Evaluating the deep learning software tools for large-scale enterprises using a novel TODIFFA-MCDM framework

Zoran Gligorić, Ömer Faruk Görçün, Miloš Gligorić, Dragan Pamucar, Vladimir Simić, Hande Küçükönder

Evaluating the deep learning software tools for large-scale enterprises using a novel TODIFFA-MCDM framework | Zoran Gligorić, Ömer Faruk Görçün, Miloš Gligorić, Dragan Pamucar, Vladimir Simić, Hande Küçükönder | Journal of King Saud University - Computer and Information Sciences | 2024 | |

10.1016/j.jksuci.2024.102079

http://dr.rgf.bg.ac.rs/s/repo/item/0009281

Journal of King Saud University - Computer and Information Sciences 36 (2024) 102079

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Journal of King Saud University - Computer and Information Sciences

journal homepage: www.sciencedirect.com

Full Length Article

Evaluating the deep learning software tools for large-scale enterprises using a novel TODIFFA-MCDM framework

Zoran Gligorić [a], Ömer Faruk Görçün [b], Miloš Gligorić [a,*], Dragan Pamucar [c], Vladimir Simic [d], Hande Küçükönder [e]

[a] University of Belgrade, Faculty of Mining and Geology, Belgrade 11000, Serbia
[b] Department of Business Administration at Kadir Has University, Cibali Av. Kadir Has St. Fatih, Istanbul 34083, Turkey
[c] University of Belgrade, Faculty of Organizational Sciences, Department of Operations Research and Statistics, Jove Ilića 154, 11000 Belgrade, Serbia
[d] University of Belgrade, Faculty of Transport and Traffic Engineering, Vojvode Stepe 305, 11010 Belgrade, Serbia
[e] Department of Numerical Methods, Faculty of Economics and Administrative Sciences, Bartın University, Bartın, Turkey

ARTICLE INFO

Keywords:
Deep learning (DL)
Large-scale industries
MAXC (MAXimum of Criterion)
TODIFFA (the total differential of alternative)
DL software selection

ABSTRACT

Deep learning (DL) is one of the most promising technological developments emerging in the fourth industrial revolution era for businesses to improve processes, increase efficiency, and reduce errors. Accordingly, hierarchical learning software selection is one of the most critical decision-making problems in integrating neural network applications into business models. However, selecting appropriate reinforcement learning software for integrating deep learning applications into enterprises' business models takes much work for decision-makers. There are several reasons for this: first, practitioners' limited knowledge and experience of DL makes it difficult for decision-makers to adapt this technology into their enterprises' business model and significantly increases complex uncertainties. Secondly, according to the authors' knowledge, no study in the literature addresses deep structured learning solutions with the help of MCDM approaches. Consequently, making inferences concerning criteria that should be considered in an evaluation process is impossible by considering the studies in the relevant literature. Considering these gaps, this study presents a novel decision-making approach developed by the authors. It involves the combination of two new decision-making approaches, MAXC (MAXimum of Criterion) and TODIFFA (the total differential of alternative), which were developed to solve current decision-making problems. When the most important advantages of this model are considered, it associates objective and subjective approaches and eliminates some critical limitations of these methodologies. Besides, it has an easily followable algorithm without the need for advanced mathematical knowledge for practitioners and provides highly stable and reliable results in solving complex decision-making problems. Another novelty of the study is that the criteria are determined with a long-term negotiation process that is part of comprehensive fieldwork with specialists. When the conclusions obtained using this model are briefly reviewed, the C2 "Data Availability and Quality" criterion is the most influential in selecting deep learning software. The C7 "Time Constraints" criterion follows the most influential factor. Remarkably, prior research has overlooked the correlation between the performance of Deep Learning (DL) platforms and the quality and accessibility of data. The findings of this study underscore the necessity for DL platform developers to devise solutions to enable DL platforms to operate effectively, notwithstanding the availability of clean, high-quality, and adequate data. Finally, the robustness check carried out to test the validity of the proposed model confirms the accuracy and robustness of the results obtained by implementing the suggested model.

1. Introduction

Artificial intelligence is associated with machine learning by imitating human behaviours and reasoning processes so that machines, equipment and devices think like humans and are trained to produce the most appropriate solutions to the situation and conditions. In this

* Corresponding author.
E-mail addresses: zoran.gligoric@rgf.bg.ac.rs (Z. Gligorić), omer.gorcun@khas.edu.tr (Ö.F. Görçün), milos.gligoric@rgf.bg.ac.rs (M. Gligorić), dragan.pamucar@fon.bg.ac.rs (D. Pamucar), vsima@sf.bg.ac.rs (V. Simic), hkucukonder@bartin.edu.tr (H. Küçükönder).

https://doi.org/10.1016/j.jksuci.2024.102079
Received 30 January 2024; Received in revised form 14 May 2024; Accepted 27 May 2024
Available online 31 May 2024

context, the development of learning processes by imitating human behaviour and intelligence is called artificial intelligence. In this context, McCulloch and Pitts (1943) modelled the human nervous system, which can be defined as the basis of Artificial Neural Networks (ANNs). However, due to many limitations and inadequacies, artificial intelligence and machine learning have not been sufficiently accepted in industrial applications. First, machine learning algorithms cannot learn by processing large-scale and complex data (Hatcher and Yu, 2018). In addition, due to the insufficient hardware qualities of artificial neural networks (ANNs), artificial intelligence applications have not become widespread in industries (Şeker et al., 2017). One of the leading hardware inadequacies was using Central Processing Units (CPUs) to process data by artificial intelligence applications. CPU applications were typically missing (Shi et al., 2017) because CPUs processed data exceptionally slowly. At the same time, CPUs were not scaling well enough for learning applications to cause problems.

To address these limitations, deep learning (DL) has led to a revolution in the field of artificial intelligence, which has resulted from technological developments and advances in recent years. Big Data is one of the most essential technological instruments enabling deep learning applications to develop and spread. In this context, Big Data has made larger-scale data required by deep learning applications accessible. In addition, Internet of Things (IoT) technologies, which make it possible to transfer data to Big Data platforms and generate data on a massive scale from connected devices, also significantly contribute to this process. Besides, Graphic Processing Unit GPUs, which have replaced CPUs in data processing applications in recent years, can process data thirty times faster than previous technologies (Kabakuş, 2020). Şeker et al. (2017) claimed that computation speed has increased a thousandfold compared to the past decade due to the use of GPUs. Because GPUs enable massively parallel computing to train more extensive and deeper models, (Hernández-Blanco et al., 2019) it has been possible to efficiently train networks with billions of parameters and variables (Bahrampour et al., 2016). That can be considered as one of the reasons why deep learning applications are successful compared to previous artificial intelligence applications.

1.1. The development of deep learning technology

Advances in Deep Learning (DL), a subset of machine learning, have propelled it as a leading trend in the field (Al-Bdour et al., 2020). One of the reasons for this is that DL has an incomparable ability in image, video, and sound processing compared to past technologies. DL architectures flexibly handle various data types (e.g., visual, audio, text) for processing, offering versatility. Accordingly, DL has been rapidly adopted by research community members and industry practitioners. The developments mentioned above have made professionals in various industries and researchers more interested in DL applications. The advantages and capabilities of this synergy and approach created by DL have enabled the development of intelligent systems in many industries and fields.

As a result, DL applications have been used to develop intelligent systems and technologies in many fields and industries. Over the last few years, there has been a steady growth in the demands of sectors related to DL applications. Today, it has become possible to frequently see DL applications in areas such as image, audio and video processing, detection, and recognition (Hatcher and Yu, 2018). In addition, the range of uses of DL technology is extensive. For example, DL applications are vital in developing intelligent transportation systems and autonomous vehicle technology. In addition, it can provide adequate responses to requirements in the healthcare industry, such as making highly accurate diagnoses for deadly diseases, monitoring people's health status with wearable devices, and designing and developing personalized medicines considering each patient's specific circumstances. Another use of DL applications is robotic systems in automotive, mining, and other industries. Thanks to DL, robotics will perform better and perform highly complex tasks autonomously with high accuracy. In addition, DL applications for the finance field are becoming widespread. DL technologies can be used to develop intelligent systems so that decision-makers can make investment decisions at a more optimal level. In addition, DL applications have had the chance to be used in many areas, such as virtual reality, computer game design, prediction of election results, biology, and education.

Tech giants like Google, Microsoft, Facebook, Amazon, and Apple invest in deep learning to leverage its potential for creating intelligent products, acknowledging its growing trend (Şeker et al., 2017; Hernández-Blanco et al., 2019). In our study, we found 19 DL software options designed for industrial use, a result of initiatives by tech firms and academic institutions in DL technology. In contrast, few DL software was available that were evaluated by studies in the literature. In the literature, DL platforms such as TensorFlow, Theano, Keras, CNTK, Caffe, Torch, Neon, PyTorch, DeepLearning4J, deepmat, Eblearn, MXNet (Shi et al., 2017; Kabakuş, 2020; Pham et al., 2020) have been evaluated by researchers in different studies. In this context, it can be concluded that the DL market is constantly expanding and that the actors producing new DL software are included.

1.2. Challenges in choosing deep learning technology

Choosing the DL platform that can provide the highest efficiency and effectiveness for industries is becoming increasingly difficult for decision-makers (Ulker et al., 2020). The variety of available software keeps expanding, with DL platforms evolving to offer increasingly personalized and specialized structures and features, driving this trend. When the same DL algorithm is used for different industries and situations, the results and performance of the DL application may vary (Al-Bdour et al., 2020). DL software optimized for drug development in computational biology may not perform effectively in designing collision avoidance systems for autonomous cars due to differences in computational speed, perception, and control requirements. It further complicates the decision-making regarding DL selection. However, selecting the appropriate DL framework and library is critical for optimal performance and accuracy in any application (Rao, 2023). Each DL platform alternative has different features, advantages, disadvantages, and capabilities, making it more difficult for decision-makers.

Further complicating the decision problem is the lack of evidence from previous studies regarding the suitability of the criteria influencing the choice of the DL platform. The absence of a consensus on criteria for DL platform selection among authors in the literature may lead to challenges in structuring decision-making and unreliable outputs. However, because of a comprehensive literature review, according to our observations, no study has evaluated DL platforms using multi-criteria decision-making approaches. It deprives industry professionals who want to develop intelligent systems using DL architectures of the support of the research community. Finally, DL algorithms and architectures became popular and developed after 2006 (Şeker et al., 2017; Hernández-Blanco et al., 2019; Kaya et al., 2019; Sherkhane and Vora, 2017). So, DL technology is exceptionally new and may have capabilities and features that industries have not adequately explored. In this respect, the limited knowledge and experience of decision-makers on DL applications makes the solution to this decision-making problem even more difficult.

Considering these critical research gaps, we have extensively examined DL technologies and architectures to reduce the knowledge gaps of decision-makers and end users in different industries. We aim to offer decision-makers comprehensive details on selection criteria, alternative DL software, and platform performance and features for their evaluation processes. The study aims to propose a strong, effective, and robust decision-making model for evaluating and selecting DL platforms. The proposed model makes it easier for decision-makers to make more optimal decisions in a selection process and dramatically shortens the evaluation process. Apart from this, the information about the proposed model and its structural qualities and advantages is as follows.

1.3. The developed solution to address the problem

The method presented in this paper is recommended to be used as an objective method for determining criteria weights. We have devised MAXC (MAXimum of Criterion), a novel objective method, determining criterion weight based on the expected distance from each criterion value to its maximum. If we consider that the weights of the criteria significantly influence the final rank of alternatives, then objectivity is an essential part of the decision-making process. Targets of the criteria (max or min) have no impact on the weight calculation, which means that any normalization of input data can be applied. The calculation procedure is simple and understandable. Besides the MAXC method, we used the ordinal priority model linear model for subjective determining criteria weights. Finally, we proposed an aggregation function to fuse subjective and objective weights to define optimal values.

The motivation and intuition behind the TODIFFA (TOtal DIFFerential of Alternative) method is based on calculating the change of a hypotenuse function, composed of cost and benefit variables, in a direction from evaluating an alternative to an optimal alternative. The change smaller the alternative, the better. The calculation needed for alternative ranking is simple and easy to understand. Comparison results show that the TODIFFA method highly correlates with MABAC, TOPSIS, WASPAS and COPRAS techniques. That indicates that the TODIFFA method is stable and reliable for solving multi-criteria decision-making problems.

1.4. Structuring the rest of the paper

The rest of the article is structured as follows. Section 2 presents the results of a comprehensive literature review. The paper concentrates on studies comparing and evaluating DL platforms, noting their contributions and unaddressed gaps, diverging from technical feature analyses of DL technology. In addition, the parameters considered in the comparison and the DL platform alternatives compared were determined. Section 3 shows the basic algorithm and procedure of the proposed model while it comprehensively shows the steps and mathematical operations for implementing the model. The model proposed in Section 4 is applied to evaluating DL platforms and selecting the most appropriate one. Section 5 summarizes the results obtained and discusses the practical implications of these results and outcomes. In Section 6, while concluding the study, the study's limitations and recommendations for future studies are noted.

2. Literature survey

Upon conducting a literature review to explore research about the utilization and integration of deep learning models and applications within the automotive industry, we identified 295 relevant studies. However, not all of these studies directly addressed the application of deep learning models in the automotive sector. Expanding our investigation to various industries, we discovered 57 studies related to the health industry, 79 on the energy sector, 54 on the logistics industry, and 240 on the food industry.

When the studies that focus directly on deep learning approaches in the automotive industry are evaluated, Werda et al. (2022) undertook an experimental investigation to automate dataset creation and annotations for programming robots used in welding applications within the automotive industry. Additionally, Luckow et al. (2016) have conducted several studies focusing on libraries, tools, and infrastructures, including GPUs and cloud computing, to explore recent advancements in this field.

Theissler et al. (2021) investigated potential use cases and obstacles for implementing predictive maintenance in the automotive sector through deep learning applications. Jhung and Kim (2021) endeavoured to enhance driver assistance systems in challenging rainy conditions by leveraging deep learning methodologies. Schmiedt et al. (2022) explored the enhancement of driving comfort in dual-clutch vehicles by applying deep-learning models. Meanwhile, Junaid et al. (2021) evaluated the pedestrian detection capabilities of self-driving cars by employing deep learning techniques for image processing.

Šabanovič et al. (2021) examined the feasibility of employing a neural network-based virtual sensor to estimate the relative velocity of vehicles. Um et al. (2019) endeavoured to forecast the energy consumption of remotely operated welding robots using deep learning methodologies. Abboush et al. (2022) conducted an empirical investigation evaluating automotive software systems for intelligent fault detection and classification, employing hybrid deep learning techniques. Kejun et al. (2018) deliberated on the functionalities and ramifications of deep learning models for autonomous vehicles. Roh and Lee (2023) compared augmented reality applications utilizing deep learning methodologies. Chinta et al. (2023) explored optimal feature selection within deep learning frameworks for predictive maintenance systems in the automotive sector. Espinosa et al. (2021) developed an algorithm for detecting and recognizing sounds employing deep learning models.

In broad terms, research regarding using deep learning models within the automotive sector has predominantly concentrated on enhancing driving algorithms for autonomous vehicles, implementing image and voice recognition technologies, and refining robotics programming within manufacturing systems. However, there is a noticeable absence of studies exploring the selection of appropriate deep learning platforms in the existing literature, except for the investigation undertaken by Luckow et al. (2016), which compared various deep learning platforms tailored for the automotive industry. Nevertheless, the study by Luckow et al. (2016) did not present a formal mathematical model or decision support system; instead, it conducted an empirical assessment to evaluate the performance of these platforms.

When the literature on the evaluation and selection of DL platforms was examined, non-numerical methods such as surveys, observations and examinations were mainly used in the preceding studies, and the researchers compared DL platform alternatives by considering specific criteria. According to the authors' information, none of these studies evaluated DL platforms using multi-criteria decision-making tools. One of the reasons for this may be that DL technologies are an extremely new field, and practitioners and members of the research community have limited knowledge of the subject. For this reason, the prior investigations that will be considered pioneering studies have tried to provide an overview of DL technology and understand the subject's depth. More importantly, DL is an ever-expanding field undergoing rapid development due to the development of technology. Accordingly, the information and inferences about DL applications can quickly become outdated. That may be one of the reasons why researchers and authors have not proposed a comprehensive mathematical model for comparing DL platforms. Table 1, created to obtain information about the studies in the literature and to show this information, presents the results of a detailed examination of these studies.

Considering the information presented in Table 1, while numerous studies on DL are in the literature, the number of studies evaluating DL platforms is surprisingly scarce. Despite a comprehensive literature review, we have only accessed ten studies dealing with DL platform comparisons. However, none of these studies proposed multi-criteria decision-making approaches, decision support systems or an optimization model, and they compared DL software within the framework of some criteria. The most focused evaluation criteria were noted as performance and speed. The authors considered the speed of data processing to be a determining factor. Except for Druzhkov and Kustikova (2016), the number of alternatives in other studies ranges from 5 to 8. Coffe is a DL platform alternative that has been evaluated in every study. In addition, DL software alternatives such as Theano (8), TensorFlow (7), Torch (6), CNTK (5), Keras (5), MXNet (3), PyTorch (3), Deep-Learning4J (2), Pylearn (2), Cuda-convnet (2) were considered in these studies. The remaining alternatives were evaluated one at a time. There

Table 1
Detailed information on the previous studies comparing the DL software.

No	Author(s) year	Methodology	Number of			Results: Most important	
			Uncertainties	Criteria	Options	Criteria	Alternatives
1	Hatcher & Yu (Hatcher and Yu, 2018)	Survey	No	4	8	Secure Deep Learning	–
2	Al-Bdour et al. (Al-Bdour et al., 2020)	Performance comparison	No	3	6	Running time	CNTK
3	Bahrampour et al. (Bahrampour et al., 2016)	Performance comparison	No	2	5	Gradient computation time	Theano
4	Kabakuş (Kabakuş, 2020)	Experimental analysis	No	3	5	Training time	Keras
5	Druzhkov & Kustikova (Druzhkov and Kustikova, 2016)	Survey	No	5	17	–	Torch
6	Rao (Rao, 2023)	Comparative analysis	No	5	5	Performance	TensorFlow
7	Şeker et al. (Şeker et al., 2017)	Survey	No	1	6	Running time	TensorFlow
8	Sherkhane & Vora (Sherkhane and Vora, 2017)	Survey	No	10	5	Performance	Torch
9	Shi et al. (Shi et al., 2017)	Experimental analysis	No	1	5	Training speed	CNTK
10	Yapıcı & Topaloğlu (Yapıcı and Topaloğlu, 2021)	Comparative analysis	No	2	7	Batch Time	TensorFlow

The studies' number (No)																				
Criteria	1		2		3		4		5		6		7		8		9		10		*Alternatives*
	C	O	C	O	C	O	C	O	C	O	C	O	C	O	C	O	C	O	C	O	
Acceleration and Optimization	✓	☺		☺		☺						☺		☺				☺		☺	TensorFlow
Distributed DL in IoT and CPS	✓	☺												☺							DeepLearning4J
Network Management and Control	✓	☺		☺		☺		☺		☺				☺		☺				☺	Theano
Secure DL	✓	☺																			Options
Running time		☺	✓	☺				☺					✓					☺		☺	CNTK
Memory consumption		☺	✓	☺		☺		☺		☺		☺		☺		☺		☺		☺	Coffee
CPU and GPU utilization		☺	✓									☺			✓					☺	MXNet
Gradient Computation Time		☺			✓	☺		☺	✓			☺								☺	Keras
Supported languages				☺			✓														Neon
Training time				☺			✓			☺				☺		☺		☺		☺	Torch
Testing time						☺	✓	☺				☺									PyTorch
Autoencoder performance									✓	☺											DeepLearnToolbox
Image classification ability									✓	☺						☺					Pylearn
Sparse coding									✓	☺											Deepnet
Deep learning models									✓	☺											Deepmat
Performance										☺	✓				✓						Darch
Ease of use										☺	✓										nnForge
Documentation										☺	✓										CXXNET
Community support										☺	✓					☺					Cuda-convnet
Extensibility										☺					✓						Cuda CNN
Hardware utilization										☺					✓						EBLearn
Ecosystem										☺					✓						Hebel
Architecture of tools										☺					✓						Crino
Cross-Platform										☺					✓						Lush

(*continued on next page*)

Table 1 (*continued*)

The studies' number (No)																					
Criteria	1		2		3		4		5		6		7		8		9		10		*Alternatives*
	C	O	C	O	C	O	C	O	C	O	C	O	C	O	C	O	C	O	C	O	
Open source										☺					✓						R-CNN
Interface														☺	✓						KNET
Modelling capability															✓						
Training speed																	✓				
Batch Time																			✓		
Epoch Time																			✓		

C: Criteria ✓, O: Alternatives ☺.

is no consensus in the literature on which DL tool is more straightforward and better. TensorFlow was identified as the highest-performing alternative in three of the ten studies, while CNTK and Torch were shown as the optimal option in two studies. Theano and Keras have been pointed out as the best alternative in one study.

The absence of literature evaluating the selection of deep learning technologies through multi-criteria decision-making approaches significantly deprives decision-makers of support from the research community. Consequently, practitioners and decision-makers across various industries lack insights from the research community on deep learning platforms, adversely affecting the accuracy and optimality of their decisions. Additionally, the absence of insights may make decision-makers hesitate to integrate deep learning technologies into their business models. It underscores the importance of studies analyzing the selection of deep learning technologies using multi-criteria decision-making approaches. Such research would not only provide valuable insights to decision-makers but also contribute to advancing knowledge in the field of deep learning technology integration. Therefore, there is a critical need for more scholarly investigations to bridge the gap between research and practice and facilitate informed decision-making in adopting deep learning technologies.

2.1. Research and industrial practices gaps

In this study, the most critical research gap identified as a result of the literature review is that there is no decision support system or decision-making tool in the literature that can be used to evaluate DL platform alternatives and select the most appropriate alternative. Researchers have generally compared alternatives by designing investigation, observation, and experimental processes. Secondly, no evidence has been put forward by the authors of this study on the importance and effects of the criteria and factors considered in the evaluation process. At the same time, it is not clear enough how these criteria are determined. Another critical research gap is that it is unclear for which industry or according to which these alternatives are being evaluated. As discussed earlier, the requirements and business models of the industries in which DL platforms will be used are decisive regarding the performance of these tools and their expectations. In the literature, researchers have ignored this issue. Therefore, current research cannot provide a verifiable, comprehensive and up-to-date analysis of the performance, efficacy and efficiency of DL software with mathematical tools.

When the existing research gaps on deep learning in the literature are evaluated in general, they may vary depending on the focus of the study, the purpose of the research and the relevant industry. In that regard, Zhang et al. (2021) pointed out the weakness of the ability to generalize as one of the most critical gaps in studies on deep learning. According to the authors, deep learning technologies often use identified and curated training data, omitting how it can be generalized to real-life data. Therefore, using real-life data to train deep learning technologies represents the elimination of this critical gap.

Another notable gap is the difficulty of training deep-learning models with insufficient data. Wong et al. (2016) delved into strategies for training deep-learning models with minimal data. Addressing this challenge could potentially enhance the performance of such models. Furthermore, Dean et al. (2012) identified substantial research gaps concerning the classification and processing of large-scale data by deep learning models. They particularly underscored the significance of scalability and distributed learning in this context.

Effectively managing uncertainties is crucial for integrating deep learning models into various industry sectors' business models. Uncertainties are prevalent across diverse industries. Gal and Ghahramani (2016) examined the capability of deep learning models to address and manage uncertainties. Challenges such as the prolonged training time of deep learning models (Parikh, 2014) and their susceptibility to substantial security vulnerabilities (Madry et al., 2018) pose significant obstacles to adopting deep learning methodologies in industry business models.

In addition to the technical aspects of deep learning models, there are significant research gaps in comparing and evaluating existing deep learning technologies. Do et al. (2019) underlined the lack of extensive research focusing on the performance and efficiency of existing deep-learning platforms. In addition, the platforms' ease of use and flexibility are subjects often overlooked by studies in the literature. Abadi et al. (2016) emphasized that user experiences are frequently not considered when evaluating deep learning platforms.

Furthermore, there is a paucity of comprehensive comparisons among the communities of deep learning platforms. This deficiency may hinder our understanding of the size, efficacy, and level of support within deep learning communities (Paszke et al., 2019). Additionally, there is limited scrutiny of the documentation quality, educational resource richness, and impact on the learning process offered by existing deep learning platforms, which is essential for a more robust evaluation (Chen et al., 2015). The subsequent section briefly discusses the main objectives and motivations of the paper.

2.2. Motivation and objectives of the work

The primary motivation of this work is to develop and recommend a practical and robust decision-making tool for industry professionals aiming to design intelligent systems by integrating DL platforms with business models. In this context, the proposed model can be used as a roadmap for DL platform manufacturers to consider when developing their products while helping decision-makers in different industries to make more optimal decisions about the DL platform. At the same time, this study provides a comprehensive field study and a set of criteria determined by expert opinions to manage the evaluation process well and adequately structure the decision-making problem. In addition, providing an overview of DL technologies is another motivation to work to reduce the lack of knowledge of decision-makers in industries towards DL applications. In addition, in this study, we provide comprehensive information about current DL software alternatives and note these platforms' relative capabilities, advantages and disadvantages, including DL software that has not been evaluated in previous studies and made available to new users.

3. Mathematical model for evaluation

3.1. Objective and subjective model for determining criteria weights

In the following section, two models are presented to define the weighting coefficients of the criteria (Fig. 1). The first model represents a novel model based on the definition of maximum criterion values from the initial matrix (MAXC method). Since the MAXC method defines criteria weights based on predefined information in the initial matrix, we will treat this model as an objective methodology.

