University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)

ISSN: 2714-3627

A Journal of the Department of Computer Science, University of Ibadan, Ibadan, Nigeria

Volume 14 No. 1, June, 2025

journals.ui.edu.ng/uijslictr
http://uijslictr.org.ng/

University of Ibadan
Journal of Science and Logics in ICT Research

A Comprehensive Review of Case Representation and Similarity Measures in Case-Based Reasoning Systems

[1]✉ Omonijo O. O., [2]Akinola S. O., [3]Ugbogbo M. J., [4]Orumgbe C. and [5]Yusuf I. O.

[1, 3, 5]Computer Science Unit, Nigeria Maritime University, Okerenkoko, Nigeria
[2]Department of Computer Science, University of Ibadan, Ibadan, Nigeria
[4]Mechanical Engineering Department, Nigeria Maritime University, Okerenkoko, Nigeria
[1]seyiomonijo10@gmail.com

Abstract

Case-Based Reasoning (CBR) is a human-inspired problem-solving approach where new problems are solved by recalling and adapting solutions from similar past cases. The performance of a CBR system critically depends on how cases are represented and how similarity between cases is computed. These two factors determine the accuracy, efficiency and applicability of CBR systems across diverse domains. This paper presents a comprehensive and comparative review of various case representation techniques and similarity measures. The review evaluates these methods based on important measures such as interpretability, scalability, adaptability, computational complexity and retrieval effectiveness. It further explores their suitability across domains including healthcare, finance, engineering and disaster management. The analysis reveals that no single technique is universally optimal; rather, the alignment between representation format and similarity computation, often through hybridization or domain-specific adaptation, is critical to achieving optimal system performance. Through rich literature insights and practical illustrations, the paper identifies emerging trends such as machine learning-driven similarity adaptation, ontology automation and real-time retrieval, offering a roadmap for the next generation of intelligent and context-aware CBR systems.

Keywords: *Cases, Similarity measures, Retrieval accuracy, Case Reuse, k-nearest neighbour*

1. Introduction

Case-Based Reasoning (CBR) is an Artificial Intelligence (Al) methodology where problem-solving and decision-making are based on the retrieval and reuse of past experiences or "cases." This operates under the principle that similar problems have similar solutions. This approach aligns with human problem-solving strategies, where the decision-making process is influenced by previous encounters with similar situations. CBR systems work by comparing a new problem with previously solved problems which are stored in a library (called case bases). If an exact match is found, the stored solution is applied. Otherwise, the system adapts an existing case to fit the new situation.

CBR systems represent problem situations as cases. A case serves as a condensed repository of knowledge extracted from past experiences. It encapsulates not only the "distilled" knowledge gained from previous encounters but also the contextual framework in which these lessons derived their significance. Depending on the type of case, a case is composed of two integral components:

i **Problem Description**: a part, which represents the attributes of the case
ii **Solution**: which gives the corresponding outcome or solution of the previous case.

Ever since Schank and Abelson first presented CBR in 1977, it has evolved significantly [1]. The most widely used abstraction of CBR is the Aarnodt and Plaza (1994) cycle [2]. The model, presented in figure 1, is commonly known as the 4R workflow model. It is expressed as a cycle comprising of four phases and include;

Retrieval Phase: cases that have some form of similarity to the new problem are retrieved;

Reuse Phase: the identified solutions from the retrieved cases are used as solution to the new problem;

Omonijo O. O., Akinola S. O, Ugbogbo M. J., Orumgbe C. and Yusuf I. O. (2025). A Comprehensive Review of Case Representation and Similarity Measures in Case-Based Reasoning Systems. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 14 No. 1, pp. 118 – 132.

Revise Phase: adapts prospective solutions to fit where available solution does not perfectly fit and needs more revision; and

Retain Phase: revised case(s) learned by the system is finally stored.

One of the several attempts to improve the performance and reasoning framework of CBR led to the integration of case representation by Finnie and Sun (2003). They proposed the "5R" model of CBR by incorporating the case representation into the “4R” model [3]. Although, case representation is not a cyclic phase, it plays a crucial role in how CBR systems store and retrieve knowledge. It is a phase that precedes the case retrieval phase.

