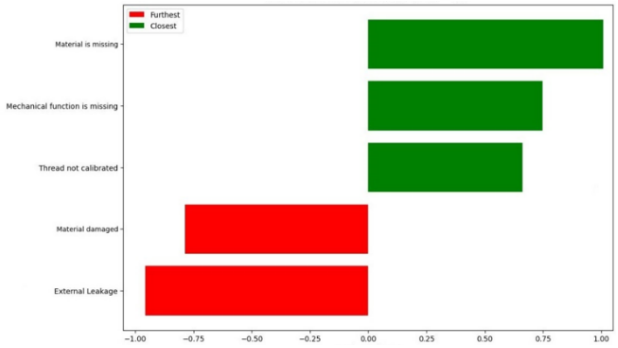

Fig. 2. Most and least similar items to "Missing part"

We evaluated the recommender with reproducible offline metrics and outlined our plan for an expert-based study. After cleaning and deduplication, the corpus contained 4208 failure mode-failure effect pairs. A project-wise split (90% train, 10% test) guaranteed that no failure mode from a given product family appeared in both sets. On the 392 test queries, we computed Precision@k($k = 1, 5, 10$), Mean Reciprocal Rank (MRR), and Mean Average Precision (MAP). The hybrid model achieved P@5 = 0.79, MRR = 0.73 and MAP = 0.68, outperforming the collaborative-only baseline (P@5 = 0.71) and the content-only baseline (P@5 = 0.62). Detailed results appear in Table 1.

Because the present validation is dataset-based, it cannot capture subjective factors such as engineer confidence or workshop dynamics. The forthcoming expert study is intended to close that gap.

Table 1 summarises Precision@5 for the collaborative, content-only, and hybrid variants described in the integrating co-occurrence signals with linguistic semantics section.

Table 1
Precision@5 results

Model variant	Retrieval signal(s) used	Precision@5
Collaborative filtering (CF-only)	Dual FM/FE embeddings, cosine similarity	0.71
Content-based (Sentence-BERT only)	Sentence-level semantic similarity	0.62
Hybrid (weighted CF + CB)	0.65 · CF + 0.35 · CB (empirically tuned)	0.79

Limitations of the proposed solution

On one hand, the system has many advantages. It possesses a deep semantic understanding, going beyond simple keyword matching to provide recommendations based on similarity metrics. Moreover, equipped with entity embeddings in neural networks, it quickly processes vast datasets, delivering rapid and accurate recommendations. The system's flexibility allows it to adapt to various datasets and applications within the DFMEA process, while its inference-generating capabilities offer valuable input for comprehensive risk assessment and decision-making.

However, there are challenges. The system's effectiveness largely depends on the quality and completeness of input data, which can lead to inaccuracies in the case of insufficient or biased data. Additionally, its implementation and refinement may require specialized knowledge in machine learning, which can be a barrier for users without such expertise. Interpretability can also be an issue; although the system provides quantitative similarity measures, understanding the reasons behind specific recommendations can be difficult, impacting transparency.

To ease these shortcomings, the current prototype incorporates two practical safeguards. First, a concise "Why-card" is displayed with every recommended effect. It lists the three most similar failure mode-failure effect pairs found in the training set and shows the proportion of the final score attributable to collaborative versus semantic similarity, giving engineers an immediate rationale for the suggestion. Second, a two-stage

data-quality gate precedes every nightly retrain. Automatic checks enforce the DFMEA schema (mandatory columns, S–O–D within 1–10, no duplicate pairs), and a short domain review flags free-text edits and maps obsolete terms to the current vocabulary. Only rows that pass both layers feed the model, reducing the risk of propagating noise or bias.

Potential development in the future

In the future, the method can be expanded by adding further elements of analysis, resulting in recommendations for each criterion of the table by providing an initial list of product components we want to analyze. Integration with real-time monitoring systems and feedback loops could enable continuous learning, ensuring the system's adaptation to evolving failure patterns. Advanced visualization techniques, such as interactive embedding visualization and graphical representations, promise to improve interpretability and usability. Combining the recommendation system with expert knowledge bases and decision support systems can enrich the analysis process, leveraging both computational capabilities and human knowledge.