The second model is based on defining weight coefficients based on the subjective preferences of decision-makers. A linear programming model based on the ordinal priority approach was used to process expert preferences. In the following part, the mentioned models are presented.

3.1.1. MAXC objective method

The new approach (MAXC method) is used to define the criteria weights objectively and is presented in the following section.

Step 1. Creation of the decision-making matrix. The common way to present the ranking of alternatives is a decision-making matrix of the following form:

$$D = [x_{ij}]_{m\times n} = \begin{bmatrix} A/C & C_1 & C_2 & \cdots & C_n \\ target & max \vee min & max \vee min & \cdots & max \vee min \\ A_1 & x_{11} & x_{12} & \cdots & x_{1j} \\ A_2 & x_{21} & x_{22} & \cdots & x_{2j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_m & x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \quad (1)$$

The elements of the decision matrix are as follows:

$A = [A_1, A_2, \cdots, A_m]$– a finite set of alternatives, where m presents the total number of alternatives that should be ranked.

$C = [C_1, C_2, \cdots, C_n]$– a finite set of criteria, where n is the total number of criteria to be used in the process of ranking.

$[x_{ij}]_{m\times n}$– an assessment of alternative A_i with respect to j-th criterion

target– the desired value of the criterion

If all desired values are uniformly distributed, i.e., all criteria values tend to get max or min target, one of the criteria must be transformed into the opposite target value. The calculation of the reciprocal value of a selected criterion does it. Such an approach simultaneously creates a binomial environment, max and min space. Binomial decision-making space concerning targets is a precondition for applying the evaluation method (TODIFFA).

Step 2. Normalization of criteria data. To define the weights of criteria objectively, we normalize elements of the decision matrix by the linearization as follows:

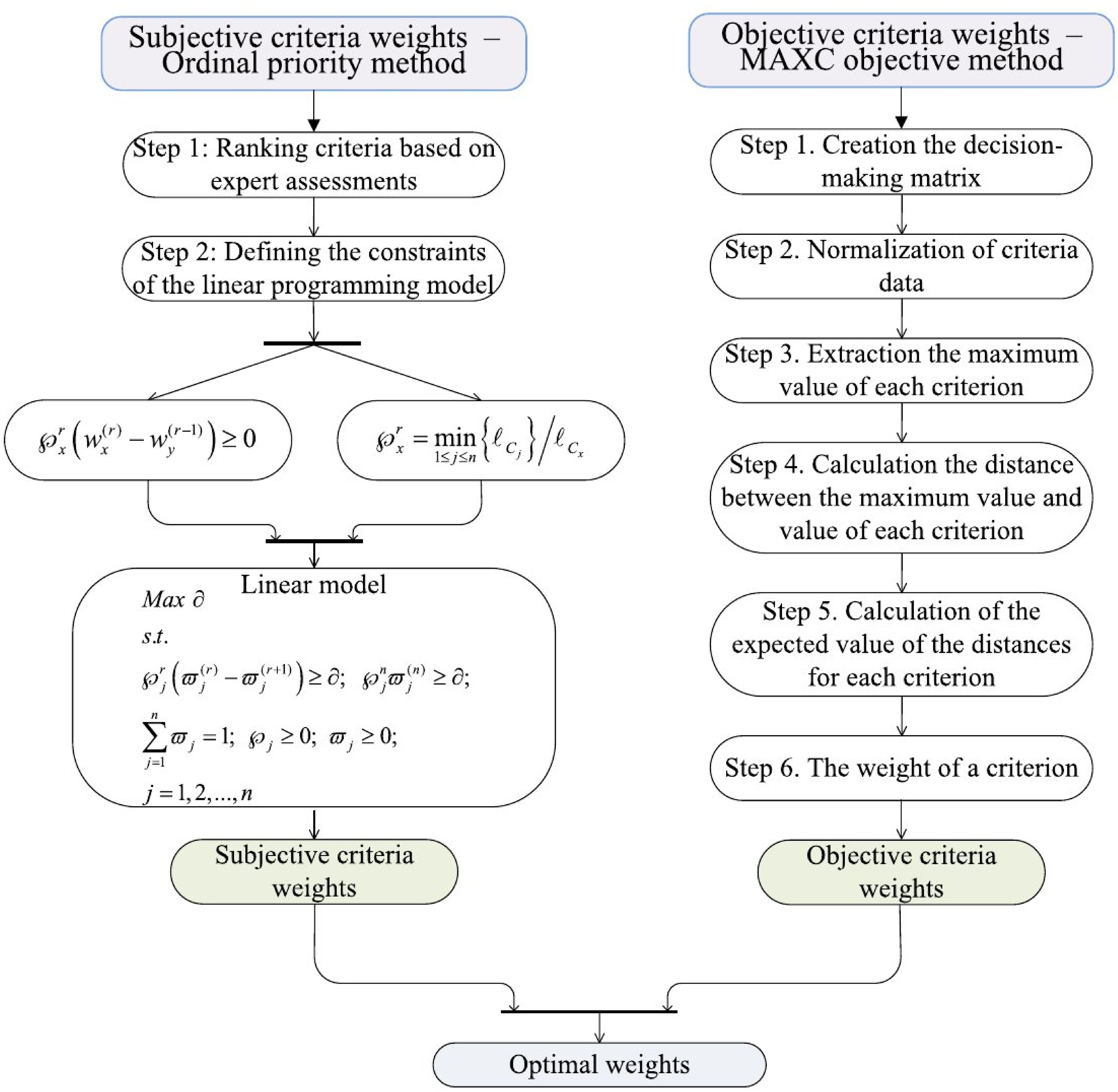

Fig. 1. Methodology for determining criteria weights.

$$r_{ij} = \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}}, \forall j \in [1, n] \tag{2}$$

The output of normalization is a normalized decision matrix:

$$R = \left[r_{ij}\right]_{m\times n} = \begin{bmatrix} A/C & C_1 & C_2 & \cdots & C_n \\ target & max \vee min & max \vee min & \cdots & max \vee min \\ A_1 & r_{11} & r_{12} & \cdots & r_{1j} \\ A_2 & r_{21} & r_{22} & \cdots & r_{2j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_m & r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix} \tag{3}$$

Step 3. Extraction the maximum value of each criterion. In this step, we extract the maximum value of the *j-th* criterion according to the following equation:

$$r_{ij}(max) = max\left(r_{ij} | 1 \leq j \leq n\right), \forall i \in [1, 2, \cdots, m] \tag{4}$$

Step 4. Calculation the distance between the maximum value and value of each criterion. The distance between the maximum value of the *j-th* criterion and the r_{ij} value of the criterion is calculated in the following way:

$$d_{ij} = r_{ij}(max) - r_{ij}, i = 1, 2, \cdots, m, \forall j \in [1, n] \tag{5}$$

Step 5. Calculation of the expected value of the distances for each criterion. The expected value of the distances for each criterion, calculated in the previous step, are expressed as:

$$E_j = \frac{d_{ij}}{\sum_{i=1}^{m} d_{ij}}, \forall j \in [1, n] \tag{6}$$

Step 6. The weight of a criterion. The weight of the *j-th* criterion is as follows:

$$\delta_j = \frac{E_j}{\sum_{j=1}^{n} E_j} \tag{7}$$

Where δ_j ($j = 1,2,\ldots,n$) represents the final objective criteria weights.

3.1.2. Ordinal priority subjective method

The ordinal priority method is based on processing expert preferences using linear programming. To eliminate inconsistencies in expert preferences, expert assessments were ranked in ascending order and were used to define the weighting coefficients of the criteria. In the following part, the ordinal priority method algorithm is presented.

Step 1. Ranking criteria based on expert assessments.

Let's assume that e experts participate in the research and that the experts' significance assessments are presented based on a predefined linguistic scale. We can then present expert assessments in the matrix $\square^b = \left[\ell_{C_j}^b\right]_{n\times 1}$, $1 \leq b \leq e$, where $\ell_{C_j}^b$ represents the significance assessment of the jth ($j = 1,2,\ldots,n$) criteria performed by the bth expert. By aggregating the expert assessments from the matrix $\square^b$, we get the aggregated matrix $\square = \left[\ell_{C_j}\right]_{n\times 1}$, where ℓ_{C_j} represents the aggregated value. Based on the value of ℓ_{C_j}, the criteria were ranked, where a higher value of ℓ_{C_j} means that the criterion has greater importance.

Step 2. Defining the constraints of the linear programming model. As defined in *Step 1*, the criteria are ranked in descending order, from the most significant (best criterion) to the least significant (worst criterion). Then, for the two criteria C_x and C_y ($1 \leq x,y \leq n$), where the criterion C_x has greater importance than the criterion C_y, we can say that their weighting coefficients meet the condition that $w_x^{(r)} - w_y^{(r+1)} \geqslant 0$. Where r represents the rank of criterion C_x, while $r+1$ represents the rank of criterion C_y. Also, based on the defined settings, we can define the following condition:

$$\wp_x^r \left(w_x^{(r)} - w_y^{(r-1)}\right) \geqslant 0 \tag{8}$$

where the value of $\wp_x^r$ is obtained by applying expression (8):

$$\wp_x^r = \frac{\min\limits_{1 \leqslant j \leqslant n} \left\{\ell_{C_j}\right\}}{\ell_{C_x}} \tag{9}$$

where $1 \leq x \leq n$, while ℓ_{C_j} represent the elements of the matrix $\mathbb{Q}$.

Step 3. Defining the linear programming model for the calculation of weight coefficients of criteria. Based on conditions (8) and (9), we can define a linear model for defining the weighting coefficients of the criteria as follows:

$$\begin{aligned} & Max\partial \\ & s.t. \\ & \wp_j^r \left(\varpi_j^{(r)} - \varpi_j^{(r+1)}\right) \geqslant \partial; \\ & \wp_j^n \varpi_j^{(n)} \geqslant \partial; \\ & \sum_{j=1}^{n} \varpi_j = 1; \\ & \wp_j \geqslant 0; \varpi_j \geqslant 0; \\ & j = 1, 2, \ldots, n \end{aligned} \tag{10}$$

where $\varpi_j = (\varpi_1, \varpi_2, \ldots, \varpi_n)^T$ represents a vector of weighting coefficients of criteria defined using expression (10).

By applying the expression (10), a fusion of the objective and subjective values of the weighting coefficients of the criteria as follows:

$$w_j = \frac{\alpha \cdot \delta_j + (1-\alpha) \cdot \varpi_j}{\alpha \cdot \sum_{j=1}^{n} \delta_j + (1-\alpha) \cdot \sum_{j=1}^{n} \varpi_j} \tag{11}$$

Where w_j ($j = 1,2,\ldots,n$) represent the aggregated values of the weighting coefficients used in the model to evaluate alternatives, while δ_j and ϖ_j respectively represent the objective and subjective weighting coefficients. The $\alpha \in [0, 1]$ parameter defines the influence of objective and subjective values (δ_j and ϖ_j) within the final decision.

3.2. New alternatives ranking by the total differential of alternative (TODIFFA) method

Since we define the criteria weights, the next phase is related to the alternatives ranking. We propose the following approach based on calculating the TOtal DIFFerential of Alternative, and the acronym is the TODIFFA method. A graphical presentation of steps for the TODIFFA method is presented in Fig. 2.

The following part presents the mathematical formulation of the steps shown in Fig. 2.

Step 1. Weighing the normalized decision matrix. Each element of a decision matrix (3) is weighted by the corresponding weight. The outcome of the weighting process is a weighted normalized decision matrix as follows:

$$Q = \left[q_{ij}\right]_{m\times n} = \begin{bmatrix} A/C & C_1 & C_2 & \cdots & C_n \\ target & max \vee min & max \vee min & \cdots & max \vee min \\ A_1 & q_{11} & q_{12} & \cdots & q_{1j} \\ A_2 & q_{21} & q_{22} & \cdots & q_{2j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_m & q_{m1} & q_{m2} & \cdots & q_{mn} \end{bmatrix} \tag{12}$$

Element q_{ij} of a matrix Q is calculated by:

$$q_{ij} = w_j r_{ij}, \forall i \in [1, 2, \cdots, m], \forall j \in [1, 2, \cdots, n] \tag{13}$$

The vector of criteria weights (criteria preference vector) is defined as:

$$W = \left[w_1, w_2, .., w_j\right], j \in [1, 2, \cdots, n], \sum_{j=1}^{n} w_j = 1 \tag{14}$$

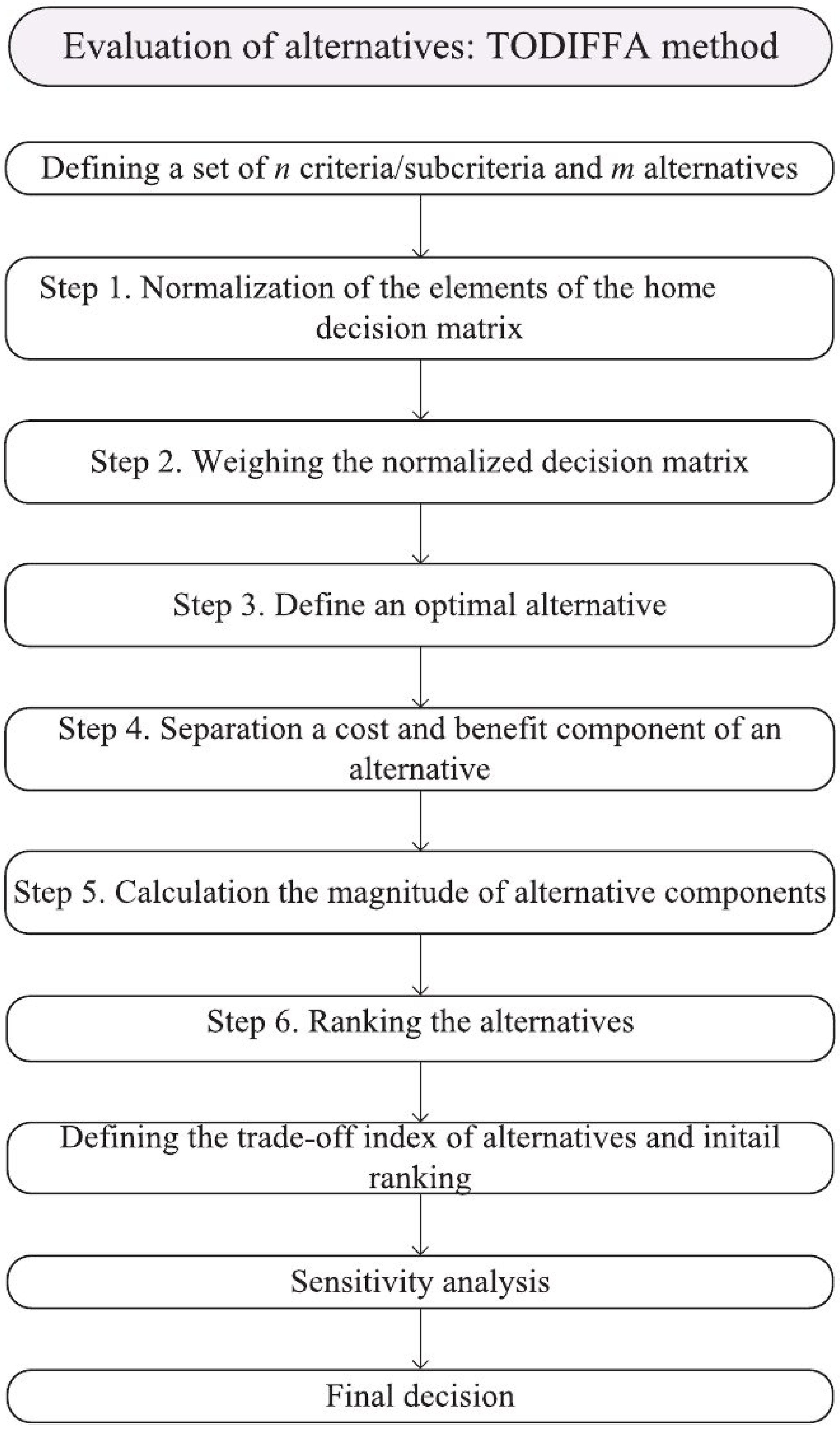

Fig. 2. TODIFFA evaluation method.

Vector W is defined using expression (11).

Step 2. Define an optimal alternative. An optimal alternative is composed of the following elements:

$$F^{opt} = \left[f_1^{opt}, f_2^{opt}, \cdots, f_j^{opt}\right], j \in [1, 2, \cdots, n] \tag{15}$$

Element f_j^{opt} is defined concerning the desired goal of the *j-th* criterion. It is calculated in the following way:

$$f_j^{opt} = \begin{cases} min\left(q_{ij} | j \in \{C\}\right), \forall i \in [1, 2, \cdots, m] \\ max\left(q_{ij} | j \in \{B\}\right), \forall i \in [1, 2, \cdots, m] \end{cases} \tag{16}$$

Where:
$\{C\}$– the set of cost criteria,
$\{B\}$– the set of benefit criteria.

Step 3. Separation a cost and benefit component of an alternative. Let c and b be the total number of cost and benefit criteria, respectively. Accordingly, an optimal alternative can be presented as the union of a cost criteria subset and a benefit criteria subset:

$$F^{opt} = F_c^{opt} \cup F_b^{opt}, c + b = j \tag{17}$$

$$F^{opt} = \left[f_1^{opt}, f_2^{opt}, \cdots, f_c^{opt}\right] \cup \left[f_1^{opt}, f_2^{opt}, \cdots, f_b^{opt}\right], b + c = j \tag{18}$$

Analogically, we can separate the cost and benefit components of each alternative:

$$F^i = F_c^i \cup F_b^i, \forall i \in [1, 2, \cdots, m], c + b = j \tag{19}$$

$$F^i = \left[q_1^i, q_2^i, \cdots, q_c^i\right] \cup \left[q_1^i, q_2^i, \cdots, q_b^i\right], \forall i \in [1, 2, \cdots, m], c + b = j \tag{20}$$

Step 4. Calculation of the magnitude of alternative components. The magnitude of extracted components equals the sum of elements that belong to a cost and benefit criteria subset.

For an optimal alternative:

$$M_c^{opt} = f_1^{opt} + f_2^{opt} + \cdots + f_c^{opt}, c \in \{C\} \tag{21}$$

$$M_b^{opt} = f_1^{opt} + f_2^{opt} + \cdots + f_b^{opt}, b \in \{B\} \tag{22}$$

Similarly, for each alternative:

$$M_c^i = q_1^i + q_2^i + \cdots + q_c^i, \forall i \in [1, 2, \cdots, m], c \in \{C\} \tag{23}$$

$$M_b^i = q_1^i + q_2^i + \cdots + q_b^i, \forall i \in [1, 2, \cdots, m], b \in \{B\} \tag{24}$$

Step 5. Ranking the alternatives. Order of alternatives is created according to the total differential of the alternative function $f(M_c, M_b)$. The total differential of the alternative function at point $\left(M_c^i, M_b^i\right), \forall i \in [1, 2, \cdots, m]$, approximates the function's change with respect to changes in variables M_c and M_b arround point $\left(M_c^i, M_b^i\right), \forall i \in [1, 2, \cdots, m]$.

Let A, B and C be a vertex of a right-angled triangle with the following coordinates: $A(0,0); B\left(M_c^i, 0\right); C\left(0, M_b^i\right)$. For simplicity, denote a right-angled triangle's base, height, and hypotenuse as x, y, and z respectively (see Fig. 3).

The function of variables x and y is a hypotenuse of a right-angled triangle, and it is defined as follows and presented in Fig. 4:

$$z = f(x, y) = \sqrt{x^2 + y^2} \tag{25}$$

Let $z = f(x, y)$ be a continuous function. Let dx and dy represent the changes in x and y, respectively. If partial derivatives $\frac{\partial z}{\partial x}$ and $\frac{\partial z}{\partial y}$ exist, then the total differential of z is as follows:

$$dz = df = \frac{\partial z}{\partial x} dx + \frac{\partial z}{\partial y} dy \tag{26}$$

It is well-known that total differential at a point (x_i, y_i) can be viewed as the linear approximation of the change of function z when the variables change by small amounts Δx and Δy. Hence, the change of function z (denoted by Δz) is approximated as $\Delta z \approx dz$. The total differential of the function $z = f(x, y) = \sqrt{x^2 + y^2}$ equals:

$$dz = \frac{x}{\sqrt{x^2 + y^2}} dx + \frac{y}{\sqrt{x^2 + y^2}} dy \tag{27}$$

Let $P_i(x_i, y_i)$ and $P_{opt}\left(x_{opt}, y_{opt}\right)$ be position of the *i-th* alternative and optimal alternative, respectively. If alternative position P_i is close enough to the optimal position P_{opt}, then the change of the function values is estimated by the total differential calculated in the following way:

$$dz_i = \frac{x_i}{\sqrt{x_i^2 + y_i^2}} \Delta x_i + \frac{y_i}{\sqrt{x_i^2 + y_i^2}} \Delta y_i, \forall i \in [1, 2, \cdots, m] \tag{28}$$

where $\Delta x_i = x_i - x_{opt}$, $\Delta y_i = y_{opt} - y_i$, $\forall i \in [1, 2, \cdots, m]$. Accordingly, the total differential of function $z_i = f_i(x_i, y_i)$ is as follows:

$$dz_i = \frac{x_i}{\sqrt{x_i^2 + y_i^2}} \left(x_i - x_{opt}\right) + \frac{y_i}{\sqrt{x_i^2 + y_i^2}} \left(y_{opt} - y_i\right), \forall i \in [1, 2, \cdots, m] \tag{29}$$

Substituting M_c^i, M_b^i, M_c^{opt}, and M_b^{opt} in equation (25), we get the following expression of the total differential of alternative function:

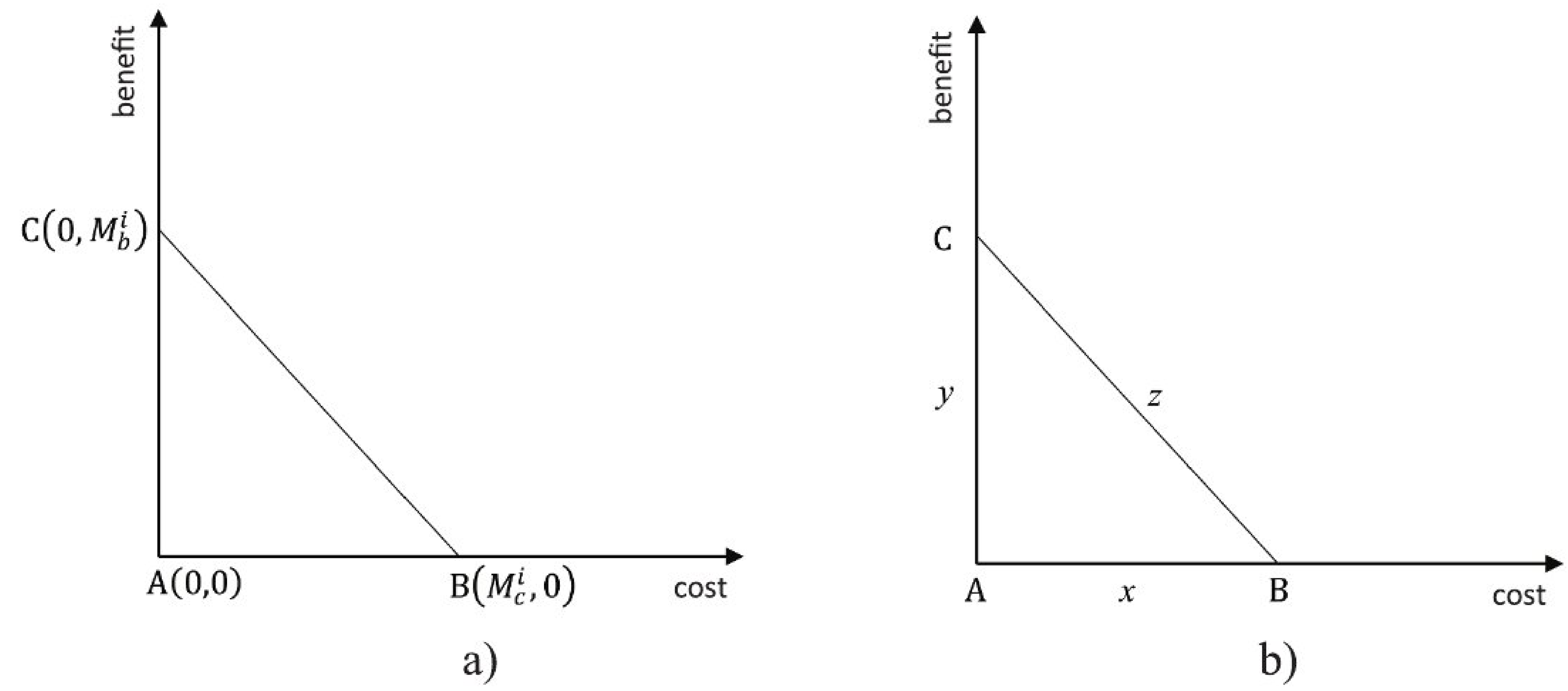

Fig. 3. (a) A right-angled triangle with cost and benefit component; (b) a simplified right-angled triangle.