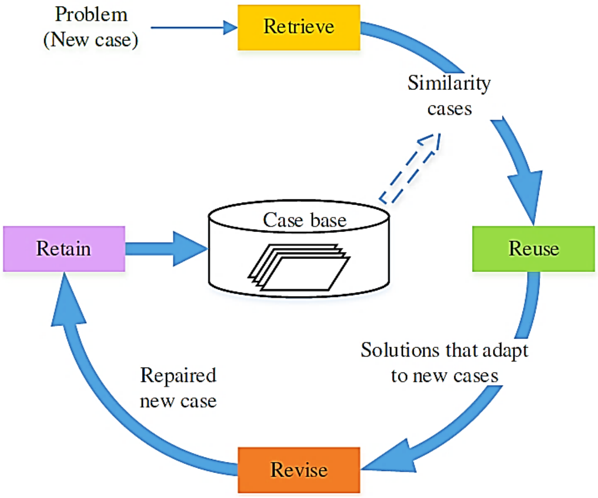

Figure 1: The 4R CBR Cyclic Structure [2]

The case retrieval phase is considered as being crucial in determining a CBR system's efficiency. Finding solutions that are most similar to a given problem requires the use of case representation and similarity measures during the retrieval phase of a CBR system [4]. The success and/or efficiency of a CBR system relies heavily on these two critical components. Case representation determines how knowledge (comprising problem descriptions, solutions and outcomes) is structured and stored. It affects not only how well the system can understand and organize past experiences but also how efficiently it can retrieve and adapt them. Similarity Measures compare a new case to a target case by using some form of matching metric to determine degree of similarity, thus influencing which solutions are considered relevant. Inaccuracies in either representation or similarity evaluation can significantly degrade the system’s performance.

Given the importance of these components, this review seeks to comparatively analyze the various methods of case representation and similarity measurement employed in CBR systems. While there are numerous studies addressing individual aspects of these components [5, 6, 7], a consolidated and comparative perspective remains underrepresented in literature. This paper fills that gap by systematically evaluating common and advanced techniques across multiple criteria such as scalability, interpretability, computational complexity and adaptability.

The adopted methodology involves an extensive literature survey of peer-reviewed publications, technical reports and applied case studies in different domains. The review categorizes representation techniques (e.g., feature vector, frame-based, ontology-based etc) and similarity measures (e.g., Euclidean, Cosine, Jaccard, Hybrid etc), and provides comparative tables summarizing their strengths, limitations and suitability across various contexts. This will provide researchers and practitioners a structured guide to selecting and combining appropriate methods for building efficient, accurate and context-sensitive CBR systems.

1. Case Representation in CBR

In CBR, the case is foundational and its representation is crucial to the system [8]. The problem-solving process of the CBR suggest that, case representation is not a cyclic phase of the abstraction of CBR but then, it is an important aspect that needs to be firstly treated for other phases to be able to perform efficiently. The primary objective of case representation is to encode past experiences, store and subsequently retrieve for problem-solving [9]. A well-structured representation not only facilitates effective case retrieval but also enhances reasoning, adaptation and reuse. The richness and accuracy of case representation influence both the quality of solutions and the system’s computational performance.

Case representation includes information that directly influences the outcome or solution of the problem described. Hence, the concept of modelling cases involves representing problem-solving instances in a structured format.

2.1 Types of Case Representation

Several methods have been developed to represent cases, depending on the domain, complexity of the data and the reasoning required. Common case representation methods include feature vector, frame representation, object-oriented representation, predicate-based representation, semantic networks and rule-based representation [10]. These approaches can be conveniently grouped into two main case representation classes: feature-vector representation and structure representation.

Feature Vector Representation: This is the most common approach, where each case is described using fixed-length attribute–value pairs. It is suitable for structured and numerical data such as medical records or sensor readings. This representation allows the use of traditional similarity metrics like Euclidean or Manhattan distance [10].

Structure Representation: This form of representation captures the relationships and dependencies between attributes in a more explicit manner. The main types of structure representation include:

a) **Frame-Based Representation**: Cases are modelled as frames, which are structured collections of slots (attributes) and fillers (values) which can inherit properties from other frames. This method allows for flexible representation of knowledge and facilitates inheritance and default reasoning [8].

b) **Object-Oriented Representation**: Cases are represented as objects encapsulating their attributes and behaviour. This representation supports hierarchical and modular designs, making it ideal for complex applications like CAD and manufacturing systems [11].

c) **Predicate-Based Representation**: It is based on first-order logic. This method uses logical predicates to describe the relationships between entities in a case. It is usually expressive but suitable for domains requiring rule-based inference, such as legal reasoning [10].

d) **Semantic Network and Ontology-Based Representation**: Semantic nets graphically model the relationships between concepts, while ontologies add structured domain knowledge and reasoning capabilities. These are especially useful in biomedical and text-based applications [7]. This method is useful for visualizing and analyzing complex relationships between cases.

e) **Rule-Based Representation**: Some cases are encoded as a set of rules or production rules (if–then statements). While powerful in well-defined domains, they can be rigid and hard to scale across heterogeneous data types [10].