Extending the application beyond DFMEA to other areas, such as quality control, predictive maintenance, and risk management in various industries, represents a promising direction for further research.

Although the recommender streamlines routine failure mode to failure effect assignments, it can overlook rare or previously unseen patterns. Recent work on deep-learning anomaly detection in manufacturing processes by Salam *et al.* (2024) demonstrates that unsupervised models can flag unusual sensor or process signatures with high precision. Coupling such detectors with our embedding space would enable the system to raise a "rare-event" alert whenever a new pair lies far outside historical clusters, and feed the flagged cases back into training, gradually enriching the model with edge examples. Designing and validating this feedback loop is therefore identified as a promising avenue for future research.

The version evaluated in the project focuses on generating relevant Failure-Effect (FE) suggestions for a given Failure Mode (FM). Prioritisation is still performed manually during the DFMEA workshop. Engineers still assign Risk-Priority Numbers (RPN) manually. To close this gap, we can propose a prototype RPN-based module that will:

1. retrieve the median Severity (S) previously logged for each recommended FE;
2. estimate Occurrence (O) from the relative frequency of the associated FM in the training corpus;
3. read Detection (D) from the current control plan (default = 10 if none is defined);

And then sort the list by RPN = S × O × D with a traffic-light colour code. Because these data fields already exist in the DFMEA tables used for training, the enhancement can be implemented without altering the recommender's core architecture.

Scalability with growing design complexity and data volume

The architecture is based on two 50-dimensional embedding matrices, one parameterizing failure modes (FM) and the other failure effects (FE). Consequently, the trainable parameter count increases linearly with the number of distinct FM and FE phrases and does not depend on the number of DFMEA sheets or the granularity of any single design.

Because FM and FE sentences remain the atomic modelling units, adding new product lines or more detailed DFMEA attributes only appends vectors, leaving the network architecture, dimensionality, and retrieval code unchanged.

Because DFMEA sheets are usually protected by non-disclosure agreements, enlarging the training set across multiple suppliers calls for techniques that never expose raw records. A recent study by Salam et al. (2023) presents a multi-party privacy-preserving machine-learning framework that exchanges only encrypted model updates while achieving near centralised accuracy. The same idea can be applied to our recommender: each company would fine-tune the FM- and FE-embedding tables on its local servers, send gradient deltas to a trusted aggregator, and receive the updated global model. No single party ever sees another's failure logs, yet the shared vector space benefits from the combined data volume. Adopting such a federated or "secure collaborative-learning" loop is therefore the logical next step for scaling the system beyond a single organisation.

In summary, although the recommendation system represents significant progress in DFMEA analysis, continued research and development are essential to overcome limitations and fully utilize its potential in enhancing the reliability and safety of products across all industries.

Discussion

Recommendations can assist in identifying effective preventative actions through the analysis of actions taken in the past in similar situations and suggest them during the initial analysis performed by the team in the

company. The central knowledge repository offered by the system allows the team easy access to relevant information, analysis of previous cases, conclusions, and recommendations. This facilitates the exchange of knowledge within the team and supports decision-making.

Furthermore, the system can provide tools for data visualization, which facilitates the interpretation of risk analysis results and the presentation of this information in a way that is understandable to all team members.

The system can significantly facilitate the DFMEA process by supporting the team in identification, analysis, and risk management associated with the project. However, the effectiveness of this solution will depend on the quality of input data and the team's ability to interpret the results.

In this case, the authors used anonymised synthetic data containing typical nomenclature for products in the energy sector to train the machine learning model.

The model itself, based on failure modes provided by the user, will recommend a list of their effects, accelerating the initial process of creating an FMEA table by the facilitator. This can save time, as all necessary information is available in one place, eliminating the need to search through various sources and gather data manually.

The difference between the presented solution and others available on the market is the deep integration of machine learning with the DFMEA process. While other systems, as described by Ouyang et al. (2022) and Mueller et al. (2019), may rely more on actual operational data, the solution also demonstrates effectiveness using anonymised synthetic data to train the model, which can be particularly valuable for new products where failure history is limited or unavailable. The presented approach is closest to the solutions proposed by Ouyang et al. (2022) in terms of multidimensional risk analysis, but it extends them by applying more advanced techniques of entity embedding in neural networks and data visualization, plus the hybrid layer, which mitigates the cold-start weakness. The semantic fallback preserves usability for previously unseen failure modes while incurring only a modest precision drop.