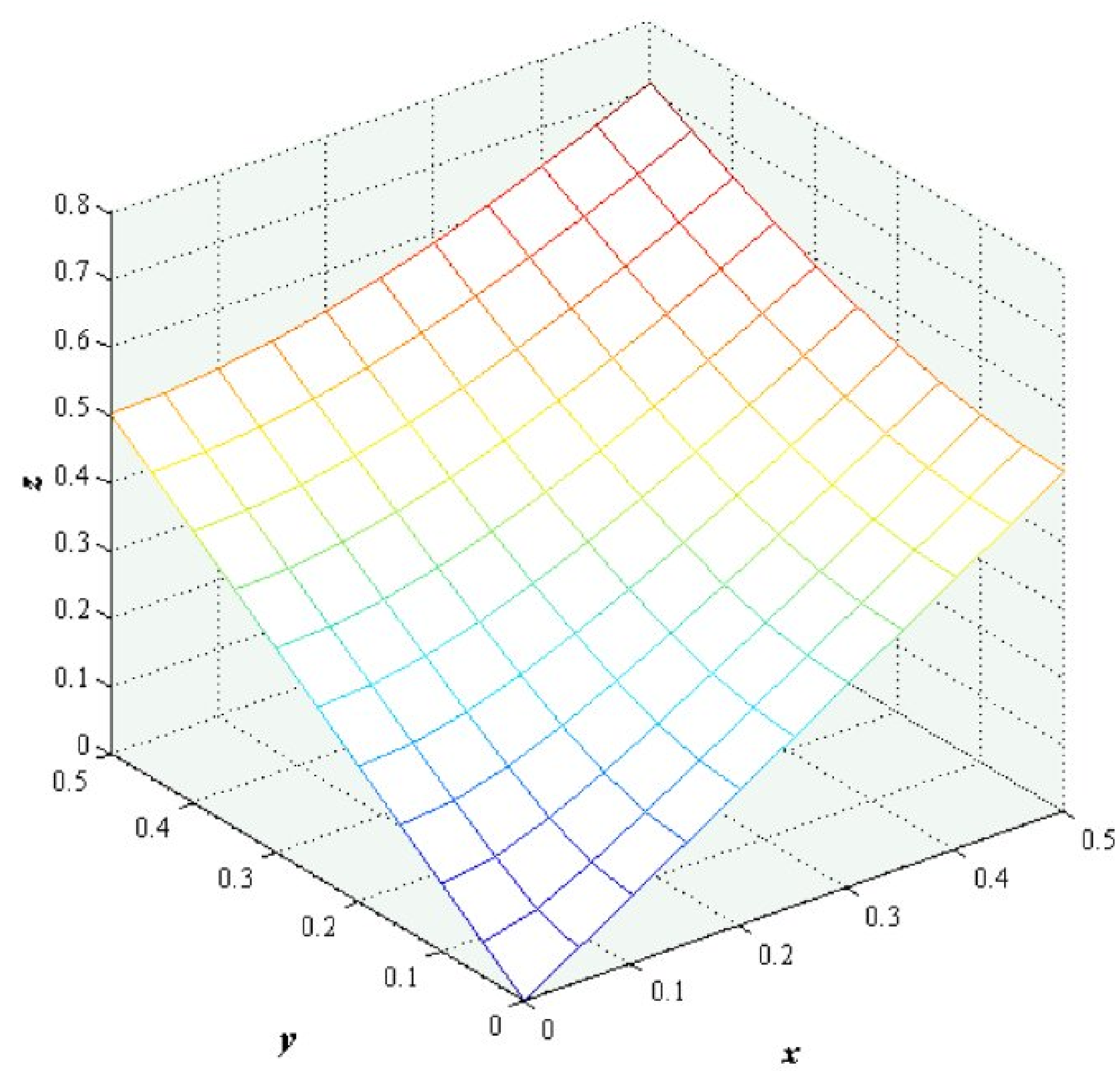

Fig. 4. Plot of a hypotenuse function.

$$dz_i = \frac{M_c^i}{\sqrt{{M_c^i}^2 + {M_b^i}^2}}\left(M_c^i - M_c^{opt}\right) + \frac{M_b^i}{\sqrt{{M_c^i}^2 + {M_b^i}^2}}\left(M_b^{opt} - M_b^i\right), \forall i \in [1, 2, \cdots, m] \tag{30}$$

Algebraically, the final equation of the total differential of the *i-th* alternative function is:

$$dz_i = \frac{M_c^i\left(M_c^i - M_c^{opt}\right) + M_b^i\left(M_b^{opt} - M_b^i\right)}{\sqrt{{M_c^i}^2 + {M_b^i}^2}}, \forall i \in [1, 2, \cdots, m] \tag{31}$$

The developed model performs alternatives ranking according to the ascending order of dz_i. The lower the total differential, the alternative is better. The pseudo-code format of the new Total Differential of Alternative (TODIFFA) Method is shown in Table 2.

4. Case study: Application of the proposed mathematical model

This study is designed to respond to the requirements of a large-scale vehicle manufacturer in the automotive industry, which is one of the largest-scale industries in Turkey. By integrating DL platforms into their business models, the top executives of this business aimed to create more intelligent systems and thus reduce disruptions and errors in business processes. They asked our research team for help in this regard. In our meeting with the company's senior executives, we defined the research problem. According to our preliminary information, the first and critical step in creating a well-functioning business intelligence by integrating DL architectures into the business models of the enterprise is the selection of an effective, efficient and high-performance DL platform. However, the research team members and the company's top managers lacked knowledge and experience in DL technologies. We decided to form a committee of experts to address this shortcoming. In order to be a member of the board of experts, We have identified three essential criteria: (1) to have graduated from computer engineering, management information systems or data processing and analysis departments of reputable universities, (2) to have experience and comprehensive knowledge in areas such as deep learning, machine learning, artificial intelligence business analytics, and (3) to have worked as a coordinator, researcher or analyst in international or national projects in the relevant field. Considering these criteria, we thoroughly screened professional networks and web pages of organizations and universities and identified nine professionals who met these criteria. Five of these professionals responded positively to our invitation. Table 3 provides detailed information on the professionals who are members of the board of experts.

After the expert committee members were determined, we held numerous meetings and interviews to evaluate and determine the alternatives. In the preliminary interviews, researchers directed participants to create two separate lists: one delineating the factors that impact the choice of deep learning technologies and the other outlining the characteristics of proficient, streamlined, and accessible deep learning platforms adaptable to automotive industry business frameworks. Experts were given a one-month timeframe to fulfil this assignment. Simultaneously, researchers conducted an extensive literature review to identify the criteria and options explored in previous studies on the same topic.

Throughout this process, the researchers conducted an extensive literature review, documenting the criteria and factors employed in studies that compared various deep learning platforms. Additionally, the platforms scrutinized in previous studies were also listed.

In the next round, the lists prepared by experts and researchers were combined, and two lists containing criteria and alternatives were prepared. Together with the experts, the researchers considered both lists and evaluated them separately; the criteria and alternatives that were repetitive in the same expression or meaning or covered by a different criterion were eliminated, and both lists were updated. The lists were then presented to the experts, who were asked to evaluate these criteria and alternatives according to the linguistic assessment scale presented in

Table 2
The pseudo-code format of TODIFFA method.

Input: Alternative criteria values $x_{ij}, i \in [1, m], j \in [1, n]$; where m equals number of alternatives A, and n equals number of criteria C
Output: Rank of alternatives $A_i, i \in [1, m]$

Begin
(1) Create a decision matrix
$$X = [x_{ij}]_{m \times n}$$
(2) Check for concurrently existence of min and max criteria
If min and max criteria concurrently exist **Then** go to (3)
ElseIf only min criteria exist **Then** calculate
$$\frac{1}{x_{ij}}, i \in [1, m], j = 1$$
Else only max criteria exist calculate
$$\frac{1}{x_{ij}}, i \in [1, m], j = 1$$
End If
(3) Calculate elements of normalized decision matrix
$$r_{ij} \leftarrow \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}}, \forall j \in [1, n]$$
(4) Define criteria weight vector
$$W = [w_1, w_2, .., w_j], j \in [1,2, \dots, n], \sum_{j=1}^{n} w_j = 1$$
(5) Calculate elements of weighted normalized decision matrix
$$q_{ij} \leftarrow w_j r_{ij}, \forall i \in [1,2, \dots, m], \forall j \in [1,2, \dots, n]$$
(6) Define an optimal alternative
$$F^{opt} = \left[f_1^{opt}, f_2^{opt}, \dots, f_j^{opt}\right], j \in [1,2, \dots, n]$$
$$f_j^{opt} \leftarrow min(q_{ij} | j \in \{C\}), \forall i \in [1,2, \dots, m] \quad \text{*for the set of cost criteria } \{C\} = \{1,2, \dots, c\}$$
$$f_j^{opt} \leftarrow max(q_{ij} | j \in \{B\}), \forall i \in [1,2, \dots, m] \quad \text{*for the set of benefit criteria } \{B\} = \{1,2, \dots, b\}$$
(7) Separation a cost and benefit component of an alternative
$$F^{opt} = \left[f_1^{opt}, f_2^{opt}, \dots, f_c^{opt}\right] \cup \left[f_1^{opt}, f_2^{opt}, \dots, f_b^{opt}\right], c + b = j \quad \text{*for an optimal alternative}$$
$$F^i = \left[q_1^i, q_2^i, \dots, q_c^i\right] \cup \left[q_1^i, q_2^i, \dots, q_b^i\right], \forall i \in [1,2, \dots, m], c + b = j \quad \text{*for each alternative}$$
(8) Calculation of the magnitude of alternative cost and benefit components
$$M_c^{opt} \leftarrow f_1^{opt} + f_2^{opt} + \dots + f_c^{opt}, c \in \{C\} \quad \text{*for an optimal alternative}$$
$$M_b^{opt} \leftarrow f_1^{opt} + f_2^{opt} + \dots + f_b^{opt}, b \in \{B\} \quad \text{*for an optimal alternative}$$
$$M_c^i \leftarrow q_1^i + q_2^i + \dots + q_c^i, \forall i \in [1,2, \dots, m], c \in \{C\} \quad \text{*for each alternative}$$
$$M_b^i \leftarrow q_1^i + q_2^i + \dots + q_b^i, \forall i \in [1,2, \dots, m], b \in \{B\} \quad \text{*for each alternative}$$
(9) Calculation of the total differential
$$dz_i \leftarrow \frac{M_c^i\left(M_c^i - M_c^{opt}\right) + M_b^i\left(M_b^{opt} - M_b^i\right)}{\sqrt{{M_c^i}^2 + {M_b^i}^2}}, \forall i \in [1,2, \dots, m] \quad \text{*for each alternative}$$
(10) Ranking the alternatives
$A_i, i \in [1, m]$ ←rank the alternatives based on the ascending order of dz_i values
End

Table 3
Details of the professionals of the board of experts.

DMs	Expertise	Title	Industry	Exp.	Graduate	Degree
DM1	Data Analysis	Assoc. Professor	Education (University)	19	MIS	Ph.D.
DM2	Data Analysis	Assist. Professor	Education (University)	15	MIS	Ph.D.
DM3	Data Analysis	Assist. Professor	Education (University)	14	Computer Eng.	Ph.D.
DM4	Data Analysis	Data analyst	Telecommunications	24	Electronic Eng.	Ph.D.
DM5	Data Management	IT Manager	Information Technology	18	Computer Eng.	Graduate

IT: Information technology, MIS: Management Information Systems, Eng.: Engineering.

step 1 within two weeks. Table 4 shows the criteria in the final list prepared in line with the literature review, the experts' opinions, and the experts' evaluations regarding the criteria.

As depicted in Table 4, upon conversion to net crisp values, the medium–low term corresponds to a calculated value of 4. Hence, criteria with an average rating falling between 1 and 4 are deemed uncritical. Conversely, criteria attaining a value of 6, corresponding to the term 4.1 to medium–high, are categorized as moderate, while those with an average value ranging from 6.1 to 9 are labelled as critical. Subsequently, each criterion underwent individual assessment, considering

Table 4
The updated criteria lists and their categories.

No	Criteria	E1	E2	E3	E4	E5	E1	E2	E3	E4	E5	Av.	Class	Reference
1	Acceleration and Optimization	L	VL	L	M	L	3	2	3	5	3	3.2	Uncritical	(Hatcher and Yu, 2018)
2	Architecture of tools	M	VL	M	L	M	5	2	5	3	5	4	Uncritical	(Sherkhane and Vora, 2017)
3	Autoencoder performance	VL	VL	AL	AL	VL	2	2	1	1	2	1.6	Uncritical	(Werda et al., 2022)
4	Batch Time	ML	AL	M	ML	AL	4	1	5	4	1	3	Uncritical	(Ulker et al., 2020)
5	Capability To Handle Multiple Inputs	H	VH	VH	VH	H	7	8	8	8	7	7.6	Critical	EO
6	Community support	L	L	L	L	L	3	3	3	3	3	3	Uncritical	(Rao, 2023)
7	Computational Resources	H	H	H	MH	H	7	7	7	6	7	6.8	Critical	EO
8	Cost of Installation	VH	P	P	P	VH	8	9	9	9	8	8.6	Critical	EO
9	CPU and GPU utilization	AL	VL	AL	AL	L	1	2	1	1	3	1.6	Uncritical	(Al-Bdour et al., 2020; Sherkhane and Vora, 2017)
10	Cross-Platform	M	L	M	AL	M	5	3	5	1	5	3.8	Uncritical	(Sherkhane and Vora, 2017)
11	Data Availability and Quality	P	P	P	P	VH	9	9	9	9	8	8.8	Critical	EO
12	Deep learning models	VL	VL	VL	VL	L	2	2	2	2	3	2.2	Uncritical	(Werda et al., 2022)
13	Distributed DL in IoT and CPS	L	VL	L	ML	L	3	2	3	4	3	3	Uncritical	(Hatcher and Yu, 2018)
14	Documentation	L	L	L	VL	L	3	3	3	2	3	2.8	Uncritical	(Rao, 2023)
15	Domain Knowledge	H	H	VH	VH	H	7	7	8	8	7	7.4	Critical	EO
16	Ease of use	L	VL	L	VL	L	3	2	3	2	3	2.6	Uncritical	(Rao, 2023)
17	Ecosystem	M	L	M	VL	M	5	3	5	2	5	4	Uncritical	(Sherkhane and Vora, 2017)
18	Epoch Time	L	L	VL	L	AL	3	3	2	3	1	2.4	Uncritical	(Ulker et al., 2020)
19	Explainability	VH	VH	VH	VH	VH	8	8	8	8	8	8	Critical	EO
20	Extensibility	M	L	L	L	L	5	3	3	3	3	3.4	Uncritical	(Sherkhane and Vora, 2017)
21	GPU size (memory)	H	H	VH	H	H	7	7	8	7	7	7.2	Critical	EO
22	Gradient Computation Time	AL	VL	AL	AL	AL	1	2	1	1	1	1.2	Uncritical	(Bahrampour et al., 2016; Werda et al., 2022)
23	Hardware utilization	M	L	L	L	M	5	3	3	3	5	3.8	Uncritical	(Sherkhane and Vora, 2017)
24	Image classification ability	VL	VL	AL	VL	VL	2	2	1	2	2	1.8	Uncritical	(Werda et al., 2022)
25	Interface	M	ML	M	ML	M	5	4	5	4	5	4.6	Moderate	(Sherkhane and Vora, 2017)
26	Memory consumption	AL	VL	L	AL	L	1	2	3	1	3	2	Uncritical	(Al-Bdour et al., 2020)
27	Modelling capability	ML	ML	M	ML	M	4	4	5	4	5	4.4	Moderate	(Sherkhane and Vora, 2017)
28	Network Management and Control	L	VL	L	M	L	3	2	3	5	3	3.2	Uncritical	(Hatcher and Yu, 2018)
29	Open source	M	M	M	ML	M	5	5	5	4	5	4.8	Moderate	(Sherkhane and Vora, 2017)
30	Performance	L	VL	VL	VL	L	3	2	2	2	3	2.4	Uncritical	(Rao, 2023; Sherkhane and Vora, 2017)
31	Problem Type	VH	P	VH	P	VH	8	9	8	9	8	8.4	Critical	EO
32	RAM size (memory)	VH	VH	VH	P	VH	8	8	8	9	8	8.2	Critical	EO
33	Running time	AL	VL	L	AL	L	1	2	3	1	3	2	Uncritical	(Şeker et al., 2017; Al-Bdour et al., 2020)
34	Scalability	H	H	H	H	H	7	7	7	7	7	7	Critical	EO
35	Secure DL	AL	VL	L	M	L	1	2	3	5	3	2.8	Uncritical	(Hatcher and Yu, 2018)
36	Sparse coding	VL	VL	VL	VL	VL	2	2	2	2	2	2	Uncritical	(Werda et al., 2022)
37	Storage size	MH	H	H	MH	H	6	7	7	6	7	6.6	Critical	EO
38	Supported languages	AL	AL	AL	AL	AL	1	1	1	1	1	1	Uncritical	(Kabakuş, 2020)
39	Testing time	AL	VL	AL	AL	VL	1	2	1	1	2	1.4	Uncritical	(Kabakuş, 2020)
40	Time Constraints	P	P	P	P	VH	9	9	9	9	8	8.8	Critical	EO
41	Training speed	ML	ML	M	ML	AL	4	4	5	4	1	3.6	Uncritical	(Shi et al., 2017)
42	Training time	AL	VL	AL	AL	AL	1	2	1	1	1	1.2	Uncritical	(Kabakuş, 2020)

the assigned classification. Ultimately, through unanimous consensus among the experts, only the criteria falling within the critical classification were retained for inclusion in the study scope, while others were excluded. Table 5 outlines the effective criteria for evaluating deep learning platforms alongside their respective definitions.

Afterwards, experts similarly evaluated all existing deep-learning platforms. In this context, the researchers identified 37 different deep-learning platforms with the help of experts. Table 6 shows the identified deep learning platforms and the experts' evaluations. At the same time, according to experts' assessments, the categories in which the alternatives are classified are shown.

Table 7 shows the alternatives identified and their definitions. According to the experts' suggestion, only the options are in the critical categorization, and others have been eliminated.

After determining the criteria and alternatives, we collected data for data analysis. Next, we passed the computational implementation to identify the relative significance of the criteria and preference ratings of the DL platform options. The mathematical model presented in this paper is implemented through two phases. In the first phase, it is necessary to calculate the weighting coefficients of the criteria using objective and subjective methods for determining the weighting coefficients. After defining the aggregate weight coefficients of the criteria, the evaluation of alternatives using the TODIFFA method was presented in the second phase. In the following part, the application of the MCMD model is presented.

4.1. Determining the weights of the criteria

4.1.1. MAXC method: Application of an objective approach for determining weighting coefficients of criteria

The following section shows the application of the MAXC method as follows:

Step 1:

Five experts participated in the research. The experts evaluated the alternatives using a nine-level scale: 1 – Absolutely Low (AL), 2 – Very Low (VL), 3 – Low (L), 4 – Medium Low (ML), 5 – Medium (M), 6 – Medium High (MH), 7 – High (H), 8 – Very High (VH), 9 – Perfect (P). Expert assessments of alternatives are presented in Table A1 (Appendix). Arithmetic averaging defined the aggregated matrix (1) presented in Table 8.

Step 2:

Applying linear normalization by equation (2), we get a normalized decision matrix, and it is presented in Table 9.

Step 3:

According to expression (4), we extract the maximum value of each criterion. Table 10 shows the extracted values.

Table 5
Detailed information on the criteria parameters used in the current work.

No	Criteria	Descriptions	Direction
C1	Problem type	The ability of tools for description of the problem as classification, regression, or clustering	Max
C2	Data Availability and Quality	DL tools must manage data size, structure, handle missing values, and outliers.	Max
C3	Computational resources	It refers to being computationally efficient and not requiring excessive resources.	Max
C4	Explainability	Being able to explain its predictions intelligibly	Max
C5	Scalability	DL tools must scale to manage large datasets and adapt to diverse data distributions.	Max
C6	Capability to handle multiple inputs	DL tools must scale to handle various inputs: numerical, categorical, and images.	Max
C7	Time constraints	DL tools should predict promptly when rapid responses are needed.	Min
C8	Domain knowledge	Being compatible with domain knowledge.	Max
C9	Cost of Installation	Acquisition costs of DL tools for a company	Min
C10	GPU size (memory)	The capacity of GPU concerning data processing	Max
C11	Storage size	Storage size is the data storage capacity of a DL platform.	Max
C12	RAM size (memory)	In DL, analysts handle large datasets exceeding CPU cache capacity.	Max

According to expression (4), the maximum value of criterion C1 is defined as follows:

$$r_1 = max(0.05581, 0.04676, 0.05279, 0.05279, 0.04827, \cdots, 0.06033, 0.05732, 0.05430) = 0.06033$$

Step 4:

The distance between the maximum value of each criterion and x_{ij} value of the criterion is calculated by expression (5). A matrix of distances is presented in Table 11.

Based on expression (5), the distance of A1C1 from r_1 equals:

$$d_{11} = r_1 - r_{11} = 0.06033 - 0.05581 = 0.00452$$

Step 5:

Applying expression (6), we calculate the expected value of the distances for each criterion, and the results are shown in Table 12.

Based on expression (6), expected value of distances for criterion C1 is calculated as follows:

$$E_1 = \frac{0.00452 + 0.01357 + 0.00754 + 0.00754 + 0.01207 + \cdots + 0.0000 + 0.00302 + 0.00603}{19} = 0.00770$$

Step 6:

Expression (7) was used to define the weight of each criterion, and the obtained weights are presented in Table 13.

Objective weight for criterion C1 equals:

$$w_1 = \frac{0.00770}{0.00770 + 0.01316 + 0.00844 + 0.00807 + \cdots + 0.00725 + 0.01168 + 0.00725} = 0.08131$$

The efficiency of the developed objective MAXC method for criteria weights calculation is estimated by comparison with the Standard Deviation and Entropy methods (Görçün, 2020). Values of criteria weights of these three methods are presented in Table 14 and Fig. 5.

The graphic representation of the weighting coefficients of the criteria from Table 14 is presented in Fig. 5.

Mean absolute error (MAE) and Pearson correlation coefficient (PCC) are used as indicators of the efficiency of the new MAXC method. These two indicators are presented in Table 15.

The maximum value of absolute error is 0.03131 (MAXC versus Entropy), and the minimum value is 0.00267 (MAXC versus St.Dev.), while the average value is 0.01448. All values show that the MAXC method is very efficient. We can draw the same conclusion even if we use only the maximum value. According to the value of PCC, r = 0.90222, the relationship between MAXC and the St.Dev. method is considered to have a high positive correlation. PCC of 0.89276 between MAXC and Entropy method also shows a high positive correlation. The average value of 0.89749 belongs to a high positive correlation as well. Hence, this indicator also shows that MAXC is very efficient. Finally, both indicators show that the new MAXC method can define criteria weights objectively and efficiently.

4.1.2. *Ordinal priority method: Application of a subjective approach to determine the weighting coefficients of the criteria*

In the following section, the application of the ordinal priority method for determining the weight coefficients of the criteria is presented:

Step 1:

Five experts evaluated the criteria. The experts assessed the significance using the same nine-point scale to evaluate the alternatives. Expert assessments are presented in Table A2 (Appendix). Aggregation of expert assessments from Table A2 defined the rank of criteria C2 > C7 > C1 > C12 > C4 > C6 > C8 > C10 > C5 > C3 > C11 > C9.

Step 2:

Table 6
The updated criteria lists and their categories.