Each representation method balances trade-offs in interpretability, expressiveness and computational cost. Typically, the appropriate representation method is chosen based on the specific application field [10]. Feature-vector representation is often preferred for simple domains with numerical data, while structure representation is more suitable for complex domains with intricate relationships between attributes.

2.2 Components of Case Representation

In a typical CBR system, a case is represented by three main components. These components include the problem description (*p*), solution (*s*) and in some case, the outcome (*o*). The problem description includes the goals, task description, constraints, initial data and other relevant information that define the problem to be solved. While the solution component encompasses the actual solution. The case outcome indicates whether the solution achieved the desired result or not. Optionally, steps taken to achieve the solution (trace), justification and annotation of the solution, alternative solutions are sometimes considered and expectations regarding the solution's outcomes. Additional, case components, such as explanations, with variations for different data types like text and image representation can be included [12].

a) Problem Description (p)

This includes the initial state, constraints, goals, and relevant features of the problem. It is represented by the problem-feature subsets and consists of a number of principal problem features. The problem description can be conceptualized as a sequence of problem features: $(f_1, f_2, f_3, ..., f_n)$ [12]. The problem features (f_i) of the problem description are represented as attribute-value pairs:

$fi = (a_i, v_i)$

The attribute-value pair is the most commonly used representation for problem features in CBR. Where a_i is the attribute of the problem which are defined in the problem features vocabulary and v_i are the values related to each attribute. Each feature describes a specific aspect of the problem.

With respect to object representation, features can be grouped into objects. This gives a more structured and organized representation of cases. By representing cases as objects, complex cases can be simplified by grouping related features [2]. Although, this concept is less frequently used than attribute-value pairs. For practical purposes, objects representations are often reduced to attribute-pair representations. Cases are represented in form of trees or graphs when relational objects representation is used [8].

The attributes are represented as nodes connected by edges. Attributes are identified by their paths from the root of the graph, requiring attribute names and paths for localization. This form of representation is beneficial for complex cases with non-homogeneous structures, such as cases involving multiple hierarchical levels or dependencies.

In more practical instances, CBR can incorporate more complex knowledge models, including plans, workflows, series, sequences and temporal components. These forms of models allow the representation of not only static attributes but also dynamic processes, sequences of actions and temporal aspects of problem-solving [13].

Selecting a knowledge model for problem descriptions in CBR depends on problem domain complexity, required detail level, retrieval and adaptation efficiency. Attribute-value pairs are often preferred for their simplicity, while more complex models like relational objects and object representations are used for handling intricate and hierarchical structures in problem descriptions.

b) Solution Description (s)

This defines the proposed or actual solution to the problem. It may include actions taken, decisions made or recommended interventions [14].

In some instances, there may be need for additional solution details. The solution description can encompass other components such as [6]:

a. **How the solution was obtained**: Information about the methodology or process used to arrive at the solution.

b. **Quality of the solution**: Metrics or assessments indicating the quality or reliability of the solution.

c. **Constraints**: Any limitations or constraints that affect the application of the solution.

d. **Alternative solutions**: Other possible solutions that were considered but not chosen as the final solution.

By including these details in the solution description, a CBR system can provide a comprehensive understanding of not just the predicted outcome but also the context, reliability and alternative possibilities related to the solution. This richness in solution representation enhances the system's ability to handle diverse problem-solving tasks across different domains and gives room for acceptability of the system.

Note that, while the structure of the base case consists basically of the problem description and the solution among others, the structure of the new case only consists of the problem description but without solution. Figure 2 represents a basic structure of a base case. The problem description is represented by a subset of problem features while each of the problem feature is represented by a principal attribute and a corresponding value. Hence, depicting an attribute-value pair.