When the recommender is eventually embedded in the company's CAD/PLM environment, two bias risks become especially relevant.

Historical DFMEA archives are weighted toward mature, high-volume products, while previously unseen or low-volume platforms appear only sporadically. If these records are used unchanged, the system will keep suggesting the "usual suspects" drawn from legacy projects and overlook effects that matter for next-generation designs. During training, we therefore apply inverse-frequency weighting until every product family supplies at least two per cent of each mini-batch. At evaluation time, the main metrics are broken out by family so that any lingering shortfall remains visible to engineers and data scientists alike.

In the early deployment phase, only a few business units will enable the plug-in, meaning their usage logs could overwhelm those from later adopters. If that imbalance makes its way into re-training, the model will end up speaking the terminology of one division and perform suboptimally in others. Every feedback event is therefore stamped with site and division identifiers, and the batch scheduler adjusts sampling so that each division influences the weight update in proportion to its active user base.

The uniqueness of the solution lies in the integration of machine learning with the DFMEA process, which not only accelerates the process of identifying potential failures but also increases the objectivity and accuracy of analyses. Unlike other methods, which often rely on the subjective assessment of experts, the created model uses data to generate objective and repeatable results, significantly increasing their credibility.

Conclusions

The recommendation system developed for identifying potential consequences of failures in the Failure Mode and Effects Analysis (DFMEA) process represents a significant step forward in the use of advanced technology to improve the reliability and safety of products. By utilizing entity embeddings in neural networks, the system offers an advanced understanding of the relationships between failure modes and their consequences, facilitating informed decision-making in the assessment of risk and strategies for its mitigation. Through systematic training and the use of cosine similarity metrics, this system provides valuable insights into the relationships between failure modes, paving the way for more robust and reliable products.

Although the recommendation system has achieved its intended goal in the DFMEA analysis, continuous improvement and testing are necessary to ensure its reliability and effectiveness in various industrial applications. Further development of user interfaces and visualization features will also be crucial to allow engineers and analysts to use the system more easily and intuitively.

In the future, the system could be integrated with more data sources and utilize more advanced machine learning algorithms to further enhance its effectiveness and range of applications. Striving for continuous improvement of the recommendation system will be key to maintaining its competitiveness and value in the context of rapidly advancing technology and increasing demands.

References

Bengio, Y., Ducharme, R., & Vincent, P. (2000). A Neural Probabilistic Language Model. *Journal of Machine Learning Research 3(6): 1*, 932–938. DOI: 10.1162/153244303322533223

Chang, D., & Sun, P. K. (2009). Applying DEA to enhance assessment capability of FMEA. *International Journal of Quality & Reliability Management*, *26*(6), 629–643. DOI: 10.1108/02656710910966165

Daramola, O. J., Stålhane, T., Omoronyia, I. & Sindre, G. (2013). Using Ontologies and Machine Learning for Hazard Identification and Safety Analysis. *Springer EBooks*, 117–141. DOI: 10.1007/978-3-642-34419-0_6

Dev, K., Gurukula, S., Vishwavidyalaya, K., Srivastava, S., & Kangri Vishwavidyalaya, G. (2018). *Failure Mode and Effect Analysis (FMEA) Implementation: A Literature Review.*

En-Naaoui, A., Gallab, M., & Kaicer, M. (2023). Intelligent Model for Improving Risk Assessment in Sterilization Units Using Revised FMEA, Fuzzy Inference, k-Nearest Neighbors and Support Vector Machine. *Journal of Applied Research and Technology (En Línea)*, *21*(5), 772–786. DOI: 10.22201/icat.24486736e.2023.21.5.2116

Filz, M.-A., Langner, J.E.B., Herrmann, C., & Thiede, S. (2021). Data-driven failure mode and effect analysis (FMEA) to enhance maintenance planning. *Computers in Industry*, *129*, 103451. DOI: 10.1016/j.compind.2021.103451

Garner, S. (1995). WEKA: The Waikato Environment for Knowledge Analysis. Proceedings of the New Zealand Computer Science Research Students Conference.