No	Criteria	E1	E2	E3	E4	E5	E1	E2	E3	E4	E5	Av.	Class	Reference
1	AWS Deep Learning AMIs	VH	H	P	VH	H	8	7	9	8	7	7.8	Critical	EO
2	Bitnami Pytorch	MH	M	VH	M	P	6	5	8	5	9	6.6	Critical	EO
3	Chainer	MH	M	H	ML	P	6	5	7	4	9	6.2	Critical	EO
4	Clarifai	H	MH	VH	H	H	7	6	8	7	7	7	Critical	EO
5	CNTK	P	VH	P	P	P	9	8	9	9	9	8.8	Critical	(Hatcher and Yu, 2018; Shi et al., 2017; Kabakuş, 2020; Al-Bdour et al., 2020; Luckow et al., 2016)
6	Coffee	MH	MH	VH	M	P	6	6	8	5	9	6.8	Critical	(Hatcher and Yu, 2018; Şeker et al., 2017; Shi et al., 2017; Kabakuş, 2020; Dong et al., 2023; Rao, 2023; Sherkhane and Vora, 2017; Werda et al., 2022; Luckow et al., 2016)
7	Crino	AL	VL	AL	AL	AL	1	2	1	1	1	1.2	Uncritical	(Werda et al., 2022)
8	Cuda CNN	VL	VL	VL	VL	AL	2	2	2	2	1	1.8	Uncritical	(Werda et al., 2022)
9	Cuda-convnet	L	VL	L	VL	L	3	2	3	2	3	2.6	Uncritical	(Sherkhane and Vora, 2017; Werda et al., 2022)
10	CXXNET	VL	VL	L	VL	VL	2	2	3	2	2	2.2	Uncritical	(Werda et al., 2022)
11	Darch	AL	L	ML	L	VL	1	3	4	3	2	2.6	Uncritical	(Werda et al., 2022)
12	DeepLearning4J	VL	L	ML	L	AL	2	3	4	3	1	2.6	Uncritical	(Hatcher and Yu, 2018; Şeker et al., 2017)
13	DeepLearnToolbox	VL	VL	VL	VL	VL	2	2	2	2	2	2	Uncritical	(Werda et al., 2022)
14	Deepmat	VL	AL	L	ML	VL	2	1	3	4	2	2.4	Uncritical	(Werda et al., 2022)
15	Deepnet	VL	L	L	ML	VL	2	3	3	4	2	2.8	Uncritical	(Werda et al., 2022)
16	DeepPy	M	H	H	ML	VH	5	7	7	4	8	6.2	Moderate	EO
17	EBLearn	AL	VL	AL	VL	AL	1	2	1	2	1	1.4	Uncritical	EO
18	Google Cloud Deep Learning Con.	H	H	VH	H	H	7	7	8	7	7	7.2	Critical	(Werda et al., 2022)
19	Hebel	L	VL	L	L	L	3	2	3	3	3	2.8	Uncritical	(Werda et al., 2022)
20	Keras	H	H	VH	VH	H	7	7	8	8	7	7.4	Critical	(Hatcher and Yu, 2018; Bahrampour et al., 2016; Kabakuş, 2020; Rao, 2023; Luckow et al., 2016)
21	Knet	P	H	P	P	H	9	7	9	9	7	8.2	Critical	(Şeker et al., 2017)
22	Lush	VL	L	L	L	AL	2	3	3	3	1	2.4	Uncritical	(Werda et al., 2022)
23	Microsoft Cognitive Toolkit	MH	MH	VH	H	MH	6	6	8	7	6	6.6	Critical	EO
24	MXNet	L	AL	ML	L	L	3	1	4	3	3	2.8	Uncritical	(Hatcher and Yu, 2018; Rao, 2023; Luckow et al., 2016)
25	Neon	P	H	P	VH	H	9	7	9	8	7	8	Critical	(Al-Bdour et al., 2020)
26	Neuroph	P	M	H	ML	P	9	5	7	4	9	6	Critical	EO
27	Neuton AutoML	VH	H	VH	VH	H	8	7	8	8	7	7.6	Critical	EO
28	nnForge	L	VL	L	VL	VL	3	2	3	2	2	2.4	Uncritical	(Werda et al., 2022)
29	NVIDIA Deep Learning	MH	M	H	M	P	6	5	7	5	9	6.4	Critical	EO
30	Options	VL	VL	AL	VL	AL	2	2	1	2	1	1.6	Uncritical	(Hatcher and Yu, 2018)
31	Pylearn	AL	AL	ML	L	L	1	1	4	3	3	2.4	Uncritical	(Sherkhane and Vora, 2017; Werda et al., 2022)
32	PyTorch	AL	AL	ML	L	VL	1	1	4	3	2	2.2	Uncritical	(Kabakuş, 2020; Bahrampour et al., 2016; Rao, 2023)
33	R-CNN	AL	AL	AL	AL	AL	1	1	1	1	1	1	Uncritical	(Werda et al., 2022)
34	Swift AI	H	MH	VH	H	MH	7	6	8	7	6	6.8	Critical	EO
35	TensorFlow	P	P	P	P	P	9	9	9	9	9	9	Critical	(Hatcher and Yu, 2018; Şeker et al., 2017; Shi et al., 2017; Dong et al., 2023; Rao, 2023; Luckow et al., 2016)
36	Theano	P	VH	P	P	H	9	8	9	9	7	8.4	Critical	(Hatcher and Yu, 2018; Şeker et al., 2017; Kabakuş, 2020; Dong et al., 2023; Sherkhane and Vora, 2017; Werda et al., 2022; Luckow et al., 2016)
37	Torch	P	VH	P	P	VH	9	8	9	9	8	8.6	Critical	(Şeker et al., 2017; Shi et al., 2017; Kabakuş, 2020; Rao, 2023; Sherkhane and Vora, 2017; Luckow et al., 2016)

Table 7
Detailed information on the DL software assessed in the current study.

No	Alternatives	Descriptions
A1	Google Cloud Deep Learning Containers	Google Cloud Deep Learning Containers is a Google Cloud Platform (GCP) DL platform.
A2	Microsoft Cognitive Toolkit	It is a deep-learning framework developed by Microsoft.
A3	Neuton AutoML	Neuton: No-code Tiny AutoML platform with patented Neural Network Framework.
A4	Knet	Knet is Koç University's deep learning framework, implemented in Julia.
A5	NVIDIA Deep Learning	NVIDIA AI Platform: DL tool for AI app development with GPU-accelerated frameworks.
A6	Swift AI	Swift AI: High-performance DL tool in Swift, supporting Apple platforms.
A7	Theano	Theano is an open-source DL platform developed at the MILA lab at the University of Montreal.
A8	Chainer	Chainer: Open-source DL framework in Python with NumPy and CuPy.
A9	Clarifai	Clarifai: Leading Generative AI and NLP for unstructured data modeling.
A10	Coffee	Caffe: DL tool by Berkeley AI Research (BAIR) for image classification.
A11	DeepPy	DeepPy is a Pythonic deep learning framework built on top of NumPy
A12	Bitnami Pytorch	PyTorch is a popular deep-learning framework developed by Facebook's AI research team in 2016.
A13	Neon	Neon: Multi-device system research framework by Autodesk Research.
A14	Neuroph	Neuroph: Java neural network framework supporting common architectures and learning rules.
A15	Torch	Torch: Scientific computing framework with ML algorithm components.
A16	AWS Deep Learning AMIs	AWS DLAMI: Unique Amazon option for cloud-based deep learning.
A17	CNTK	CNTK is an open-source DL framework developed by Microsoft Research.
A18	TensorFlow	TensorFlow is an open-source DL tool developed by the Google Brain Team.
A19	Keras	Keras: High-level DL framework with intuitive model building and training.

Standardized expert assessments were defined using the expression (9) used within the model (10). Standardized expert assessments (SEA) are shown in Table 16.

Step 3:

SEA values from Table 16 were used to define constraints in the model (10). The linear model for defining the subjective weight coefficients of the criteria is presented as follows:

$$\begin{aligned}
&Max\partial\\
&s.t.\\
&0.390\cdot(\varpi_2-\varpi_7)\geqslant\partial;\\
&0.400\cdot(\varpi_7-\varpi_1)\geqslant\partial;\\
&0.410\cdot(\varpi_1-\varpi_{12})\geqslant\partial;\\
&0.410\cdot(\varpi_{12}-\varpi_4)\geqslant\partial;\\
&0.432\cdot(\varpi_4-\varpi_6)\geqslant\partial;\\
&\ldots\\
&0.842\cdot(\varpi_{11}-\varpi_9)\geqslant\partial;\\
&1.000\cdot\varpi_9\geqslant\partial;\\
&\sum_{j=1}^{n}\varpi_j=1;\\
&\varpi_j\geqslant 0; j=1,2,\ldots,n
\end{aligned}$$

Subjective values of weight coefficients were obtained by solving a linear model using Lingo 17.0 software, Table 17.

To evaluate the efficiency of the applied subjective weighting method (Ordinal priority method) as well as the proposed model's effectiveness, a comparative analysis with RANCOM method (Więckowski et al., 2023; Shekhovtsov and Dobryakova, 2023) is presented in Table 18 and graphically illustrated in Fig. 6.

Pearson correlation coefficient (PCC) is used as indicator of the efficiency of the applied Ordinal priority method. PCC of 0.99585 between Ordinal priority and RANCOM method indicates on extremely positive correlation. According to obtained PCC value, it can be concluded that the Ordinal priority method is absolutely acceptable for defining the weights of criteria.

By applying expression (11), objective and subjective weighting coefficients from Tables 13 and 17 were merged. The final aggregated values of weighting coefficients are shown in Table 19 and Fig. 7.

The final values of the weighting coefficients from Fig. 7 were obtained for the value of the parameter $\alpha = 0.5$. By adopting the value $\alpha = 0.5$, the equal influence of subjective and objective weight coefficients in the final decision was simulated. As part of the sensitivity analysis, the influence of the parameter α on the change of the final weighting coefficients and the final decision was analyzed in detail.

Table 8
Aggregated decision matrix.

A-C	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
target	*max*	*max*	*max*	*max*	*max*	*max*	*min*	*max*	*min*	*max*	*max*	*max*
A1	7.4	6.2	7.6	6.6	6.2	7.2	7.2	6.2	6.6	6.6	6.6	7.2
A2	6.2	6.2	7	6.2	6.2	7	7.2	6.2	6.6	6.2	6.2	7.2
A3	7	7	6.2	7	7	6.2	7	7	6.6	7	7	6.2
A4	7	7	7	7	7	7	7	7	7	7	7	7
A5	6.4	6	6.4	6.4	6	6.4	7.4	6	6.8	7	6.2	7
A6	6.2	7	6	6.2	7	6	7.2	7	7.2	6.6	7	6.2
A7	8	6.4	7	7.8	6.4	7	7.4	6.6	7	8	7	7
A8	6.2	6	6.4	6.2	6	6.4	7.2	6	6.8	6.2	6	7
A9	7	6.2	7	6.8	6.2	7	7.4	6.2	7	7	6.6	7
A10	7	6.2	6	7	6.2	6	7	6.2	7.2	7	6.6	6.2
A11	6.4	5.4	6	6.4	5.4	6	7.4	5.4	7.2	7	5.6	6
A12	7	5.4	7	7	5.4	7	7	6	7	7	6	7
A13	7	6.2	8	7	6.2	8	7	6.2	6.8	7.2	6.2	8
A14	7	5.2	6.4	7	5.2	6.4	7	5.2	6.8	7.2	5.2	7
A15	7	7	8	7	7	8	7	7	6.8	7	7	8
A16	7	7	6.2	7	7	6.2	7	7	7	7.2	7	6.2
A17	8	7	8	8	7	8	7	7	6.8	8	7.2	8
A18	7.6	8	7.6	8	7.6	8	7	7.4	6.8	7.2	8	8
A19	7.2	6.2	7.2	7.2	6	7.2	7.2	6.4	6.4	7.2	6	7.4

Table 9
Normalized decision matrix.

A-C *target*	C1 *max*	C2 *max*	C3 *max*	C4 *max*	C5 *max*	C6 *max*	C7 *min*	C8 *max*	C9 *min*	C10 *max*	C11 *max*	C12 *max*
A1	0.05581	0.05099	0.05802	0.05008	0.05124	0.05496	0.05310	0.05082	0.05061	0.04940	0.05305	0.05389
A2	0.04676	0.05099	0.05344	0.04704	0.05124	0.05344	0.05310	0.05082	0.05061	0.04641	0.04984	0.05389
A3	0.05279	0.05757	0.04733	0.05311	0.05785	0.04733	0.05162	0.05738	0.05061	0.05240	0.05627	0.04641
A4	0.05279	0.05757	0.05344	0.05311	0.05785	0.05344	0.05162	0.05738	0.05368	0.05240	0.05627	0.05240
A5	0.04827	0.04934	0.04885	0.04856	0.04959	0.04885	0.05457	0.04918	0.05215	0.05240	0.04984	0.05240
A6	0.04676	0.05757	0.04580	0.04704	0.05785	0.04580	0.05310	0.05738	0.05521	0.04940	0.05627	0.04641
A7	0.06033	0.05263	0.05344	0.05918	0.05289	0.05344	0.05457	0.05410	0.05368	0.05988	0.05627	0.05240
A8	0.04676	0.04934	0.04885	0.04704	0.04959	0.04885	0.05310	0.04918	0.05215	0.04641	0.04823	0.05240
A9	0.05279	0.05099	0.05344	0.05159	0.05124	0.05344	0.05457	0.05082	0.05368	0.05240	0.05305	0.05240
A10	0.05279	0.05099	0.04580	0.05311	0.05124	0.04580	0.05162	0.05082	0.05521	0.05240	0.05305	0.04641
A11	0.04827	0.04441	0.04580	0.04856	0.04463	0.04580	0.05457	0.04426	0.05521	0.05240	0.04502	0.04491
A12	0.05279	0.04441	0.05344	0.05311	0.04463	0.05344	0.05162	0.04918	0.05368	0.05240	0.04823	0.05240
A13	0.05279	0.05099	0.06107	0.05311	0.05124	0.06107	0.05162	0.05082	0.05215	0.05389	0.04984	0.05988
A14	0.05279	0.04276	0.04885	0.05311	0.04298	0.04885	0.05162	0.04262	0.05215	0.05389	0.04180	0.05240
A15	0.05279	0.05757	0.06107	0.05311	0.05785	0.06107	0.05162	0.05738	0.05215	0.05240	0.05627	0.05988
A16	0.05279	0.05757	0.04733	0.05311	0.05785	0.04733	0.05162	0.05738	0.05368	0.05389	0.05627	0.04641
A17	0.06033	0.05757	0.06107	0.06070	0.05785	0.06107	0.05162	0.05738	0.05215	0.05988	0.05788	0.05988
A18	0.05732	0.06579	0.05802	0.06070	0.06281	0.06107	0.05162	0.06066	0.05215	0.05389	0.06431	0.05988
A19	0.05430	0.05099	0.05496	0.05463	0.04959	0.05496	0.05310	0.05246	0.04908	0.05389	0.04823	0.05539

Table 10
Maximum values of criteria.

Criterion	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
max	0.06033	0.06579	0.06107	0.06070	0.06281	0.06107	0.05457	0.06066	0.05521	0.05988	0.06431	0.05988

Table 11
Distances from the maximum value of each criterion.

A-C	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
A1	0.00452	0.01480	0.00305	0.01062	0.01157	0.00611	0.00147	0.00984	0.00460	0.01048	0.01125	0.00599
A2	0.01357	0.01480	0.00763	0.01366	0.01157	0.00763	0.00147	0.00984	0.00460	0.01347	0.01447	0.00599
A3	0.00754	0.00822	0.01374	0.00759	0.00496	0.01374	0.00295	0.00328	0.00460	0.00749	0.00804	0.01347
A4	0.00754	0.00822	0.00763	0.00759	0.00496	0.00763	0.00295	0.00328	0.00153	0.00749	0.00804	0.00749
A5	0.01207	0.01645	0.01221	0.01214	0.01322	0.01221	0.00000	0.01148	0.00307	0.00749	0.01447	0.00749
A6	0.01357	0.00822	0.01527	0.01366	0.00496	0.01527	0.00147	0.00328	0.00000	0.01048	0.00804	0.01347
A7	0.00000	0.01316	0.00763	0.00152	0.00992	0.00763	0.00000	0.00656	0.00153	0.00000	0.00804	0.00749
A8	0.01357	0.01645	0.01221	0.01366	0.01322	0.01221	0.00147	0.01148	0.00307	0.01347	0.01608	0.00749
A9	0.00754	0.01480	0.00763	0.00910	0.01157	0.00763	0.00000	0.00984	0.00153	0.00749	0.01125	0.00749
A10	0.00754	0.01480	0.01527	0.00759	0.01157	0.01527	0.00295	0.00984	0.00000	0.00749	0.01125	0.01347
A11	0.01207	0.02138	0.01527	0.01214	0.01818	0.01527	0.00000	0.01639	0.00000	0.00749	0.01929	0.01497
A12	0.00754	0.02138	0.00763	0.00759	0.01818	0.00763	0.00295	0.01148	0.00153	0.00749	0.01608	0.00749
A13	0.00754	0.01480	0.00000	0.00759	0.01157	0.00000	0.00295	0.00984	0.00307	0.00599	0.01447	0.00000
A14	0.00754	0.02303	0.01221	0.00759	0.01983	0.01221	0.00295	0.01803	0.00307	0.00599	0.02251	0.00749
A15	0.00754	0.00822	0.00000	0.00759	0.00496	0.00000	0.00295	0.00328	0.00307	0.00749	0.00804	0.00000
A16	0.00754	0.00822	0.01374	0.00759	0.00496	0.01374	0.00295	0.00328	0.00153	0.00599	0.00804	0.01347
A17	0.00000	0.00822	0.00000	0.00000	0.00496	0.00000	0.00295	0.00328	0.00307	0.00000	0.00643	0.00000
A18	0.00302	0.00000	0.00305	0.00000	0.00000	0.00000	0.00295	0.00000	0.00307	0.00599	0.00000	0.00000
A19	0.00603	0.01480	0.00611	0.00607	0.01322	0.00611	0.00147	0.00820	0.00613	0.00599	0.01608	0.00449

Table 12
Expected values of distances.

A-C	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
E_j	0.00770	0.01316	0.00844	0.00807	0.01018	0.00844	0.00194	0.00802	0.00258	0.00725	0.01168	0.00725

Table 13
Objective criteria weights.

A-C	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
w_j	0.08131	0.13894	0.08909	0.08518	0.10748	0.08909	0.02049	0.08473	0.02728	0.07654	0.12331	0.07654

Table 14
Criteria weights obtained by three methods.

Method	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
MAXC	0.08131	0.13894	0.08909	0.08518	0.10748	0.08909	0.02049	0.08473	0.02728	0.07654	0.12331	0.07654
St.Dev.	0.07853	0.11098	0.10413	0.08090	0.10481	0.10598	0.02345	0.09410	0.03295	0.06651	0.10225	0.09541
Entropy	0.06645	0.13270	0.11658	0.07002	0.11941	0.12040	0.00592	0.09693	0.01177	0.04764	0.11361	0.09857

St.Dev. – Standard Deviation.

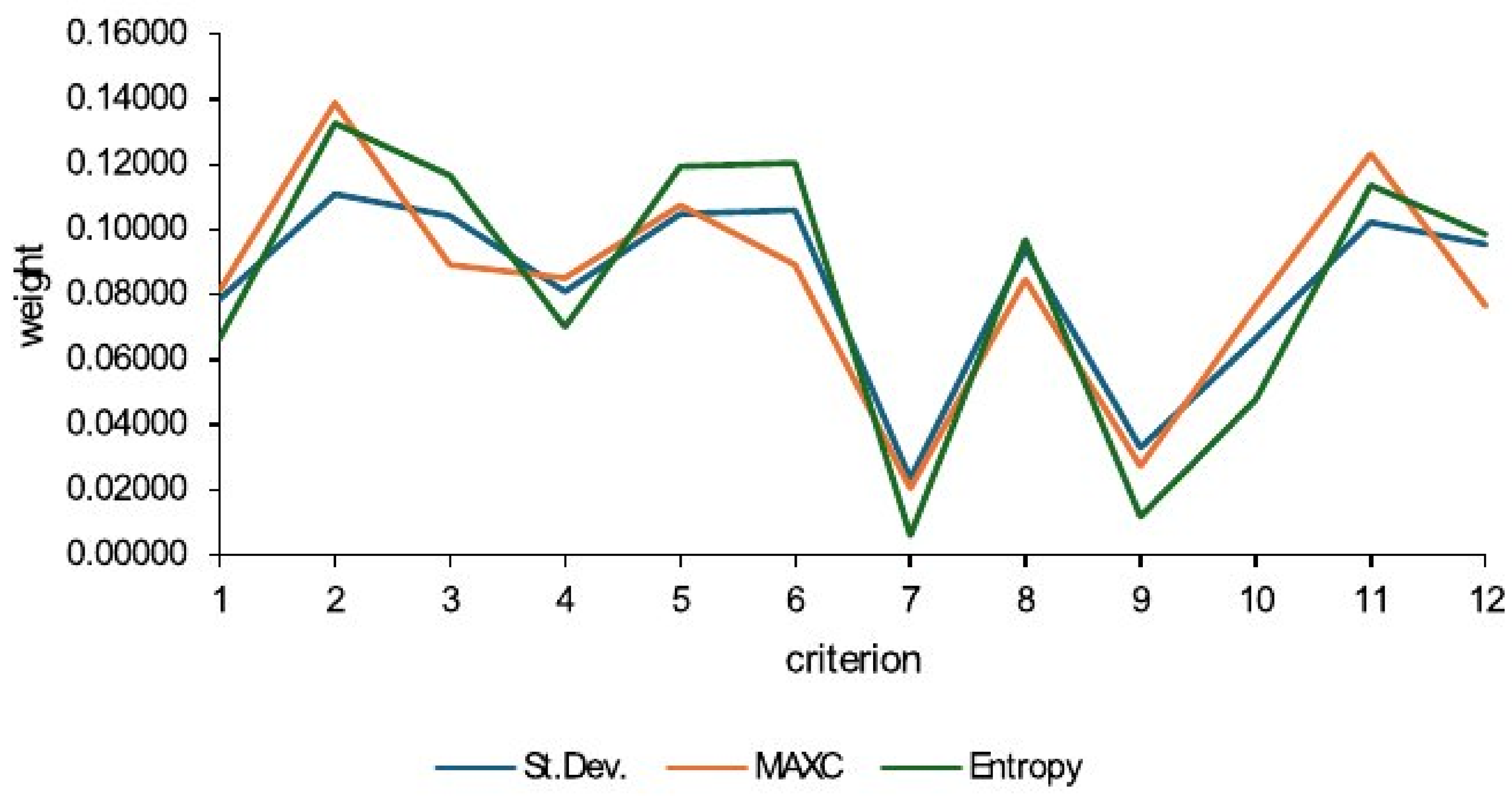

Fig. 5. Weights of criteria obtained by St.Dev., Entropy and MAXC methods.

Table 15
Efficiency indicators.

Comparison	Indicator	
	MAE	PCC
MAXC and St.Dev.	0.01147	0.90222
MAXC and Entropy	0.01749	0.89276
Average value	0.01448	0.89749

4.2. *Application of TODIFFA method for evaluation of alternatives*

Upon calculating the criteria weights, we ranked the Deep Learning Tools. Analyzing the decision matrix, we can see the existence of both target values, maximum and minimum, respectively. Accordingly, the precondition of binomial decision-making space is met, and there is no need to transform any criterion into its reciprocal value. Therefore, the new TODIFFA method can be applied to the Deep Learning Tools ranking.

Step 1:

Preference of criteria significantly influences the rank of alternatives, so it is necessary to weigh the normalized decision matrix. Weighing the normalized decision matrix is performed according to equation (13), and outcomes are shown in Table 20.

Based on expression (13), we get the weighted normalized element A1C1 as follows:

$$q_{11} = 0.11017 \bullet 0.05581 = 0.00615$$

Step 2:

An optimal alternative is created in accordance with the desired targets, and the application of expression (16) enables it. The values of an optimal alternative concerning a given set of criteria and targets are presented in Table 21.

Based on expression (16), we can define the optimal alternative for criterion C1. Since the desired value of criterion C1 is maximum, then the value of an optimal alternative concerning C1 is as follows:

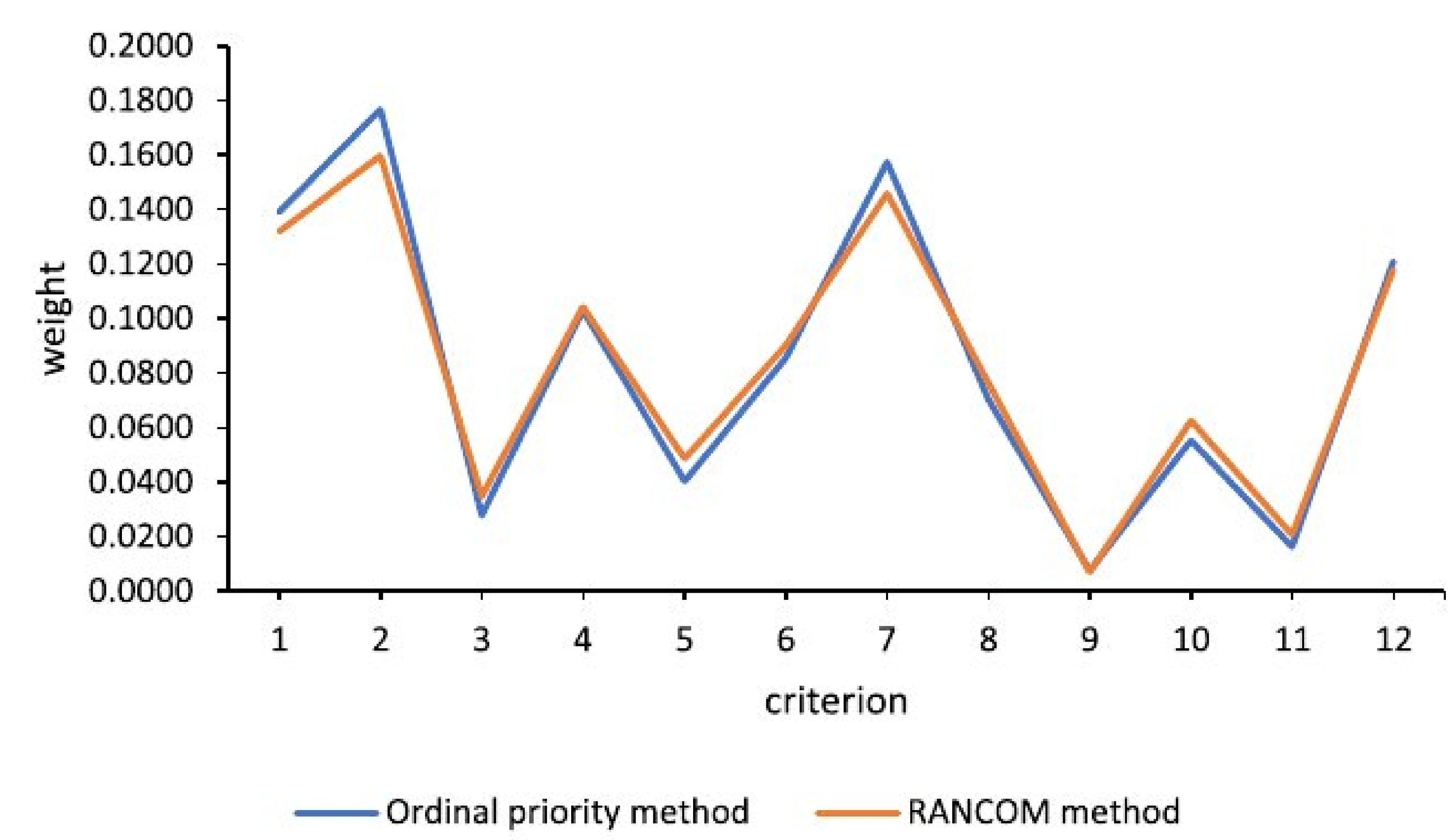

Fig. 6. Subjective weights of criteria obtained by Ordinal priority method and RANCOM method.