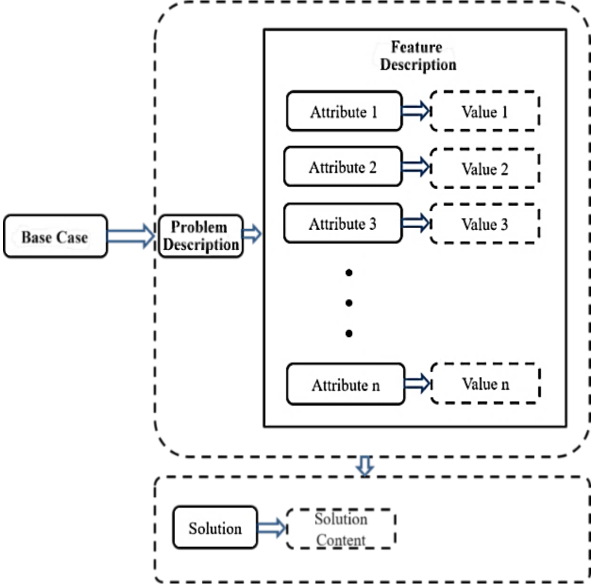

Figure 2: Simplified Structure of a Base Case

c) Outcome (o)

The outcome component in CBR refers to the result or resolution of a case's solution. This can be success or failure, or a numeric performance measure. It is critical for case reuse and adaptation. It enables the system to learn from both successful and failed cases, enhancing its reasoning over time [12]. This choice means accepting that some cases in the case base may contain incorrect or suboptimal solutions. These cases can still add value if they help prevent repeated mistakes. Failures can be analyzed and adjustments can be made to improve future problem-solving outcomes.

Keeping the outcome information can be beneficial for future reference or analysis, especially if they represent rare but significant scenarios. The general structure of a case can thus be represented as a triplet;

Case = {P, S, O}, where the outcome is optional depending on the domain [10].

2.3 Feature Selection and Weight Assignment

Feature selection aims to identify the most relevant attributes for case retrieval. Choosing the most relevant and informative features helps CBR systems to reduce computational costs and improve retrieval accuracy [15]. Selection can be manual (expert-driven), statistical (e.g., mutual information, chi-square) or automated using machine learning algorithms [15].

Feature selection methods can become computationally expensive and less effective when a large number of features are selected, this is referred to as curse of dimensionality. Also, selection of features based on a specific training dataset might not generalize well to unseen cases can lead to overfitting. Therefore, feature identification may use domain knowledge to extract a case description suitable for the case base system [12].

By carefully choosing the most relevant and informative features, CBR systems can make better decisions, solve problems more effectively thereby, gain users' trust. Weights are values assigned to attributes according the level of their importance, thereby allowing the system to prioritize certain attributes over others when determining the similarity between cases. Assigning weights to attributes will allow the CBR system to focus on the most relevant, informative and/or important features when retrieving cases from the case base [16]. This will in turn help to improve the accuracy and efficiency of the retrieval process.

Assigning appropriate weights to different attributes in the similarity retrieval measure is sort of a challenge. Assigning improper weights can result in biased similarity calculations which will lead to inaccurate case retrieval [17]. In many instances, multiple similarity measures may be available or required to capture different facets of similarity. Combining these measures or assigning appropriate weights to each measure can be challenging. Determining the relative importance of different measures and finding an effective combination technique is crucial for accurate retrieval and adaptation of cases.

The challenge lies in determining how to combine or weight different similarity measures to obtain a comprehensive and meaningful similarity score. The assignment of weights to case attributes can be done using different methods. The choice of weighting method depends on the specific problem domain and the availability of data and expert knowledge. The major weights assignment methods are [3, 18]:

a) **Expert-Based Weighting**: This method involves having domain experts assign weights to attributes based on their experience and knowledge of the problem domain.

b) **Data-Driven Weighting**: Uses statistical techniques to assign significance to the features based on analyzed historical data [18].

c) **Hybrid Weighting**: Combines expert-based and empirical or data-driven weighting techniques [3, 18].

By leveraging expert knowledge, data-driven analysis or a combination of both, CBR systems can effectively prioritize attributes and improve their decision-making capabilities [19]. Techniques such as Artificial Neural Networks (ANNs), genetic algorithms and inductive learning have also been proposed to solve feature weighting challenges in CBR [16].