Hassan, F., Nguyen, T.T., Le, T., & Le, C. (2023). Automated prioritization of construction project requirements using machine learning and fuzzy Failure Mode and Effects Analysis (FMEA). *Automation in Construction*, *154*, 105013–105013. DOI: 10.1016/j.autcon.2023.105013

Jomthanachai, S., Wong, W.-P., & Lim, C.-P. (2021). An Application of Data Envelopment Analysis and Machine Learning Approach to Risk Management. *IEEE Access*, *9*, 85978–85994. DOI: 10.1109/ACCESS.2021.3087623

Linda, A., & Sahayam, A. (2023). *FMEA: DFMEA vs. FMECA.*

Mangeli, M., Shahraki, A. & Saljooghi, F.H. (2019). Improvement of risk assessment in the FMEA using nonlinear model, revised fuzzy TOPSIS, and support vector machine. *International Journal of Industrial Ergonomics*, *69*, 209–216. DOI: 10.1016/j.ergon.2018.11.004

Mueller, T., Kiesel, R., & Schmitt, R.H. (2019). *Automated and Predictive Risk Assessment in Modern Manufacturing Based on Machine Learning* (R. Schmitt & G. Schuh, Eds.).

Ouyang, L., Che, Y., Yan, L., & Park, C. (2022). Multiple perspectives on analyzing risk factors in FMEA. *Computers in Industry*, *141*, 103712. DOI: 10.1016/j.compind.2022.103712

Park, J.-H., Seok, J.-H., Cheon, K.-M., & Hur, J.-W. (2020). Machine Learning Based Failure Prognostics of Aluminum Electrolytic Capacitors. *Journal of the Korean Society of Manufacturing Process Engineers*, *19*(11), 94–101. DOI: 10.14775/ksmpe.2020.19.11.094

Sader, S., Husti, I., & Daróczi, M. (2020). Enhancing Failure Mode and Effects Analysis Using Auto Machine Learning: A Case Study of the Agricultural Machinery Industry. *Processes*, *8*(2), 224. DOI: 10.3390/pr8020224

Salam A., Abrar M., Ullah F., et al. (2023). Efficient Data Collaboration Using Multi-Party Privacy-Preserving Machine Learning Framework. *IEEE Access, 11*, 138151–138164. DOI: 10.1109/ACCESS.2023.3339750

Salam, M., Chen, X., & Rossi, T. (2024). Unsupervised Deep Anomaly Detection for Smart-Manufacturing Streams. *IEEE Access, 12*, 3373697. DOI: 10.1109/ACCESS.2024.3373697

Sellappan, N., Deivanayagampillai, N., & Palanikumar, K. (2013). Evaluation of risk priority number (RPN) in design failure modes and effects analysis (DFMEA) using factor analysis Modified Prioritization Methodology for Risk Priority Number in Failure Mode and Effects Analysis. *International Journal of Applied Science and Technology*, *3*(4).

Song, W., & Zheng, J. (2024). A new approach to risk assessment in failure mode and effect analysis based on engineering textual data. *Quality Engineering*, 1–19. DOI: 10.1080/08982112.2024.2304815

Wang, Z., Du, H., Tao, L. and Javed, S.A. (2024). Risk assessment in machine learning enhanced failure mode and effects analysis, *Data Technologies and Applications*, *58(1)*, 95–112. DOI: 10.1108/DTA-06-2022-0232

Wu, Z., Liu, W., & Nie, W. (2021). Literature review and prospect of the development and application of FMEA in manufacturing industry. *The International Journal of Advanced Manufacturing Technology*, *112*(5-6), 1409–1436. DOI: 10.1007/s00170-020-06425-0

Yang, C., Zou, Y., Pinhua, L., & Jiang, N. (2015). Data mining-based methods for fault isolation with validated FMEA model ranking. *Applied Intelligence*, *43*(4), 913–923. DOI: 10.1007/s10489-015-0674-x