Table 16
Standardized expert assessments.

Crit.	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
SEA	0.410	0.390	0.640	0.432	0.593	0.485	0.400	0.485	1.000	0.500	0.842	0.410

Table 17
Subjective weight coefficients of criteria.

Crit.	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Weight	0.13903	0.17658	0.02780	0.10287	0.04031	0.08571	0.15757	0.07042	0.00741	0.05513	0.01622	0.12095

Table 18
Criteria weights obtained by two compared subjective weighting methods.

Method	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Ordinal priority	0.13903	0.17658	0.02780	0.10287	0.04031	0.08571	0.15757	0.07042	0.00741	0.05513	0.01622	0.12095
RANCOM	0.13194	0.15972	0.03472	0.10417	0.04861	0.09028	0.14583	0.07639	0.00694	0.06250	0.02083	0.11806

Table 19
Aggregated weight coefficients of criteria.

Crit.	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Weight	0.11017	0.15776	0.05845	0.09402	0.07389	0.08740	0.08903	0.07758	0.01735	0.06584	0.06976	0.09875

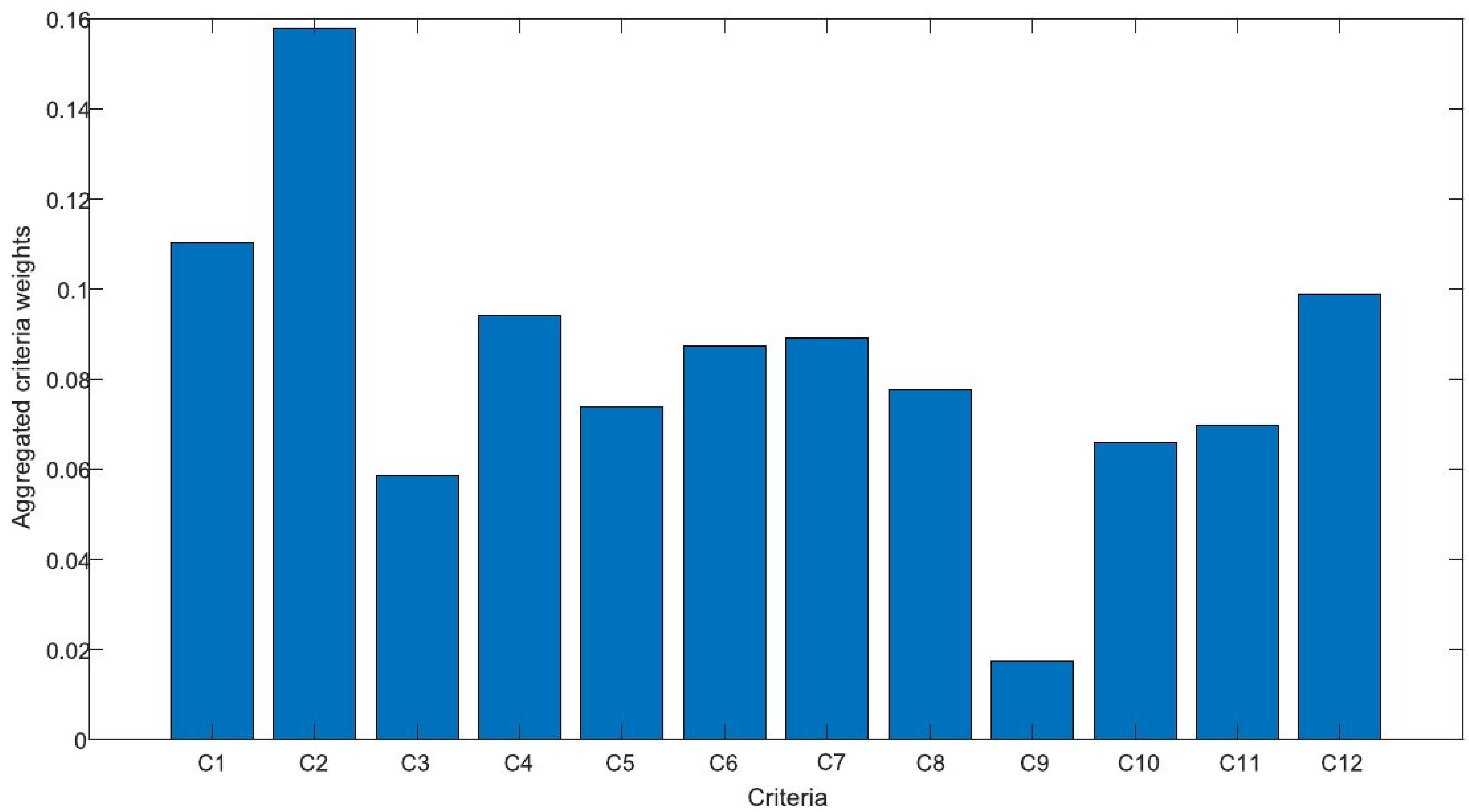

Fig. 7. Aggregated values of weight coefficients.

Table 20
Weighted normalized data.

A-C *target*	C1 *max*	C2 *max*	C3 *max*	C4 *max*	C5 *max*	C6 *max*	C7 *min*	C8 *max*	C9 *min*	C10 *max*	C11 *max*	C12 *max*
A1	0.00615	0.00804	0.00339	0.00471	0.00379	0.00480	0.00473	0.00394	0.00088	0.00325	0.00370	0.00532
A2	0.00515	0.00804	0.00312	0.00442	0.00379	0.00467	0.00473	0.00394	0.00088	0.00306	0.00348	0.00532
A3	0.00582	0.00908	0.00277	0.00499	0.00427	0.00414	0.00460	0.00445	0.00088	0.00345	0.00393	0.00458
A4	0.00582	0.00908	0.00312	0.00499	0.00427	0.00467	0.00460	0.00445	0.00093	0.00345	0.00393	0.00517
A5	0.00532	0.00778	0.00286	0.00457	0.00366	0.00427	0.00486	0.00382	0.00090	0.00345	0.00348	0.00517
A6	0.00515	0.00908	0.00268	0.00442	0.00427	0.00400	0.00473	0.00445	0.00096	0.00325	0.00393	0.00458
A7	0.00665	0.00830	0.00312	0.00556	0.00391	0.00467	0.00486	0.00420	0.00093	0.00394	0.00393	0.00517
A8	0.00515	0.00778	0.00286	0.00442	0.00366	0.00427	0.00473	0.00382	0.00090	0.00306	0.00336	0.00517
A9	0.00582	0.00804	0.00312	0.00485	0.00379	0.00467	0.00486	0.00394	0.00093	0.00345	0.00370	0.00517
A10	0.00582	0.00804	0.00268	0.00499	0.00379	0.00400	0.00460	0.00394	0.00096	0.00345	0.00370	0.00458
A11	0.00532	0.00701	0.00268	0.00457	0.00330	0.00400	0.00486	0.00343	0.00096	0.00345	0.00314	0.00443
A12	0.00582	0.00701	0.00312	0.00499	0.00330	0.00467	0.00460	0.00382	0.00093	0.00345	0.00336	0.00517
A13	0.00582	0.00804	0.00357	0.00499	0.00379	0.00534	0.00460	0.00394	0.00090	0.00355	0.00348	0.00591
A14	0.00582	0.00675	0.00286	0.00499	0.00318	0.00427	0.00460	0.00331	0.00090	0.00355	0.00292	0.00517
A15	0.00582	0.00908	0.00357	0.00499	0.00427	0.00534	0.00460	0.00445	0.00090	0.00345	0.00393	0.00591
A16	0.00582	0.00908	0.00277	0.00499	0.00427	0.00414	0.00460	0.00445	0.00093	0.00355	0.00393	0.00458
A17	0.00665	0.00908	0.00357	0.00571	0.00427	0.00534	0.00460	0.00445	0.00090	0.00394	0.00404	0.00591
A18	0.00631	0.01038	0.00339	0.00571	0.00464	0.00534	0.00460	0.00471	0.00090	0.00355	0.00449	0.00591
A19	0.00598	0.00804	0.00321	0.00514	0.00366	0.00480	0.00473	0.00407	0.00085	0.00355	0.00336	0.00547

Table 21
Optimal alternative values.

A-C *target*	C1 *max*	C2 *max*	C3 *max*	C4 *max*	C5 *max*	C6 *max*	C7 *min*	C8 *max*	C9 *min*	C10 *max*	C11 *max*	C12 *max*
Opt.	0.00665	0.01038	0.00357	0.00571	0.00464	0.00534	0.00460	0.00471	0.00085	0.00394	0.00449	0.00591

Opt. – optimal.

$$f_1^{opt} = max(0.00615, 0.00515, 0.00582, 0.00582, \cdots, 0.00665, 0.00631, 0.00598) = 0.00665$$

Optimal alternatives for the other criteria can be calculated similarly.

Step 3:

In our numerical example (case study), twelve criteria exist, of which two cost and ten benefit criteria exist. Criteria C7 and C9 are the cost criteria, while the rest are the benefit criteria. Partition of the given set of twelve criteria into two subsets is done according to the desired targets of criteria. Partition of the optimal and alternative A1, applying equations (18) and (20), are presented in the following way:

$$F^{opt} = [0.00460, 0.00085] \cup [0.00665, 0.01038, 0.00357, 0.00571, 0.00464, 0.00534, 0.00471, 0.00394, 0.00449, 0.00591]$$

$$F^1 = [0.00473, 0.00088] \cup [0.00615, 0.00804, 0.00339, 0.00471, 0.00379, 0.00480, 0.00394, 0.00325, 0.00370, 0.00532]$$

Step 4:

Summing up the cost elements generates the magnitude of the cost component of an alternative, while summing up the benefit elements generates the magnitude of the benefit component of an alternative. For example, the cost and benefit component magnitudes of the optimal alternative and alternative A1 equal:

$$M_c^{opt} = 0.00460 + 0.00085 = 0.00545$$
$$M_b^{opt} = 0.00665 + 0.01038 + 0.00357 + \cdots + 0.00394 + 0.00449 + 0.00591 = 0.05533$$
$$M_c^1 = 0.00473 + 0.00088 = 0.00561$$
$$M_b^1 = 0.00615 + 0.00804 + 0.00339 + \cdots + 0.00325 + 0.00370 + 0.00532 = 0.04710$$

The other values can be calculated similarly. Values of the component magnitudes are calculated by expressions (21)-(24). The magnitudes of extracted cost and benefit components of all alternatives are presented in Table 22. Positions of alternatives are also presented in Table 22.

Applying equation (30), we calculate the total differential when alternative A1 for position $P_1(0.00561, 0.04710)$ and position $P_{opt}(0.00545, 0.05533)$ as follows:

$$dz_1 = \frac{0.00561}{\sqrt{0.00561^2 + 0.04710^2}}(0.00561 - 0.00545) + \frac{0.04710}{\sqrt{0.00561^2 + 0.04710^2}}(0.05533 - 0.04710) = 0.00815$$

The total differentials of all alternatives are shown in Table 23.

For easier viewing of the results from Table 23, the total differentials of alternatives are presented graphically in Fig. 8.

Step 5:

The ascending order of the obtained total differential values of alternatives produces the final rank of alternatives, as presented in Table 24.

We can select A18 and A17 as the best alternatives based on the defined rank.

5. Sensitivity analysis and comparisons of the results

The following section presents the sensitivity analysis of the TODIFFA model and the validation of the results through comparison with other MCDM techniques. As part of the sensitivity analysis, the influence of objective and subjective values of the weighting coefficients on the model results was considered. The results were validated by comparing the initial results of the TODIFFA model (Fig. 8 and Table 24) with the selected multi-criteria techniques.

5.1. Analysis of the sensitivity of the model to the change of parameter α

This study used two approaches to define the weighted coefficients of the criteria. The first approach is presented through the objective definition of weight coefficients using the MAXC method. In the second approach, the weighting coefficients were defined by applying the OPA linear model in which the subjective preferences of experts were used.

Since applying the MAXC and OPA models defined two vectors of weight coefficients of the criteria, the aggregation function (11) was used to calculate the aggregated weight vector. In expression (11), the parameter α is used, which varies in the interval [0,1]. In the initial results, the value $\alpha = 0.5$ was adopted, which simulated the equal influence of both vectors of weighting coefficients.

Numerous studies have shown that the variation of the α parameter, i.e., the variation of the influence of objective and subjective weighting coefficients, can influence the change in the initial results (Lee and Kang, 2019; Torkayesh et al., 2021; Wen et al., 2021). For example, Lee and Kang (Lee and Kang, 2019) used a variation of the parameter α in a study in which the evaluation of airline service quality was presented. To define the initial results, Lee and Kang (Lee and Kang, 2019) adopted the value $\alpha = 0.5$, while within the sensitivity analysis, they presented an analysis of the influence of other values of α on the change of the initial results. Torkayesh et al. (Torkayesh et al., 2021) presented the application of objective and subjective weight coefficients of criteria for comparative assessment of social sustainability performance. Within the sensitivity analysis, they showed that the variation of the parameter α can lead to a change in the initial results, which can indicate the necessity of additional analysis of the proposed solution. Wen et al. (2021) used an objective-subjective approach to define the weighting coefficients of the criteria within the framework of risk assessment. The

Table 22
The magnitudes of cost and benefit components and alternative positions.

Alternative	Cost component $M_c^{opt}M_c^i, i = 1, 2, \cdots, 19$	Benefit component $M_b^{opt}M_b^i, i = 1, 2, \cdots, 19$	Position of alternative $P_{opt}(M_c^{opt}, M_b^{opt})P_i(M_c^i, M_b^i), i = 1, 2, \cdots, 19$
Optimal	0.00545	0.05533	(0.00545, 0.055533)
A1	0.00561	0.04710	(0.00561, 0.04710)
A2	0.00561	0.04499	(0.00561, 0.04499)
A3	0.00547	0.04748	(0.00547, 0.04748)
A4	0.00553	0.04896	(0.00553, 0.04896)
A5	0.00576	0.04437	(0.00576, 0.04437)
A6	0.00569	0.04582	(0.00569, 0.04582)
A7	0.00579	0.04945	(0.00579, 0.04945)
A8	0.00563	0.04356	(0.00563, 0.04356)
A9	0.00579	0.04656	(0.00579, 0.04656)
A10	0.00555	0.04500	(0.00555, 0.04500)
A11	0.00582	0.04133	(0.00582, 0.04133)
A12	0.00553	0.04471	(0.00553, 0.04471)
A13	0.00550	0.04843	(0.00550, 0.04843)
A14	0.00550	0.04280	(0.00550, 0.04280)
A15	0.00550	0.05081	(0.00550, 0.05081)
A16	0.00553	0.04758	(0.00553, 0.04758)
A17	0.00550	0.05296	(0.00550, 0.05296)
A18	0.00550	0.05442	(0.00550, 0.05442)
A19	0.00558	0.04729	(0.00558, 0.04729)

Table 23
The total differentials of alternatives.

Alternative	The total differential	Value
A1	dz_1	0.00815
A2	dz_2	0.01024
A3	dz_3	0.00780
A4	dz_4	0.00632
A5	dz_5	0.01082
A6	dz_6	0.00940
A7	dz_7	0.00579
A8	dz_8	0.01165
A9	dz_9	0.00866
A10	dz_{10}	0.01024
A11	dz_{11}	0.01381
A12	dz_{12}	0.01053
A13	dz_{13}	0.00685
A14	dz_{14}	0.01242
A15	dz_{15}	0.00448
A16	dz_{16}	0.00769
A17	dz_{17}	0.00235
A18	dz_{18}	0.00090
A19	dz_{19}	0.00796

research showed that the variation of the parameter α can lead to a variation of the defined risk level within the FMEA model. A similar connection between the initial results and the α parameter was pointed out by Zhao et al. (2023) and Chang (2024).

Variating parameter in the interval $0 \le \alpha \le 1$ changes the influence of objective and subjective weighting coefficients in the final decision. The initial results were obtained for the value $\alpha = 0.5$, which simulated an equal influence of objective and subjective factors on the final decision. However, changing the value of α in the interval $0 \le \alpha < 0.5$ increases the influence of subjective values (ϖ_j), while for values $0.5 < \alpha \le 1$, the influence of objective values (δ_j) increases, Fig. 9.

For the values $\alpha = 0$, we get that $w_j = \varpi_j$, that is, the final values of the weighting coefficients are equal to the subjective values. Also, for the value $\alpha = 1$, we obtain that $w_j = \delta_j$, that is, we obtain the objective weight coefficients obtained by the MAXC method. Fig. 9 shows the variations of the weighting coefficients depending on the change in the parameter α. Thirty-four scenarios were generated, so in the first scenario, the value $\alpha = 0.00$ was adopted, while in each subsequent one, the value α was increased by 0.03. Newly generated vectors of weighting coefficients from Fig. 9 were used to calculate new values of total differentials of alternatives, Fig. 10.

The results from Fig. 10 show that the TODIFFA model is sensitive to the change in the weight coefficients of the criteria. However, most of the considered alternatives kept their ranks despite the changes in the significance of the criteria. The statistical analysis determined that the changes from Fig. 10 do not affect the change of the final decision related to the choice of the optimal alternative in the considered problem. Fig. 11 shows the statistical dependence of changes during the 34 scenarios considered.

The Spearman correlation coefficient was used to show the statistical correlation. The results from Fig. 11 show minimal deviations during the scenarios ranging between 0.996 and 1.00. Such results show that the dominant alternatives, which include the first three ranked, have sufficient dominance concerning the other alternatives. Also, we can conclude that the initial results are credible and stable. In addition, based on the presented analysis, we can conclude that the TODIFFA model has adequate sensitivity to changes in subjectively defined input parameters.

5.2. Comparison with other MCDM techniques

The following part compares the TODIFFA model with other multi-criteria techniques from the literature. Multi-criteria techniques using different normalization procedures were chosen for comparison: Multi-Attributive Border Approximation Area Comparison (MABAC) method (Pamučar and Ćirović, 2015) – applies max–min normalization; Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method (Hwang and Yoon, 1981) – applies vector normalization; Weighted Aggregated Sum Product ASessment (WASPAS) method (Zavadskas et al., 2012) – applies additive normalization; COmpressed PRoportional ASsessment – (COPRAS) (Zavadskas et al., 2004) – applies max linear normalization. Fig. 12 presents the results of the comparison of the mentioned MCDM techniques.

To define the statistical dependence of the results from Fig. 12, Spearman's correlation coefficient (SCC) was applied:

$$SCC = \begin{array}{cccccc} M/M & M1 & M2 & M3 & M4 & M5 \\ M1 & 1.000 & 0.979 & 0.958 & 1.000 & 1.000 \\ M2 & 0.979 & 1.000 & 0.904 & 0.979 & 0.979 \\ M3 & 0.958 & 0.904 & 1.000 & 0.958 & 0.958 \\ M4 & 1.000 & 0.979 & 0.958 & 1.000 & 1.000 \\ M5 & 1.000 & 0.979 & 0.958 & 1.000 & 1.000 \end{array}$$

where M1 – TODIFFA, M2 – TOPSIS, M3 – MABAC, M4 – WASPAS and M5 – COPRAS.

From the statistical analysis, obtained values of SCC between each MCDM method belong to a very high level of relationship. TODIFFA

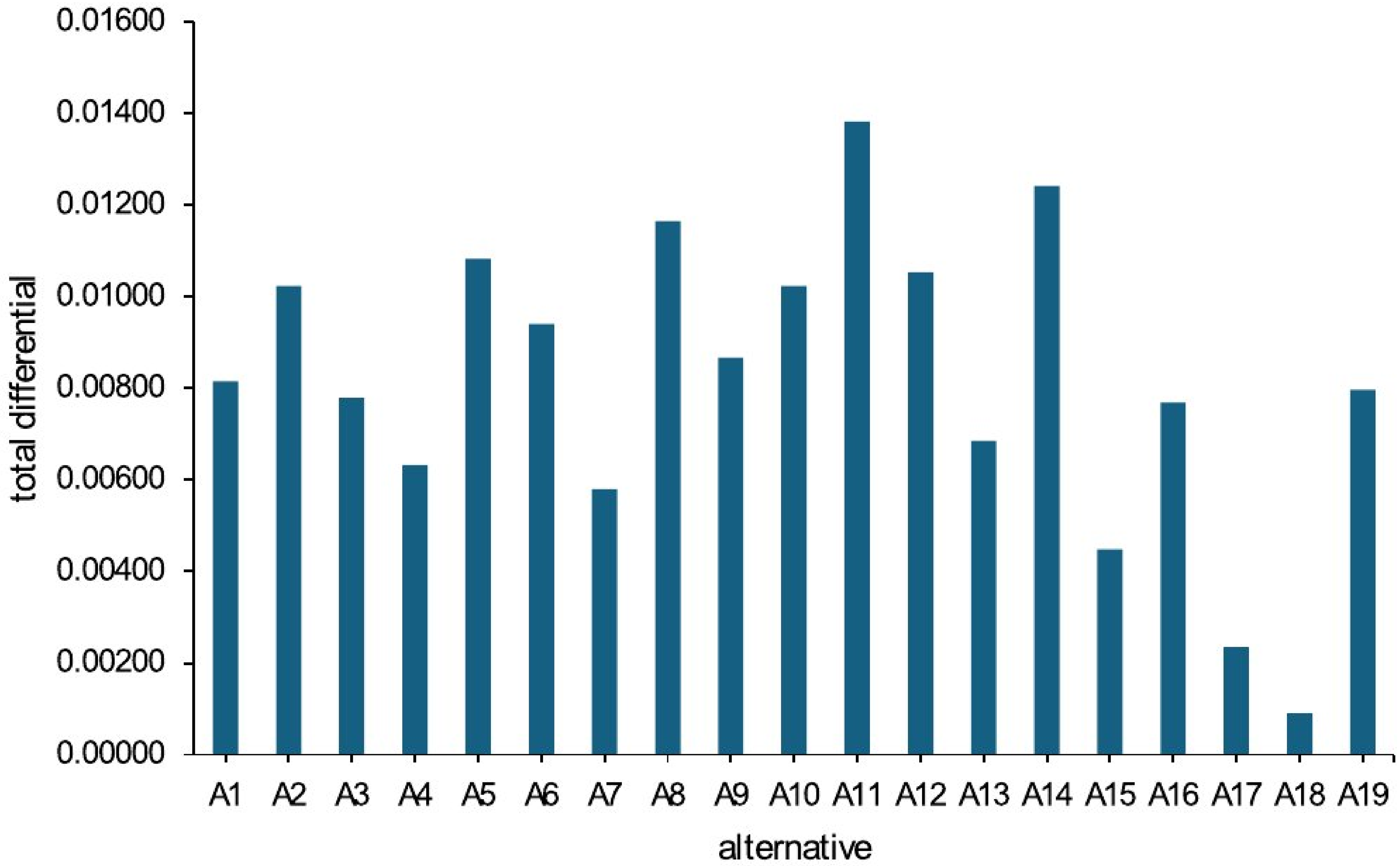

Fig. 8. Plot of the total differentials of alternatives.

Table 24
Rank of Deep Learning Tools (Alternatives).

Codes	Deep Learning Tool	Rank
A1	Google Cloud Deep Learning Containers	10
A2	Microsoft Cognitive Toolkit	14
A3	Neuton AutoML	8
A4	Knet	5
A5	NVIDIA Deep Learning	16
A6	Swift AI	12
A7	Theano	4
A8	Chainer	17
A9	Clarifai	11
A10	Coffee	13
A11	DeepPy	19
A12	Bitnami Pytorch	15
A13	Neon	6
A14	Neuroph	18
A15	Torch	3
A16	AWS Deep Learning AMIs	7
A17	CNTK	2
A18	TensorFlow	1
A19	Keras	9

method is extremely correlated with WASPAS (1.000) and COPRAS (1.000). Slightly less, but still very high correlation TODIFFA achieves with two others compared MCDM methods, TOPSIS (0.979) and MABAC (0.958). It is a clear indicator that this developed methodology stays side by side with other compared methods and presents a powerful tool for solving various MCDM problems. Also, average SCC value of TODIFFA method (0.984) is equivalent to the average SCC values of WASPAS (0.984) and COPRAS (0.984) methods, while TOPSIS (0.960) and MABAC (0.945) methods have a slightly lower average SCC value. Accordingly, TODIFFA method is absolutely acceptable and applicable mechanism to solve such complex MCDM problems.

The SCC shows that the results of all models are highly correlated. Based on the results, we can conclude that the WASPAS, COPRAS and proposed TODIFFA methods are fully correlated, while other models have minimal deviations. Also, the results show that the analysed MCMD techniques are mutually correlated more than 95 %, which indicates the validity of the results of the TODIFFA model.

The performances of the TODIFFA method are listed as follows: data type is quantitative, the average correlation coefficient is 0.984, transparency is good, the complexity of the method is less, consumption time is low, and mathematical calculus required is low. The TODIFFA method for alternative ranking is a reliable tool that is easy to understand and implement for solving any multi-criteria decision-making problem.

The superiority of the proposed TODIFFA method and its performance evaluation compared to other MCDM techniques is explained in Table 25.

One of the important factors for performance evaluation of an MCDM model is the possibility of including or excluding the number of alternatives in the decision-making process. The capability for making the consistent rank of alternatives under conditions of variation in the number of alternatives actually represents the adequacy to changes of alternatives or in other words rank reversal problem of the MCDM methodology. TOPSIS and COPRAS methods are very sensitive to variations in the number of alternatives. MABAC and WASPAS methods are moderately sensitive to a rank reversal while the proposed TODIFFA methodology is the most stable under conditions of changing the number of alternatives.

Similarly to the previous factor, there is an attribute that deals with the possibility of changing the values of some criteria during the decision-making process. MCDM models should be resistant to this deviation producing consistent results of alternatives ranking. TOPSIS and COPRAS methods are sensitive to varying the values of criteria while MABAC and WASPAS methods are little less sensitive to changing the values of criteria. TODIFFA method shows stable and credible results of the alternative ranking under conditions of changing the values of criteria.

Complexity and computation time are closely related factors that describe performance evaluation of the MCDM models. If the MCDM model is complex, it means that this model is easy to understand and requires more computation time and vice versa. Developed TODIFFA method, MABAC and TOPSIS methods are simple and easy to use. Accordingly, these models do not require more and complex mathematical computation. On the other side, WASPAS and COPRAS methods have a slightly higher complexity and computation time. However, these shortcomings of the mentioned methods can be effectively eliminated by creating software which would significantly speed up the input data processing.

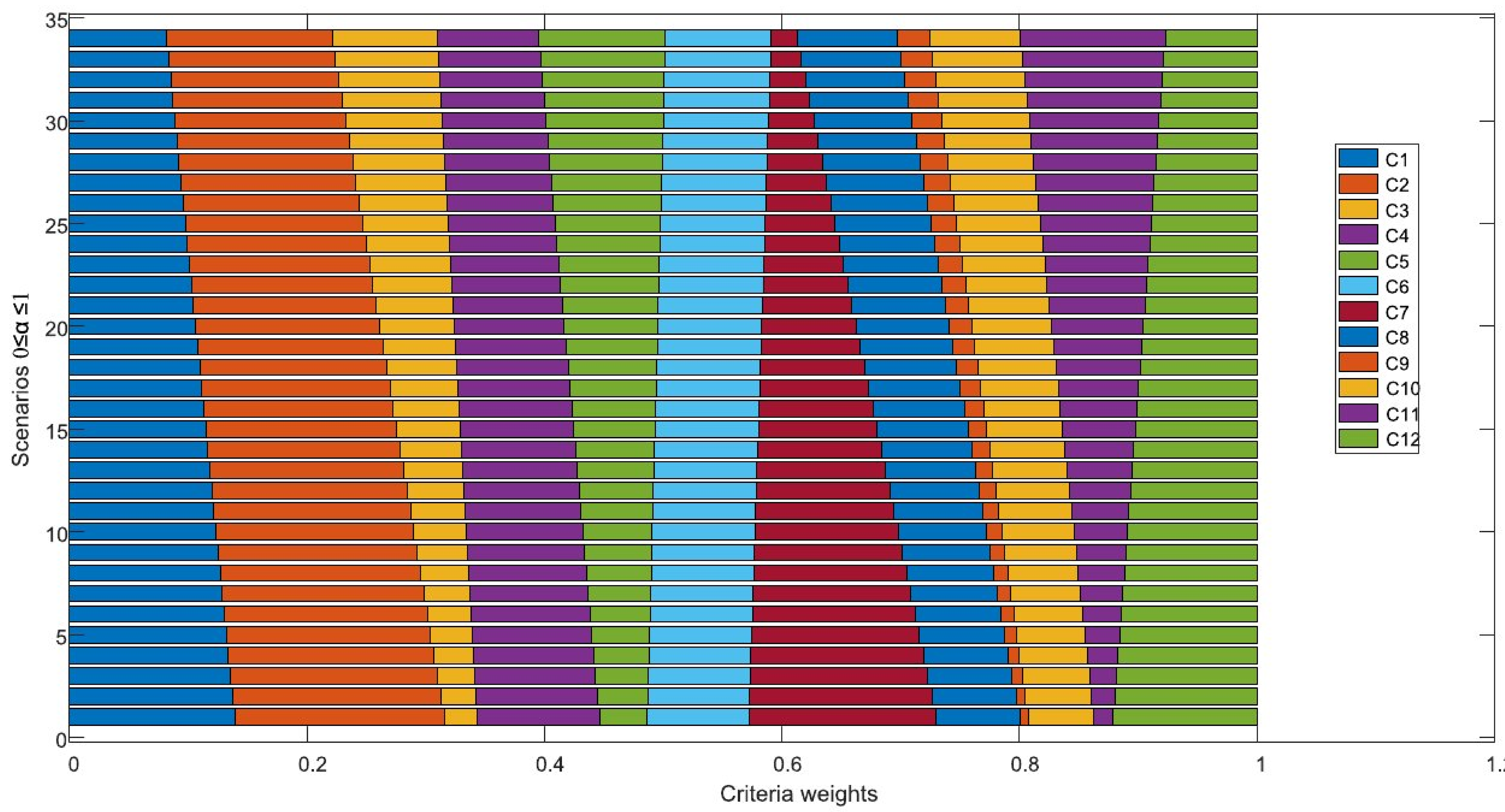

Fig. 9. Dependence of the final weight coefficients of the criteria on α.

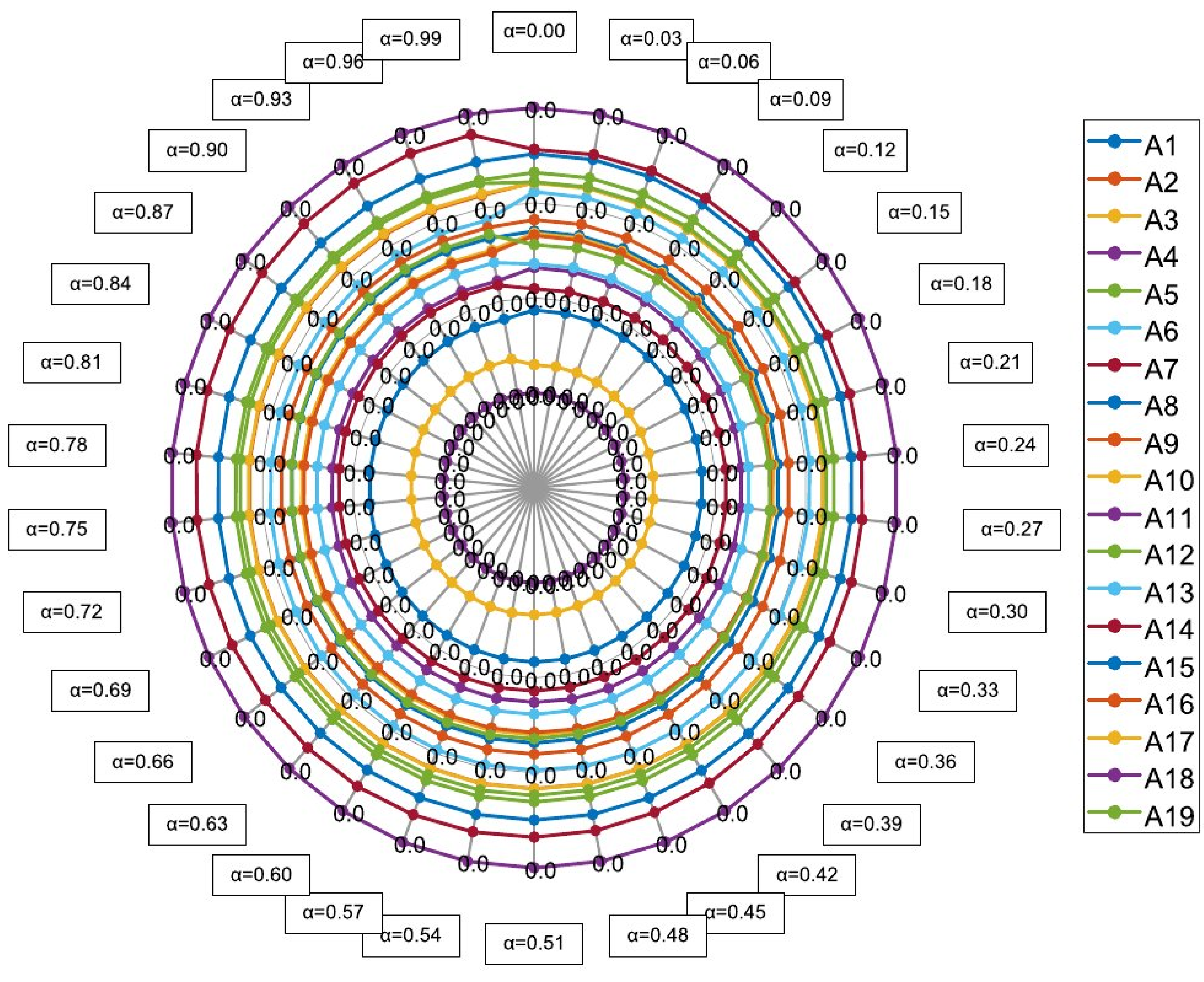

Fig. 10. New values of total differentials of alternatives.

6. Results and discussions

Deep learning is still maturing, and development is essential to the highly disruptive digital transformation that has emerged in the Industry 4.0 process. In many industries, including the automotive industry, the idea of developing more innovative business models, products, and applications to survive in an increasingly competitive environment is becoming increasingly widespread. In the digital transformation, practitioners consider DL architectures and applications a unique opportunity to design more intelligent systems. Accordingly, numerous global technology companies focus on DL applications and make significant investments. In contrast, the satisfactory and successful performance of any DL platform in every industry may not be possible due to the industries' specific circumstances, expectations and different business models.

The disruptive digital technologies that emerged in the industry 4.0 process are changing the automotive industry and the usual business models. In the context of Industry 4.0, new job opportunities are

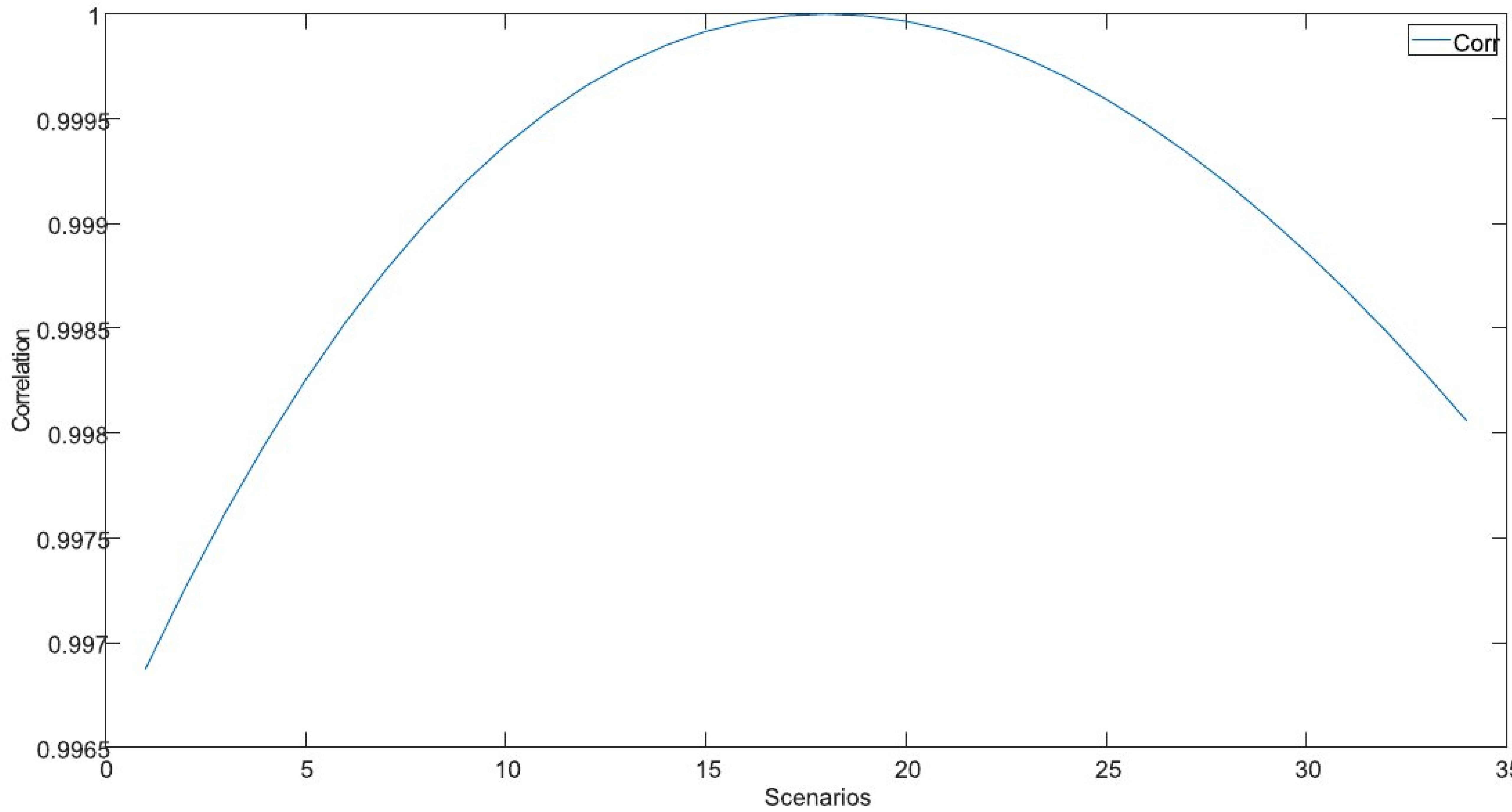

Fig. 11. Statistical correlation of ranks.

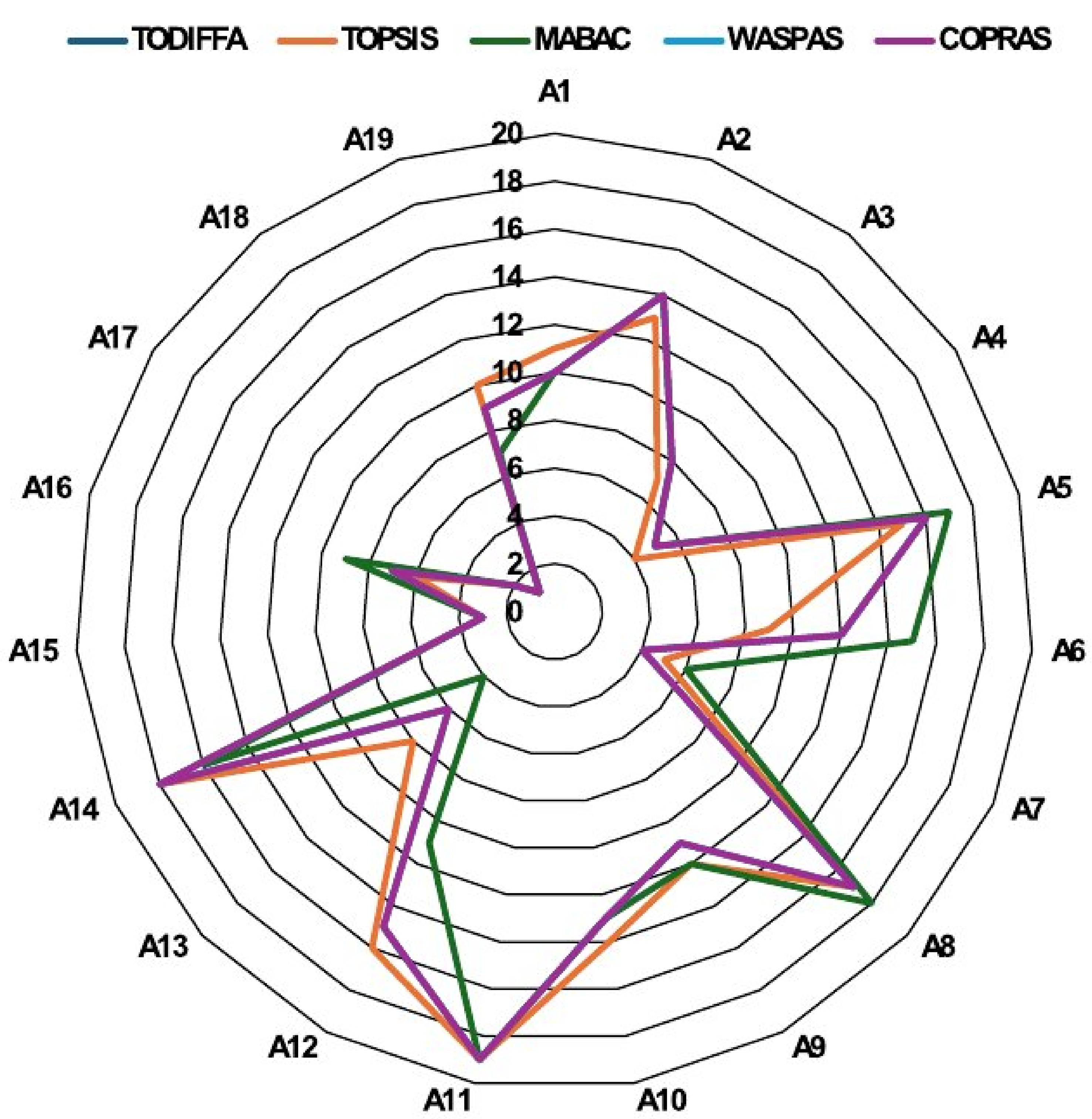

Fig. 12. Comparison results of different MCDM techniques.

emerging, so companies need to adapt to the new conditions (Llopis-Albert et al., 2021). From this point of view, deep learning applications can enable the processing of information and data, which is large-scale, diverse, fast, and constantly increasing in the automotive industry. They can resolve issues arising from existing business models exceptionally quickly. Deep learning methods may improve manufacturing processes apart from maintenance and repair applications in the automotive industry.

Deep learning models can formulate sophisticated service plans by leveraging historical service data. Furthermore, customized applications tailored to each tool can be developed using deep learning models. Deep learning applications can help to deliver more efficient and expert services by integrating variables such as weather patterns, road conditions, driver behaviour (e.g., excessive braking), and traffic congestion into the analytical processes. They can also anticipate future needs, such as identifying components requiring replacement in advance, implementing specialized service interventions, and estimating the necessary time for completion.

Deep learning platforms can enhance the organization of operations such as service, maintenance, and repair, leading to greater effectiveness, efficiency, and performance. Furthermore, they facilitate the restructuring and effective management of automotive supply chains. Specifically, they enable the development of high-quality solutions, such as directing recalled vehicles to the appropriate warehouses and sales points.

The deep learning technologies in areas such as quality, maintenance and repair has extremely important positive effects and advantages. Considering that the cost of stopping an automobile production line for just one hour is $1.3 million (Sharma et al., 2021), the potential of deep learning models to mitigate such risks is very promising. At the same time, automotive companies may suffer severe financial and reputational losses in cases such as unexpected situations, vehicle recalls, production defects that cannot be tolerated by clients.

Deep learning technology is widely used in many online and mobile services, such as Siri and Google Assistant, voice recognition and dialogue systems like Amazon's Alexa and Microsoft Cortana, and image classification systems in Google Photo and Facebook (Luckow et al., 2016). Deep learning has many potential applications in the automotive industry in autonomous driving and robotics, computer vision, improvements in manufacturing processes (e.g., monitoring quality issues), and connected vehicles, infotainment services (e.g., voice recognition systems).

The infrastructure and tooling landscape for training and deploying deep neural networks are evolving rapidly. While deep learning applications share similarities with traditional big data systems, Deep Neural Networks (DNNs) pose unique challenges in terms of training and

Table 25
Performance evaluation of the proposed TODIFFA method and comparison with MCMD methodologies.

MCDM methodology	Adequacy to changes of alternatives	Adequacy to changes of criteria	Complexity	Mathematical calculation required
MABAC	Moderate	Moderate	Low	Low
TOPSIS	Low	Low	Low	Low
WASPAS	Moderate	Moderate	Moderate	Moderate
COPRAS	Low	Low	Moderate	Moderate
TODIFFA (proposed)	High	High	Low	Low

scalability due to the scale of the data and models involved (Singh and Arat, 2019). Unlike simpler models, deep learning necessitates video, image, or text data for training, as well as millions of parameters and extensive datasets. Training such models demands scalable storage, distributed processing capabilities, compute resources, and accelerators.

Furthermore, deploying these models presents challenges concerning their integration into mobile devices. Minimizing the number of parameters and the volume of input data required for efficient operation is imperative. Modern convolutional neural networks, for example, often demand billions of operations for a single inference. Hence, optimizing the deployment process to ensure efficient operation on mobile devices is crucial.

Notwithstanding these obstacles, Deep Learning techniques find application across a spectrum of scenarios within the automotive sector. Notably, advancements in computer vision within Deep Learning systems have been remarkable in recent times. The proliferation of mobile devices and Internet of Things (IoT) sensors has resulted in a substantial surge in image and video data, often unstructured (El et al., 2021). Deep learning methodologies can aid in the organization of this data and enhance data acquisition procedures. Furthermore, computer vision applications extend to social media analysis, where the abundance of image data shared by the public holds valuable insights. Deep learning plays a pivotal role in enhancing both the collection and analysis of such data, thereby contributing significantly to improved processes in this domain.

Moreover, the realm of Computer Vision encompasses applications that extend to social media analysis. Publicly available tools offer a means to extract valuable insights from image data generated by consumers. Deep learning methodologies are pivotal in augmenting data collection and analysis in this context. Additionally, various facets of autonomous driving necessitate the utilization of machine learning technologies. These technologies are instrumental in processing vast volumes of sensor data, including camera-based sensors and Lidar, and learning driving scenarios and driver behaviours. Conversely, robotics relies heavily on intricate computer vision subsystems. Deep learning techniques identify features within camera images and other sensor data types. While object detection using Deep Neural Networks (DNN) is well-established, it can also be leveraged for more sophisticated tasks such as object tracking. Furthermore, deep learning facilitates the development of brilliant robots capable of autonomous learning throughout their operational lifespan.

Hence, integrating deep learning platforms into business models becomes imperative for processing vast and dynamic information within the automotive sector. Accommodating these systems within the operational frameworks of automotive manufacturers poses several decision-making challenges that require resolution. Among these challenges, the foremost concern for industry practitioners is selecting a suitable deep-learning platform, given the inadequate support from the research community. Consequently, numerous scholars contend that despite their significance, deep learning technologies are still nascent in the automotive industry (Fenn, 2020; Wang et al., 2018; Nikitas et al., 2020; Qi et al., 2020; Chavan, 2024).

In this context, evaluating the existing DL platforms is highly challenging for decision-makers. In addition, selecting the most appropriate one and adapting it to the enterprises' business models is a very complex, time-consuming and risky decision-making problem. A wrong choice can lead to the failure of all operations of the enterprise, the investment made for the DL platform and the waste of resources, monetary losses, and a severe loss of reputation of the enterprise (Hatcher and Yu, 2018). Therefore, for DL architectures to be successfully integrated into the business models of enterprises, decision-makers must take measured steps and determine the appropriate DL platform for the needs of the enterprise as a priority. Choosing the appropriate platform is critical for successfully implementing DL applications in enterprises.

In contrast, no literature supports the encouragement and motivation of decision-makers in the automotive industry to choose DL platforms to integrate into their company's business models. The number of studies evaluating DL platforms is surprisingly scarce, and there is no mathematical model that decision-makers can use as a roadmap in the evaluation process. In these studies, DL platform alternatives were compared within the framework of several criteria, such as speed, performance and GPU usage. Therefore, the analyses in these studies are insufficient to provide sufficient insight to decision-makers in the selection process.

This study developed and proposed a new decision-making model to fill these research gaps. The proposed model provides decision-makers with a highly flexible decision-making environment. In addition, the implementation steps of the model and the algorithm are straightforward and easily applicable without requiring advanced mathematical knowledge. In addition, whether the criteria are maximum or minimum does not affect the results to be obtained. In this respect, the proposed model is not dependent on the direction of the criteria. Besides, the calculation required to sort the alternatives is simple and easy to understand.

TODIFFA-MCDM framework belongs to a group of MCDM methods characterized by simplicity and short time for the computational complexity process. High correlation with compared methods and great potential for involving the uncertain or fuzzy data makes this method very suitable for analysing decision-making problems in different areas. This method is quite stable in the case of solving large-scale problems with no limit on the number of alternatives or criteria.

When the results obtained in the study were reviewed, the C2 Data Availability and Quality criterion was determined as the most influential criterion. The efficiencies and performance of DL platforms depend heavily on quality and accessible data. In practice, this is expressed in the words clean or dirty used for the data. Poor data quality can lead to severe deterioration and deviations in the results obtained using DL applications (McDonald, 2023). From this point of view, it is critical that a DL platform can process data of different quality, characteristics, and structure with satisfactory performance (Mahajan et al., 2022; Gudivada et al., 2017; Zizka et al., 2021). It is not always possible to find clean and high-quality data in industries. Surprisingly and interestingly, in studies trying to compare DL platforms, no emphasis has been placed on the relationship between data quality and accessibility and the performance of DL platforms. This study's finding may provide research motivation for DL platform manufacturing technology companies to develop solutions on how DL platforms can perform at a high level without clean, high quality and sufficient data.

The second important criterion is the C7 Time constraints. This criterion can be related to the concepts of working time, calculation time and speed defined in the studies in the literature. It is an essential factor that DL platforms are not affected by time constraints and perform highly in all conditions. In this context, researchers continue to work on software and hardware developments that will reduce the time constraints of DL architecture and applications. Accordingly, there are successful studies in the literature (Bejnordi et al., 2017; Rudy et al., 2019; Altenmüller et al., 2020; Kim and Ha, 2023). The third influential criterion was determined as the C1 Problem type. DL platforms are configured to solve different learning problems. Others have a more flexible structure and may be able to solve different learning problems such as classification, regression, clustering, density estimation, visualization, and projection (Shi et al., 2019; Azer, 2009; Georgevici and Terblanche, 2019). In this context, the ability of a DL platform to solve problems of different types and qualities is an essential factor in its selection. Finally, the C9 Cost of Installation was determined as the relatively least important criterion. The main reason for this is that almost every DL architecture is an open-source platform, and most of these platforms are either free or offered to end users at meagre fees. It is, therefore, perfectly reasonable for decision-makers not to consider costs as an essential criterion. The remaining criteria have been ranked as C12 RAM size (memory) > C4 Explainability > C6 Capability to handle multiple inputs > C8 Domain knowledge > C10 GPU size (memory) > C5 Scalability > C3 Computational resources > C11 Storage size.

The RAM size is often adjusted to be comparable to a standard accelerator for multimedia processing, such as video encoding (Sze et al., 2013). Consequently, determining the memory capacity to be allocated for a deep learning model is constrained by the same spatial limitations as each data stream requires (Sze et al., 2017). As a result, RAM size is not as important as the other criteria mentioned earlier.

Explainability stands out as a paramount criterion for deep learning platforms in fields like medicine and health sciences. In medical diagnostics, clarity, transparency, and comprehensibility of data derived from imaging technologies are imperative for accurate diagnoses (Singh et al., 2020). While explainability remains crucial for deep learning models directly processing images, as exemplified in the healthcare sector, the data generated from image processing in the automotive industry is comparatively less intricate and uncertain. Although explainability is deemed significant for the automotive sector, it holds less weight than the preceding criteria in the hierarchy of importance.

The capability to handle multiple inputs signifies the proficiency of deep learning platforms in handling various types and characteristics of data derived from various sources. However, having this capability is nearly a prerequisite for end-users and practitioners across almost all deep-learning platforms. For instance, consider a deep learning platform equipped with capabilities for processing text, audio, etc. In such cases, the absence of image processing functionality would be uncommon. While this criterion remains fundamental, its significance is lower than the preceding criteria in the hierarchy of importance.

Like the Explainability criterion, the domain knowledge criterion holds significant importance for deep learning platforms utilized in health sciences. It necessitates predefined domain-specific knowledge to effectively process acquired data, particularly within image recognition applications, to yield meaningful outcomes. Deep learning models ensure dependable results through alignment with domain-specific information. Consequently, fault tolerance in healthcare industry applications employing deep learning models approaches zero. Conversely, while restructuring business models in the automotive industry to align with planned service application development and optimize supply chains and production processes according to deep learning models has an acceptable level of tolerance compared to the healthcare sector, the requisite domain knowledge can also be cultivated subsequently based on the data acquired.

The GPU size (memory) and Storage size criteria can be assessed with a similar perspective to the RAM size criteria. The requirements and capacities of industries and enterprises determine these characteristics of deep-learning platforms. Businesses can increase their capacity for these criteria if data processing at a higher level and speed is required. Therefore, it is impossible to say that higher capacity is better for both criteria.

Scalability is a criterion that allows deep learning platforms to process and train more data per unit of time. This criterion is also relatively important. It is determined by the needs of an industry that develops deep learning applications. For this criterion, it can be said that higher capability does not mean better results. Computational resources represent a significant facet of deep learning, offering potential enhancements to established business models and operations within the automotive sector. This capability is crucial in emergencies, hazardous tasks, and scenarios necessitating rapid decision-making (Sharifi et al., 2021). From this perspective, the proficiency of deep learning models is paramount, especially for rescue robots deployed during accidents during the transportation or manufacturing of hazardous materials. Conversely, the need for urgent decision-making in routine automotive operations is uncommon, except in autonomous vehicle technology.

Ultimately, the suggested installation expenses proposed by deep learning platforms are notably modest and reasonable in influencing decision-making. A preliminary investigation reveals that these platforms are available to consumers at a range of costs, from $10,500 to $85,000, contingent upon the attributes and scale of the data requiring processing (HackerNoon, 2024).

When the ranking performances of DL platform alternatives acquired by implementing the suggested model are evaluated, A18 TensorFlow, A17 CNTK, and A15 Torch are the DL platforms in the top three. The results are similar to the general results of studies comparing DL platforms in the literature. TensorFlow was identified as the best option in three of the ten studies comparing DL platforms in the literature. Al-Bdour et al. (2020) In an experimental study to compare Caffe, Neon, TensorFlow, Theano, and Torch DL platforms, TensorFlow found that TensorFlow is significantly faster in terms of processing time compared to other alternatives. Especially in autonomous vehicles (Self-Driving Car) developed in the automotive industry, quickly analyze the images collected from the external environment and convert them into behaviour. TensorFlow seems to be an option with unique advantages and advantages depending on its processing speed. Bahrampour et al. (2016) have argued that TensorFlow is a highly flexible platform compared to other DL alternatives. In their experimental study, Pham (2021) found that TensorFlow's error rate is meagre compared to other DL platform alternatives, while its accuracy is relatively high. Ultimately, TensorFlow has the most extensive and vigorous developer community compared to any other DL platform (Rao, 2023). Accordingly, this DL platform is exceptionally advantageous in terms of being able to debug and develop in a short time. Despite these advantages, TensorFlow also has some disadvantages. First, it requires more memory usage than many competing DL platforms, and excessive use of GP and memory results in higher energy consumption (Mohajer et al., 2022). On the contrary, optimizing UL/DL Reserved NOMA heterogeneous networks can lead to a substantial reduction in energy consumption (Dong et al., 2023). Furthermore, the service computing optimization model, introduced by Mohajer et al. (2023), seeks to enhance energy efficiency while ensuring the necessary scope and capacity of edge computing networks. Second, its performance on a single GPU is not competitive compared to other DL frames (Hatcher and Yu, 2018). Therefore, when TensorFlow is preferred, the number of GPUs should be increased to achieve the required efficiency and performance. In addition, although TensorFlow provides end users with an easy-to-use interface, it requires more programming knowledge than its competitors (Rao, 2023).

In conclusion, to summarize the managerial implications of the study: (a) The most crucial relative capability of DL platforms is the ability to provide solutions with the highest possible performance even when data quality and availability are low. For example, even in situations that affect image quality, such as rain and fog, it is essential for vehicle, road and driver safety that DL applications used in autonomous vehicles can process these images and convert them into autonomous behaviour. (b) In addition, DL platforms must be capable of calculating and analyzing data at the highest possible speed. In cases where data cannot be processed in real time, DL applications can cause serious problems instead of creating solutions. For example, regarding accident and collision avoidance, the DL platform needs to predict situations that create an accident risk before the accident occurs and develop behaviour accordingly. (c) There is a correlation between the performance of DL platforms and the number and usage of GPUs. Accordingly, even if a suitable platform is preferred, GPU usage may also need to be planned for the best performance of this platform. Accordingly, Luckow et al. (2016) recommend using multi-GPUs to achieve high performance in DL applications in the automotive industry. Finally, although they are seen as infancy, there is a significant motivation in the automotive industry for the impact and benefits of DL applications. This motivation has also begun to be reflected in the studies in the literature. In this context, these studies on the use of DL applications in the automotive industry, evaluation of the latest trends in DL applications in the automotive industry (Luckow et al., 2016), inspection of printing defects on stamped metal surfaces (Block et al., 2021), automated quality inspection for completed vehicles (Rio-Torto et al., 2021), predictive testing strategy (Schoch et al., 2023), software-based testing strategies (Ruospo et al., 2021), driverless cars (Rao and Frtunikj, 2018), preventive maintenance strategies (Theissler et al., 2021), quality control automation (El et al., 2021). Based on the analyses of research comparing various deep learning platforms and assessments published on online platforms (Hatcher and Yu, 2018; Yin, 2024), the findings outlined in Table 26 have been derived.

As shown in Table 26, TensorFlow provide more advantages for all criteria except for cost of installation and GPU size. Its performance can be accepted as satisfactory for these criteria when it compared to the others.

Finally, our sensitivity and comparative analyses to test the robustness and validity of the proposed decision-making model confirm its validity. Despite all the extreme changes and modifications, no significant changes were observed in the ranking results. Accordingly, it can be said that the proposed model is highly consistent, robust and successful in solving complex decision-making problems, depending on the results obtained.

7. Conclusion

In this study, we sought to provide practical insights into deep learning technologies, a hot topic in recent years and of interest to decision-makers in industries and research community members. We have extensively analysed and discussed the evaluation and selection of DL platforms, a topic that has been largely neglected in the literature. In addition, we have developed a convenient and robust decision-making model so that decision-makers can select the most appropriate DL platforms in an evaluation process. The results and outputs we have obtained by applying this model offer efficient implications and valuable theoretical contributions for almost every industry that wants to integrate DL applications into business models and design intelligent systems, especially the automotive industry. These managerial implications can also be a helpful roadmap for technology companies and organizations developing DL platforms to improve their products.

Although it has essential contributions and inferences, this study has some limitations. First, this study was conducted for a large-scale automotive manufacturer in Turkey, and the members of the expert committee were selected from among the experts working in the institutions and organizations in the country. Accordingly, the study has geographical limitations and may be biased on the conditions of the country where the study is carried out and the local requirements of the industry. Therefore, the proposed model can be repeated for industries in different countries, and the results can be compared. In addition, the members of the board of experts may be selected from among experts from different countries. Another limitation was the limited number of studies evaluating DL platforms using multi-criteria decision-making or decision-support systems. That eliminated the possibility of comparing our study with previous studies, making it difficult for us to determine the criteria used in the literature and take them into account. Accordingly, we negotiated with the decision-makers and tried to determine effective criteria within the framework of the opinions and thoughts of the experts.

Furthermore, the model suggested in the current paper can be employed for various industries across different countries, including textiles, construction, and food sectors, apart from the automotive industry. These investigations could incorporate diverse attributes and features specific to each country into the assessment processes. Additionally, the criteria utilized in the evaluation procedures can be reassessed based on the distinct characteristics and needs of the industries and the particular circumstances of the countries involved.

In this context, an experimental investigation was undertaken to assess the feasibility of the proposed model across various industries. To achieve this, four specific sectors were identified: Logistics, Healthcare, Food, and Energy. Accordingly, professional social networking platforms like LinkedIn were utilized to locate experts with relevant experience. Professionals who had prior involvement in these industries or had expertise in developing machine learning or deep learning models for them, assuming roles such as software developers, data managers,

Table 26
Comparisons of the deep learning platforms considering assessment in diverse.

	Problem type	Data Availability and Quality	Computational resources	Explainability	Scalability	Capability to handle multiple inputs	Time constraints	Domain knowledge	Cost of Installation	GPU size (memory)	Storage size	RAM size (memory)
Google Cloud Deep Learning Containers	VH	H	P	H	H	VH	P	VH	P	H	H	VH
Microsoft Cognitive Toolkit	H	H	VH	H	H	VH	P	VH	P	H	H	VH
Neuton AutoML	VH	VH	H	VH	VH	H	P	P	P	VH	VH	H
Knet	VH	VH	VH	VH	VH	VH	P	P	VH	VH	VH	VH
NVIDIA Deep Learning	H	H	H	H	H	H	P	H	VH	VH	H	VH
Swift AI	H	VH	H	H	VH	H	P	P	VH	H	VH	H
Theano	P	H	VH	P	VH	VH	P	VH	VH	P	VH	VH
Chainer	H	H	H	H	H	H	P	H	VH	H	H	VH
Clarifai	VH	H	VH	VH	H	VH	P	VH	VH	VH	H	VH
Coffee	VH	H	H	VH	H	H	P	VH	VH	VH	H	H
DeepPy	H	MH	H	H	MH	H	P	H	VH	VH	MH	H
Bitnami Pytorch	VH	MH	VH	VH	MH	VH	P	H	VH	VH	H	VH
Neon	VH	H	P	VH	H	P	P	VH	VH	VH	H	P
Neuroph	VH	MH	H	VH	MH	H	P	MH	VH	VH	MH	VH
Torch	VH	VH	P	VH	VH	P	P	P	VH	VH	VH	P
AWS Deep Learning AMIs	VH	VH	H	VH	VH	H	P	P	VH	VH	VH	H
CNTK	P	VH	P	P	VH	P	P	P	VH	P	VH	P
TensorFlow	P	P	P	P	P	P	P	P	VH	VH	P	P
Keras	VH	H	VH	VH	H	VH	P	VH	P	VH	H	VH

and project managers, were contacted. Subsequently, five experts from each sector were selected to participate in this experimental study. Table 27 presents the particulars of these experts.

The survey administered to the experts on the analyst board was similarly conducted with these experts. Researchers requested their assessment of the alternatives based on each criterion. Each expert evaluated the alternatives according to the criteria, considering the dynamics and needs of the industry in which they possessed prior experience and expertise. We applied the proposed model to each sector based on these assessments and recomputed the criteria' weight coefficients and the alternatives' preference levels. Initially, the proposed model was applied repeatedly across five industries, encompassing the automotive sector, resulting in the ranking outcomes for these industries, as displayed in Table 28.

As depicted in Table 28, slight variations were observed in the rankings of the A4 Knet, A7 Theano, A9 Clarifai, and A12 Bitnami Pytorch platforms compared to the assessment conducted for the automotive industry. These alterations occurred during evaluations for the health and energy sectors, while no alterations were noted for the logistics and food industries. Apart from these four alternatives, there were no alterations in the rankings of the other options. Moreover, the rankings of the alternatives occupying the top three positions remained unchanged, as did those in the bottom three spots. Table 29 illustrates the correlation among the obtained results.

As illustrated in Table 29, the findings for the automotive sector align closely with those for the logistics and food industries. Minor discrepancies exist between the outcomes for the health and energy sectors compared to the automotive industry, the primary focus of this study, which does not significantly alter the overall conclusion. Notably, only the rankings of criteria C9 and C10 experienced alterations, while the remaining criteria maintained their positions with comparable importance ratings. Thus, disregarding these minimal and inconsequential deviations, it is evident that the proposed model can be widely applied, particularly concerning the processes associated with selecting the DL platform for various industries beyond automotive.

In future studies, researchers may address the problem using recent, providing reasonable conclusions and popular decision-making approaches, such as ARTASI (Alternative ranking technique based on adaptive standardized intervals) (Pamucar et al., 2024), the ranks alternatives based on median similarity (RAMS) (Abdulaal and Bafail, 2022), ranking the alternatives based on the trace to median index (RATMI) (Abdulaal and Bafail, 2022), the multiple criteria ranking by alternative trace (MCRAT) (Urošević et al., 2021), Ranking of

Table 28
The ranking results of the alternatives for various industries.

Codes	Alternatives	Logistics	Health	Food	Energy	Automobile
A1	Google Cloud Deep Learning Containers	9	9	9	9	10
A2	Microsoft Cognitive Toolkit	13	13	13	13	14
A3	Neuton AutoML	8	8	8	8	8
A4	Knet	5	4	5	5	5
A5	NVIDIA Deep Learning	16	15	16	15	16
A6	Swift AI	12	11	12	11	12
A7	Theano	4	5	4	4	4
A8	Chainer	17	17	17	17	17
A9	Clarifai	11	12	11	12	11
A10	Coffee	14	14	14	14	13
A11	DeepPy	19	19	19	19	19
A12	Bitnami Pytorch	15	16	15	16	15
A13	Neon	6	6	6	6	6
A14	Neuroph	18	18	18	18	18
A15	Torch	3	3	3	3	3
A16	AWS Deep Learning AMIs	7	7	7	7	7
A17	CNTK	2	2	2	2	2
A18	TensorFlow	1	1	1	1	1
A19	Keras	10	10	10	10	9

Table 29
The ranking results of the alternatives for various industries.

	Logistics	Health	Food	Energy	Automobile
Logistics	1.000	0.995	1.000	0.996	0.996
Health	0.995	1.000	0.995	0.998	0.991
Food	1.000	0.995	1.000	0.996	0.996
Energy	0.996	0.998	0.996	1.000	0.993
Automobile	1.000	0.991	0.996	0.993	1.000

Table 27
Details of the professionals in five industries.

Es	Expertise	Title	Industry	Exp.	Graduate	Language
L_{E1}	Applications development	Senior engineer	Logistics	12	Computer Eng.	Python
L_{E2}	Project development	Senior consultant	Logistics	7	Information Tech.	Python, C++
L_{E3}	Data scientist	IT manager	Logistics	6	Industrial Eng.	Python
L_{E4}	Safety management	Transportation engineer	Logistics	5	Industrial Eng.	Python, C++
L_{E5}	Data Science	Senior data scientist	Logistics	8	Industrial Eng.	Java, R
H_{E1}	Medical informatics	Team leader	Health	11	Computer Eng.	Python, Java, R
H_{E2}	AI Researcher	Researcher	Health	9	Electronic Eng.	Python, Java, Julia
H_{E3}	Data science	Health Consultant	Health	7	Software Eng.	Python, R
H_{E4}	Data science	Team Leader	Health	12	Electronic Eng.	C++, Java
H_{E5}	Software development	Software Engineer	Health	8	Computer Eng.	Python
F_{E1}	Research	Agronomy researcher	Food	12	Food Eng.	Python
F_{E2}	Deep Learning	Data Scientist	Food	7	Mechanical Eng.	Python
F_{E3}	Automation	Data Scientist	Agri-food	9	Computer Tech.	Python
F_{E4}	Data safety	Marketing Analyst	Food	13	Mechanical Eng.	Python
F_{E5}	Genetic	Genetic scientist	Agri-food	7	Biotechnology	Python, Java
E_{E1}	Power systems	Software analysist	Energy	9	Electric Eng.	Python, Java, R
E_{E2}	Energy systems	Manager	Energy	14	Electric Eng.	Python, Java
E_{E3}	Energy management	Automation engineer	Energy	4	Mechatronic Eng.	Java
E_{E4}	Power systems	Engineer	Energy	5	Power Systems	Java, R
E_{E5}	Computer Vision	Specialist	Energy	8	Mechanical Eng.	Python
A_{A1}	Data Analysis	Assoc. Professor	Automotive	19	MIS	Python, Java
A_{A2}	Data Analysis	Assist. Professor	Automotive	15	MIS	Python
A_{A3}	Data Analysis	Assist. Professor	Automotive	14	Computer Eng.	Java, Julia
A_{A4}	Data Analysis	Data analyst	Automotive	24	Electronic Eng.	Python
A_{A5}	Data Management	IT Manager	Automotive	18	Computer Eng.	Python, Java

Alternatives through Functional mapping of criterion sub-intervals into a Single Interval (RAFSI) (Ali et al., 2024), Weighted integrated Sum-Product (WISP) (Stanujkic et al., 2023). Afterwards, they can compare the results obtained with the results of this study. In addition, the model proposed in this study can be extended with the help of different fuzzy sets and used to solve the decision-making problem in question. In addition, alternatives and evaluation criteria may be updated and added to future studies considering current developments. In addition, the researchers may prefer to extend the suggested model by employing various fuzzy sets such as Intuitionistic FSs (Tripathi et al., 2023), Neutrosophic FSs (Neutrosophy, 1998), Hesitant FSs (Zhou et al., 2022), q-Rung Orthopair FSs (Yager, 2017) and Spherical FSs (Biswas et al., 2023).

CRediT authorship contribution statement

Zoran Gligorić: Conceptualization, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Ömer Faruk Görçün:** Data curation, Investigation, Validation, Visualization, Writing – original draft. **Miloš Gligorić:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **Dragan Pamucar:** Conceptualization, Supervision, Validation, Writing – review & editing. **Vladimir Simic:** Formal analysis, Investigation, Validation, Visualization. **Hande Küçükönder:** Formal analysis, Investigation, Validation, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

Table A1
Expert matrix.

Code	DLT	DM	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
A1	T1	DM1	P	H	VH	P	H	P	P	H	H	H	P	VH
		DM2	P	H	VH	H	H	VH	H	H	VH	H	H	VH
		DM3	MH	MH	H	MH	MH	H	VH	MH	MH	H	MH	H
		DM4	VH	MH	P	MH	MH	MH	H	MH	MH	MH	MH	MH
		DM5	M	M	MH	M	M	MH	M	M	MH	MH	M	H
A2	T2	DM1	H	H	VH	H	H	VH	P	H	H	H	H	VH
		DM2	H	H	VH	H	H	VH	H	H	VH	H	H	VH
		DM3	MH	MH	H	MH	MH	H	VH	MH	MH	MH	MH	H
		DM4	MH	MH	MH	MH	MH	MH	H	MH	MH	MH	MH	MH
		DM5	M	M	MH	M	M	MH	M	M	MH	M	M	H
A3	T3	DM1	VH	VH	H	VH	VH	H	VH	VH	VH	VH	VH	H
		DM2	VH	VH	H	VH	VH	H	VH	VH	H	VH	VH	H
		DM3	H	H	MH	H	H	MH	H	H	H	H	H	MH
		DM4	MH	MH	MH	MH	MH	MH	MH	MH	MH	MH	MH	MH
		DM5	MH	MH	M	MH	MH	M	MH	MH	M	MH	MH	M
A4	T4	DM1	VH	VH	VH	VH	VH	VH	VH	VH	VH	VH	VH	VH
		DM2	VH	VH	VH	VH	VH	VH	VH	VH	VH	VH	VH	VH
		DM3	H	H	H	H	H	H	H	H	H	H	H	H
		DM4	MH	MH	MH	MH	MH	MH	MH	MH	MH	MH	MH	MH
		DM5	MH	MH	MH	MH	MH	MH	MH	MH	MH	MH	MH	MH
A5	T5	DM1	H	H	H	H	H	H	P	H	VH	VH	H	VH
		DM2	H	H	H	H	H	H	H	H	H	VH	H	VH
		DM3	H	MH	H	H	MH	H	VH	MH	H	H	MH	H
		DM4	MH	M	MH	MH	M	MH	H	M	MH	MH	MH	MH
		DM5	M	M	M	M	M	M	MH	M	MH	MH	M	MH
A6	T6	DM1	H	VH	H	H	VH	H	P	VH	P	H	VH	H
		DM2	H	VH	H	H	VH	H	H	VH	H	H	VH	H
		DM3	MH	H	MH	MH	H	MH	VH	H	VH	H	H	MH
		DM4	MH	MH	M	MH	MH	M	H	MH	H	MH	MH	MH
		DM5	M	MH	M	M	MH	M	M	MH	M	MH	MH	M
A7	T7	DM1	P	H	VH	P	H	VH	VH	H	VH	P	VH	VH
		DM2	P	H	VH	P	H	VH	P	H	VH	P	VH	VH
		DM3	VH	H	H	VH	H	H	H	H	H	VH	H	H
		DM4	H	MH	MH	H	MH	MH	MH	MH	MH	H	MH	MH
		DM5	H	M	MH	MH	M	MH	H	MH	MH	H	MH	MH
A8	T8	DM1	H	H	H	H	H	H	P	H	VH	H	H	VH
		DM2	H	H	H	H	H	H	H	H	H	H	H	VH
		DM3	MH	MH	H	MH	MH	H	VH	MH	H	MH	MH	H
		DM4	MH	M	MH	MH	M	MH	H	M	MH	MH	M	MH
		DM5	M	M	M	M	M	M	M	M	MH	M	M	MH
A9	T9	DM1	VH	H	VH	VH	H	VH	VH	H	VH	VH	H	VH
		DM2	VH	H	VH	VH	H	VH	VH	H	VH	VH	H	VH
		DM3	H	MH	H	H	MH	H	VH	MH	H	H	H	H
		DM4	MH	MH	MH	MH	MH	MH	H	MH	MH	MH	MH	MH
		DM5	MH	M	MH	M	M	MH	MH	M	MH	MH	MH	MH
A10	T10	DM1	VH	H	H	VH	H	H	VH	H	P	VH	H	H
		DM2	VH	H	H	VH	H	H	VH	H	H	VH	H	H

(continued on next page)

Table A1 (*continued*)

Code	DLT	DM	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
		DM3	H	MH	MH	H	MH	MH	H	MH	VH	H	H	MH
		DM4	MH	MH	M	MH	MH	M	MH	MH	H	MH	MH	MH
		DM5	MH	M	M	MH	M	M	MH	M	M	MH	MH	M
A11	T11	DM1	H	MH	H	H	MH	H	P	MH	P	VH	MH	H
		DM2	H	MH	H	H	MH	H	H	MH	H	VH	MH	H
		DM3	H	MH	MH	H	MH	MH	VH	MH	VH	H	MH	MH
		DM4	MH	M	M	MH	M	M	H	M	H	MH	M	M
		DM5	M	ML	M	M	ML	M	MH	ML	M	MH	M	M
A12	T12	DM1	VH	MH	VH	VH	MH	VH	VH	H	VH	VH	H	VH
		DM2	VH	MH	VH	VH	MH	VH	VH	H	VH	VH	H	VH
		DM3	H	MH	H	H	MH	H	H	MH	H	H	MH	H
		DM4	MH	M	MH	MH	M	MH	MH	M	MH	MH	M	MH
		DM5	MH	ML	MH	MH	ML	MH	MH	M	MH	MH	M	MH
A13	T13	DM1	VH	H	P	VH	H	P	VH	H	H	VH	H	P
		DM2	VH	H	P	VH	H	P	VH	H	P	VH	H	P
		DM3	H	MH	VH	H	MH	VH	H	MH	MH	H	MH	VH
		DM4	MH	MH	H	MH	MH	H	MH	MH	M	MH	MH	H
		DM5	MH	M	H	MH	M	H	MH	M	H	H	M	H
A14	T14	DM1	VH	MH	H	VH	MH	H	VH	MH	VH	VH	MH	VH
		DM2	VH	MH	H	VH	MH	H	VH	MH	H	VH	MH	VH
		DM3	H	M	H	H	M	H	H	M	H	H	M	H
		DM4	MH	M	MH	MH	M	MH	MH	M	MH	MH	M	MH
		DM5	MH	ML	M	MH	ML	M	MH	ML	MH	H	ML	MH
A15	T15	DM1	VH	VH	P	VH	VH	P	VH	VH	H	VH	VH	P
		DM2	VH	VH	P	VH	VH	P	VH	VH	P	VH	VH	P
		DM3	H	H	VH	H	H	VH	H	H	MH	H	H	VH
		DM4	MH	MH	H	MH	MH	H	MH	MH	M	MH	MH	H
		DM5	MH	MH	H	MH	MH	H	MH	MH	H	MH	MH	H
A16	T16	DM1	VH	VH	H	VH	VH	H	VH	VH	VH	VH	VH	H
		DM2	VH	VH	H	VH	VH	H	VH	VH	H	VH	VH	H
		DM3	H	H	MH	H	H	MH	H	H	VH	H	H	MH
		DM4	MH	MH	MH	MH	MH	MH	MH	MH	H	MH	MH	MH
		DM5	MH	MH	M	MH	MH	M	MH	MH	M	H	MH	M
A17	T17	DM1	P	VH	P	P	VH	P	H	VH	H	P	VH	P
		DM2	P	VH	P	P	VH	P	P	VH	P	P	VH	P
		DM3	VH	H	VH	VH	H	VH	MH	H	MH	VH	H	VH
		DM4	H	MH	H	H	MH	H	MH	MH	M	H	MH	H
		DM5	H	MH	H	H	MH	H	H	MH	H	H	H	H
A18	T18	DM1	P	P	P	P	H	P	H	P	H	P	P	P
		DM2	H	P	H	P	P	P	P	MH	P	M	P	P
		DM3	VH	VH	VH	VH	VH	VH	MH	VH	MH	VH	VH	VH
		DM4	H	H	H	H	H	H	MH	H	M	H	H	H
		DM5	H	H	H	H	H	H	H	H	H	H	H	H
A19	T19	DM1	VH	H	VH	VH	H	VH	VH	H	H	VH	H	VH
		DM2	P	VH	P	P	H	P	P	P	H	P	MH	P
		DM3	H	MH	H	H	MH	H	H	MH	MH	H	MH	H
		DM4	MH	M	MH	MH	M	MH	MH	M	MH	MH	MH	MH
		DM5	MH	M	MH	MH	M	MH	MH	M	MH	MH	M	H

T1 – Google Cloud Deep Learning Containers; T2 – Microsoft Cognitive Toolkit; T3 – Neuton AutoML; T4 – Knet; T5 – NVIDIA Deep Learning; T6 – Swift AI; T7 – Theano; T8 – Chainer; T9 – Clarifai; T10 – Coffee; T11 – DeepPy; T12 – Bitnami Pytorch; T13 – Neon; T14 – Neuroph; T15 – Torch; T16 – AWS Deep Learning AMIs; T17 – CNTK; T18 – TensorFlow; T19 – Keras

Table A2
Estimation of criteria by experts.

Codes	DM1	DM2	DM3	DM4	DM5
C1	H	VH	VH	P	H
C2	P	H	VH	VH	P
C3	ML	M	M	MH	M
C4	H	H	VH	P	MH
C5	MH	MH	ML	M	MH
C6	H	H	VH	MH	M
C7	P	P	VH	H	H
C8	VH	H	H	MH	M
C9	L	ML	ML	VL	L
C10	M	MH	MH	H	VH
C11	ML	L	L	L	MH
C12	H	H	VH	P	VH

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al., 2016. TensorFlow: a system for large-scale machine learning. In: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016.

Abboush, M., Bamal, D., Knieke, C., Rausch, A., 2022. Intelligent fault detection and classification based on hybrid deep learning methods for hardware-in-the-loop test of automotive software systems. Sensors 22. https://doi.org/10.3390/s22114066.

Abdulaal, R.M.S., Bafail, O.A., 2022. Two new approaches (RAMS-RATMI) in multi-criteria decision-making tactics. J. Math. 2022. doi: 10.1155/2022/6725318.

Al-Bdour, G., Al-Qurran, R., Al-Ayyoub, M., Shatnawi, A., 2020. Benchmarking open source deep learning frameworks. Int. J. Electr. Comput. Eng. 10 https://doi.org/10.11591/IJECE.V10I5.PP5479-5486.

Ali, J., Naser Hussain, K., Alnoor, A., Muhsen, Y.R., Atiyah, A.G., 2024. Benchmarking methodology of banks based on financial sustainability using CRITIC and RAFSI techniques. Decis. Mak.: Appl. Manag. Eng. 7 (1), 315–341. https://doi.org/10.31181/dmame712024945.

Altenmüller, T., Stüker, T., Waschneck, B., Kuhnle, A., Lanza, G., 2020. Reinforcement learning for an intelligent and autonomous production control of complex job-shops under time constraints. Prod. Eng. 14 https://doi.org/10.1007/s11740-020-00967-8.

Azer, S.A., 2009. Interactions between students and tutor in problem-based learning: the significance of deep learning. Kaohsiung J. Med. Sci. 25 https://doi.org/10.1016/S1607-551X(09)70068-3.

Bahrampour, S., Ramakrishnan, N., Schott, L., Shah, M., 2016. Comparative study of deep learning software frameworks. ArXiv:151106435, 1–9. doi: 10.48550/arXiv.1511.06435.

Bejnordi, B.E., Veta, M., Van Diest, P.J., Van Ginneken, B., Karssemeijer, N., Litjens, G., et al., 2017. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. JAMA - J. Am. Med. Assoc. 318 https://doi.org/10.1001/jama.2017.14585.

Biswas, S., Bozanic, D., Pamucar, D., Marinkovic, D., 2023. A spherical fuzzy based decision making framework with Einstein aggregation for comparing preparedness of SMES in quality 4.0. Facta Univ.: Ser. Mech. Eng. 21 (3), 453–478. https://doi.org/10.22190/FUME230831037B.

Block, S.B., Da Silva, R.D., Dorini, L.B., Minetto, R., 2021. Inspection of imprint defects in stamped metal surfaces using deep learning and tracking. IEEE Trans. Ind. Electron. 68 https://doi.org/10.1109/TIE.2020.2984453.

Chang, K.H., 2024. Combining subjective and objective weights considerations to solve the emergency location selection problems under spherical fuzzy environments. Appl. Soft Comput. 153, 111272.

Chavan, S., 2024. Applications of Machine Learning in Automotive Verification and Validation: A Review. Techno-societal 2022. Springer International Publishing, Cham, pp. 291–303. doi: 10.1007/978-3-031-34644-6_32.

Chen, T., Li, M., Cmu, U.W., Li, Y., Lin, M., Wang, N., et al., 2015. MXNet: a flexible and efficient machine learning library for heterogeneous distributed systems arXiv: 1512.01274v1 [cs.DC] 3 Dec 2015. Emerald Group Publishing Limited. 36.

Chinta, V.S., Kethi Reddi, S., Yarramsetty, N., 2023. Optimal feature selection on Serial Cascaded deep learning for predictive maintenance system in automotive industry with fused optimization algorithm. Adv. Eng. Inf. 57 https://doi.org/10.1016/j.aei.2023.102105.

Dean, J., Corrado, G.S., Monga, R., Chen, K., Devin, M., Le, Q.V., et al., 2012. Large scale distributed deep networks. Adv. Neural Inf. Process. Syst. 2.

Do, H.H., Prasad, P.W.C., Maag, A., Alsadoon, A., 2019. Deep learning for aspect-based sentiment analysis: a comparative review. Expert Syst. Appl. 118 https://doi.org/10.1016/j.eswa.2018.10.003.

Dong, S., Zhan, J., Hu, W., Mohajer, A., Bavaghar, M., Mirzaei, A., 2023. Energy-efficient hierarchical resource allocation in uplink-downlink decoupled NOMA HetNets. IEEE Trans. Netw. Serv. Manag. 20. doi: 10.1109/TNSM.2023.3239417.

Druzhkov, P.N., Kustikova, V.D., 2016. A survey of deep learning methods and software tools for image classification and object detection. Patt. Recogn. Image Anal. 26. doi: 10.1134/S1054661816010065.

El, H.C., Perrot, G., Painvin, L., Couturier, R., 2021. Automation of quality control in the automotive industry using deep learning algorithms. In: 2021 International Conference on Computer, Control and Robotics, ICCCR 2021. https://doi.org/10.1109/ICCCR49711.2021.9349273.

Espinosa, R., Ponce, H., Gutiérrez, S., 2021. Click-event sound detection in automotive industry using machine/deep learning. Appl. Soft. Comput. 108 https://doi.org/10.1016/j.asoc.2021.107465.

Fenn, P., 2020. The deep learning revolution. Inf. Commun. Soc. 23. doi: 10.1080/1369118x.2020.1726988.

Gal, Y., Ghahramani, Z., 2016. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: 33rd International Conference on Machine Learning, ICML 2016, Vol. 3.

Georgevici, A.I., Terblanche, M. 2019. Neural networks and deep learning: a brief introduction. Intensive Care Med. 45. doi: 10.1007/s00134-019-05537-w.

Görçün, Ö.F., 2020. Efficiency analysis of black sea container seaports: application of an integrated MCDM approach. Marit. Policy Manag. 48, 672–699. https://doi.org/10.1080/03088839.2020.1783467.

Gudivada, V.N., Apon, A., Ding, J., 2017. Data quality considerations for big data and machine learning: going beyond data cleaning and transformations. Int. J. Adv. Softw. 10.

HackerNoon, 2024. Machine Learning Costs: Price Factors and Real-World Estimates | HackerNoon. https://hackernoon.com/machine-learning-costs-price-factors-and-real-world-estimates (accessed March 16, 2024).

Hatcher, W.G., Yu, W., 2018. A survey of deep learning: platforms, applications and emerging research trends. IEEE Access 6. https://doi.org/10.1109/ACCESS.2018.2830661.

Hernández-Blanco, A., Herrera-Flores, B., Tomás, D., Navarro-Colorado, B., 2019. A systematic review of deep learning approaches to educational data mining. Complexity 2019. https://doi.org/10.1155/2019/1306039.

Hwang, C.L., Yoon, K., 1981. Methods for Multiple Attribute Decision Making. Lecture Notes in Economics and Mathematical Systems Vol. 186, 58–191.

Jhung, J., Kim, S., 2021. Behind-the-scenes (Bts): Wiper-occlusion canceling for advanced driver assistance systems in adverse rain environments. Sensors 21. https://doi.org/10.3390/s21238081.

Junaid, M., Szalay, Z., Török, Á., 2021. Evaluation of non-classical decision-making methods in self driving cars: pedestrian detection testing on cluster of images with different luminance conditions. Energies (Basel) 14. doi: 10.3390/en14217172.

Kabakuş, A.T., 2020. A comparison of the state-of-the-art deep learning platforms: an experimental study. Sakarya Univ. J. Comput. Inf. Sci. 3 https://doi.org/10.35377/saucis.03.03.776573.

Kaya, U., Yılmaz, A., Dikmen, Y., 2019. Sağlık Alanında Kullanılan Derin Öğrenme Yöntemleri. Eur. J. Sci. Technol. https://doi.org/10.31590/ejosat.573248.

Kejun, W., Yandong, Z., Xianglei, X., 2018. Deep learning in driverless vehicles. CAAI Trans. Intell. Syst.

Kim, J., Ha, S., 2023. Energy-aware scenario-based mapping of deep learning applications onto heterogeneous processors under real-time constraints. IEEE Trans. Comput. 72 https://doi.org/10.1109/TC.2022.3218991.

Lee, S., Kang, D., 2019. Development of interval-valued fuzzy GRA with SERVPERF based on subjective and objective weights for evaluation of airline service quality: a case study of Korea low-cost carriers. PLoS One 14 (8), e0219739.

Llopis-Albert, C., Rubio, F., Valero, F., 2021. Impact of digital transformation on the automotive industry. Technol. Forecast. Soc. Change 162. https://doi.org/10.1016/j.techfore.2020.120343.

Luckow, A., Cook, M., Ashcraft, N., Weill, E., Djerekarov, E., Vorster, B., 2016. Deep learning in the automotive industry: applications and tools. In: Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016. https://doi.org/10.1109/BigData.2016.7841045.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A., 2018. Towards deep learning models resistant to adversarial attacks. 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings.

Mahajan, N., Chauhan, A., Kumar, H., Kaushal, S., Sangaiah, A.K., 2022. A deep learning approach to detection and mitigation of distributed denial of service attacks in high availability intelligent transport systems. Mob. Netw. Appl. 27. doi: 10.1007/s11036-022-01973-z.

McCulloch, W.S., Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. 5 https://doi.org/10.1007/BF02478259.

McDonald, A., n.d. Data Quality Considerations for Machine Learning Models | by Andy McDonald | Towards Data Science. https://towardsdatascience.com/data-quality-considerations-for-machine-learning-models-dcbe9cab34cb (accessed September 21, 2023).

Mohajer, A., Sorouri, F., Mirzaei, A., Ziaeddini, A., Jalali Rad, K., Bavaghar, M., 2022. Energy-aware hierarchical resource management and backhaul traffic optimization in heterogeneous cellular networks. IEEE Syst. J. 16 https://doi.org/10.1109/JSYST.2022.3154162.

Mohajer, A., Sam Daliri, M., Mirzaei, A., Ziaeddini, A., Nabipour, M., Bavaghar, M., 2023. Heterogeneous computational resource allocation for NOMA: toward green mobile edge-computing systems. IEEE Trans. Serv. Comput. 16 https://doi.org/10.1109/TSC.2022.3186099.

Neutrosophy, S.F., 1998. Neutrosophic Probability, Set, and Logic: Analytic Synthesis & Synthetic Analysis. American Research Press.

Nikitas, A., Michalakopoulou, K., Njoya, E.T., Karampatzakis, D., 2020. Artificial intelligence, transport and the smart city: Definitions and dimensions of a new mobility era. Sustainability (Switzerland) 12. doi: 10.3390/su12072789.

Pamučar, D., Ćirović, G., 2015. The selection of transport and handling resources in logistics centers using Multi-Attributive Border Approximation area Comparison (MABAC). Expert Syst. Appl. 42, 3016–3028. https://doi.org/10.1016/j.eswa.2014.11.057.

Pamucar, D., Simic, V., Görçün, Ö.F., Küçükönder, H., 2024. Selection of the best Big Data platform using COBRAC-ARTASI methodology with adaptive standardized intervals. Expert Syst. Appl. 239, 122312 https://doi.org/10.1016/j.eswa.2023.122312.

Parikh, N., 2014. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour (FIXME). Foundations and Trends® in Optimization, 1.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al., 2019. PyTorch: An imperative style, high-performance deep learning library. Adv. Neural Inf. Process. Syst. 32.

Pham, V.T., 2021. A study on supply chain management and characteristics of container port operation efficiency in Vietnam by emerging measure tools. Webology 18. https://doi.org/10.14704/WEB/V18SI04/WEB18122.

Pham, H.V., Qian, S., Wang, J., Lutellier, T., Rosenthal, J., Tan, L., et al., 2020. Problems and opportunities in training deep learning software systems: an analysis of variance. In: Proceedings - 2020 35th IEEE/ACM International Conference on Automated Software Engineering, ASE 2020. doi: 10.1145/3324884.3416545.

Qi, S., Yang, J., Zhong, Z., 2020. A review on industrial surface defect detection based on deep learning technology. In: ACM International Conference Proceeding Series. https://doi.org/10.1145/3426826.3426832.

Rao, Q., Frtunikj, J., 2018. Deep learning for self-driving cars: chances and challenges: extended abstract. In: Proceedings - International Conference on Software Engineering. https://doi.org/10.1145/3194085.3194087.

Rao, M.N., 2023. A comparative analysis of deep learning frameworks and libraries. Int. J. Intell. Syst. Appl. Eng. 11.

Rio-Torto, I., Campanico, A.T., Pereira, A., Teixeira, L.F., Filipe, V., 2021. Automatic quality inspection in the automotive industry: a hierarchical approach using simulated data. In: 2021 IEEE 8th International Conference on Industrial Engineering and Applications, ICIEA 2021. https://doi.org/10.1109/ICIEA52957.2021.9436742.

Roh, D.H., Lee, J.Y., 2023. Augmented reality-based navigation using deep learning-based pedestrian and personal mobility user recognition - a comparative evaluation for driving assistance. IEEE Access 11. https://doi.org/10.1109/ACCESS.2023.3286872.

Rudy, S.H., Nathan Kutz, J., Brunton, S.L., 2019. Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. J. Comput. Phys. 396 https://doi.org/10.1016/j.jcp.2019.06.056.

Ruospo, A., Piumatti, D., Floridia, A., Sanchez, E.A., 2021. A suitability analysis of software based testing strategies for the on-line testing of artificial neural networks applications in embedded devices. In: Proceedings - 2021 IEEE 27th International Symposium on On-Line Testing and Robust System Design, IOLTS 2021. https://doi.org/10.1109/IOLTS52814.2021.9486704.

Šabanovič, E., Kojis, P., Šukevičius, Š., Shyrokau, B., Ivanov, V., Dhaens, M., et al., 2021. Feasibility of a neural network-based virtual sensor for vehicle unsprung mass relative velocity estimation. Sensors 21. https://doi.org/10.3390/s21217139.

Schmiedt, M., He, P., Rinderknecht, S., 2022. Target state optimization: drivability improvement for vehicles with dual clutch transmissions. Appl. Sci. (Switzerland) 12. doi: 10.3390/app122010283.

Schoch, A., Refflinghaus, R., Zivkovic, P., 2023. Deep learning based predictive testing strategy in the automotive industry. Proc. CIRP 118, 1108–1113. https://doi.org/10.1016/j.procir.2023.06.190.

Şeker, A., Diri, B., Balık, H., 2017. Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme. Gazi J. Eng. Sci. 3.

Sharifi, A., Zibaei, A., Rezaei, M., 2021. A deep learning based hazardous materials (HAZMAT) sign detection robot with restricted computational resources. Mach. Learn. Appl. 6 https://doi.org/10.1016/j.mlwa.2021.100104.

Sharma, A., Zhang, Z., Rai, R., 2021. The interpretive model of manufacturing: a theoretical framework and research agenda for machine learning in manufacturing. Int. J. Prod. Res. 59 https://doi.org/10.1080/00207543.2021.1930234.

Shekhovtsov, A., Dobryakova, L., 2023. Comparison of multi-criteria decision methods for customer-centered decision making: a practical study case. Proc. Comput. Sci. 225, 4824–4835.

Sherkhane, P., Vora, D., 2017. Survey of deep learning software tools. In: 2017 International Conference on Data Management, Analytics and Innovation, ICDMAI 2017. https://doi.org/10.1109/ICDMAI.2017.8073516.

Shi, S., Wang, Q., Xu, P., Chu, X., 2017. Benchmarking state-of-the-art deep learning software tools. In: Proceedings - 2016 7th International Conference on Cloud Computing and Big Data, CCBD 2016, 2017. doi: 10.1109/CCBD.2016.029.

Shi, Z., Huang, X., Jain, A., Lin, C., 2019. Applying deep learning to the cache replacement problem. In: Proceedings of the Annual International Symposium on Microarchitecture. https://doi.org/10.1145/3352460.3358319.

Singh, K.B., Arat, M.A., 2019. Deep learning in the automotive industry: recent advances and application examples. Comput. Sci. https://doi.org/10.48550/arXiv.1906.08834.

Singh, A., Sengupta, S., Lakshminarayanan, V., 2020. Explainable deep learning models in medical image analysis. J. Imaging 6. doi: 10.3390/JIMAGING6060052.

Stanujkic, D., Popovic, G., Karabasevic, D., Meidute-Kavaliauskiene, I., Ulutas, A., 2023. An integrated simple weighted sum product method - WISP. IEEE Trans. Eng. Manag. 70 https://doi.org/10.1109/TEM.2021.3075783.

Sze, V., Budagavi, M., Sullivan, G.J., 2013. High efficiency video coding (HEVC) Algorithms and Architecture. Vol. 36.

Sze, V., Chen, Y.H., Yang, T.J., Emer, J.S., 2017. Efficient processing of deep neural networks: a tutorial and survey. Proc. IEEE 105. doi: 10.1109/JPROC.2017.2761740.

Theissler, A., Pérez-Velázquez, J., Kettelgerdes, M., Elger, G., 2021. Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry. Reliab. Eng. Syst. Saf. 215. doi: 10.1016/j.ress.2021.107864.

Torkayesh, A.E., Ecer, F., Pamucar, D., Karamaşa, Ç., 2021. Comparative assessment of social sustainability performance: integrated data-driven weighting system and CoCoSo model. Sustain. Cities Soc. 71, 102975.

Tripathi, D.K., Nigam, S.K., Rani, P., Shah, A.R., 2023. New intuitionistic fuzzy parametric divergence measures and score function-based CoCoSo method for decision-making problems. Decis. Mak.: Appl. Manag. Eng. 6 (1), 535–563. https://doi.org/10.31181/dmame0318102022t.

Ulker, B., Stuijk, S., Corporaal, H., Wijnhoven, R., 2020. Reviewing inference performance of state-of-the-art deep learning frameworks. In: Proceedings of the 23rd International Workshop on Software and Compilers for Embedded Systems, SCOPES 2020. https://doi.org/10.1145/3378678.3391882.

Um, J., Stroud, I.A., Park, Y., 2019. Deep learning approach of energy estimation model of remote laser welding. Energies (Basel) 12. https://doi.org/10.3390/en12091799.

Urošević, K., Gligorić, Z., Miljanović, I., Beljić, C., Gligorić, M., 2021. Novel methods in multiple criteria decision-making process (Mcrat and raps)—application in the mining industry. Mathematics 9. https://doi.org/10.3390/math9161980.

Wang, J., Ma, Y., Zhang, L., Gao, R.X., Wu, D., 2018. Deep learning for smart manufacturing: Methods and applications. J. Manuf. Syst. 48 https://doi.org/10.1016/j.jmsy.2018.01.003.

Wen, T.C., Chung, H.Y., Chang, K.H., Li, Z.S., 2021. A flexible risk assessment approach integrating subjective and objective weights under uncertainty. Eng. Appl. Artif. Intel. 103, 104310.

Werda, M.S., Saify, T.A., Kouiss, K., Gaber, J., 2022. Automating the dataset generation and annotation for a deep learning based robot trajectory adjustment application for welding processes in the automotive industry. Comput. Inf. 41. doi: 10.31577/cai_2022_1_271.

Więckowski, J., Kizielewicz, B., Shekhovtsov, A., Sałabun, W., 2023. RANCOM: a novel approach to identifying criteria relevance based on inaccuracy expert judgments. Eng. Appl. Artif. Intel. 122, 106114.

Wong, S.C., Gatt, A., Stamatescu, V., McDonnell, M.D., 2016.. Understanding data augmentation for classification: when to warp?. In: 2016 International Conference on Digital Image Computing: Techniques and Applications 2016 https://doi.org/10.1109/DICTA.2016.7797091.

Yager, R.R., 2017. Generalized Orthopair fuzzy sets. IEEE Trans. Fuzzy Syst. 25, 1222–1230. https://doi.org/10.1109/TFUZZ.2016.2604005.

Yapıcı, M.M., Topaloğlu, N., 2021. Performance comparison of deep learning frameworks. Comput. Inf. 1.

Yin, L., 2024. Comparison between Deep Learning Tools | by Li Yin | Medium. https://liyin2015.medium.com/comparison-between-deep-learning-tools-42560e61a5d5 (accessed March 16, 2024).

Zavadskas, E.K., Kaklauskas, A., Banaitis, A., Kvederyte, N., 2004. Housing credit access model: the case for Lithuania. Eur. J. Oper. Res. 155 https://doi.org/10.1016/S0377-2217(03)00091-2.

Zavadskas, E.K., Turskis, Z., Antucheviciene, J., Zakarevicius, A., 2012. Optimization of weighted aggregated sum product assessment. Elektronika Ir Elektrotechnika 122, 3–6.

Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O., 2021. Understanding deep learning (still) requires rethinking generalization. Commun. ACM 64. https://doi.org/10.1145/3446776.

Zhao, H., Wang, Y., Guo, S., 2023. A hybrid MCDM model combining Fuzzy-Delphi, AEW, BWM, and MARCOS for digital economy development comprehensive evaluation of 31 provincial level regions in China. PLoS One 18 (4), e0283655.

Zhou, B., Chen, J., Wu, Q., Pamucar, D., Wang, W., Zhou, L., 2022. Risk priority evaluation of power transformer parts based on hybrid FMEA framework under hesitant fuzzy environment. Facta Univ., Ser.: Mech. Eng. 20 (2), 399–420. https://doi.org/10.22190/FUME220223013Z.

Zizka, A., Silvestro, D., Vitt, P., Knight, T.M., 2021. Automated conservation assessment of the orchid family with deep learning. Conserv. Biol. 35 https://doi.org/10.1111/cobi.13616.