2.4 Challenges in Case Representation

Despite the importance of case representation, the complexity of a domain can significantly impact how cases are represented. When representing cases with complex, rich, multidimensional data (such as medical diagnosis) it can be difficult to encode such data without losing important details. When important details are lost during case representation, then the efficiency of the CBR system is questioned. This is because, the retrieved case(s) may not be the required solution.

Also, in domains where available case data are incomplete or sparse, it will be difficult to construct meaningful case representations. Sparse datasets reduce the effectiveness of similarity measures and can lead to poor case retrieval performance [20].

Also important is the case base cardinality. As the number of cases in the case base increases, retrieval performance can degrade. Large-scale case bases require efficient indexing and retrieval mechanisms to maintain fast query response times. Traditional methods struggle with high-dimensional case spaces, leading to slower retrieval speeds and increased memory usage [21]. Approaches such as hierarchical case organization, clustering-based indexing and distributed case storage have been proposed to address this challenge [22].

Furthermore, the computational cost of case representation and retrieval can be high, particularly when using complex similarity measures. Feature selection and dimensionality reduction techniques, such as Principal Component Analysis (PCA) and feature weighting algorithms, help mitigate this issue by reducing the number of attributes considered in similarity calculations [23]. Additionally, hybrid approaches have shown promise in optimizing computational efficiency [24].

These challenges impact the system's performance and efficiency. Addressing these challenges is essential for improving the accuracy, speed and adaptability of CBR systems. Future research must now focus on developing more scalable, data-efficient and computationally optimized representation techniques to enhance CBR applications in various domains.

2. Similarity Measures in CBR

Similarity measures serve as a link between case representation and retrieval [2]. They determine how well a new (query) case matches the existing (base) cases in the case base. A well-chosen similarity function directly influences the quality and relevance of the solutions retrieved, making it a core determinant of the system's overall performance [6].

They are mathematical or computational functions that measures the similarity amongst pairs of problem descriptions of cases, $Sim(p_m, p_q)$, such that, the solutions of the base cases C_m can be used to find the solution of the query case C_q where, p_m represents the problem description of cases in the case base and p_q is the problem description of the query case. The higher the similarity score, the more relevant the base case is considered for reuse.

Most case-based reasoning agents select the best matching case(s) using heuristic functions or distances, which may be domain-specific [25]. Choosing the right similarity measures for a specific problem domain can be challenging. Different problem domains require different measures to capture relevant similarities between cases.

Retrieving the most similar cases involves searching the case base or library. If the case base is too small, there may be too few similar past cases, whereas a large case base can lower retrieval efficiency because the entire library must be searched. This is because on so many instances, the whole case base needs to be searched for the most similar case. Hence, there must be a balance in the quantity and quality of cases in the case library.

Multiple algorithms and techniques are employed in CBR systems for the purpose of retrieving cases from the case repository [26].
CBR relies on similarity rather than exact accuracy. Therefore, specific algorithms should be chosen based on case representation, attribute types, solution accuracy requirements and whether sequential or parallel search is needed [27].

3.1 Classical Measures

Several classical similarity measures are widely used in CBR systems due to their simplicity and effectiveness in structured data:

3.1.1 Euclidean Distance

Euclidean Distance is a widely used method for measuring similarity, defined as the straight-line distance between two points in Euclidean space, making it suitable for continuous or dense data [28]. It is most effective for continuous, normalized numeric features.

$$\boldsymbol{Dist(X_k, Y_k)} = \sqrt{\textstyle\sum_{k=1}^{n}(X_k - Y_k)^2} \qquad (1)$$

Where: *Dist()* is the distance that exist between the two compared vectors.

n is the number of attributes and *k* refers to the index of attributes in each case.

x_k, y_k represent the *k*th attribute in the vector *x* and *y* respectively.

When the similarity between the two compared vectors increases, *Dist()* decreases.

3.1.2 Manhattan Distance

Also known as City Block Distance, it sums the absolute differences of the coordinates. Suitable for high-dimensional spaces with sparse data.

$$\boldsymbol{Dist(X_k, Y_k)} = \textstyle\sum_{k=1}^{n}|X_k - Y_k| \qquad (2)$$

3.1.3 Minkowski Distance

Metrics like Euclidean and Manhattan Distances are specific cases within the broader Minkowski Distance when defined by varying the parameter “p”. The choice of values for "p" and the threshold significantly influences system accuracy [29]. For two feature vectors X and Y in n-dimensional space, the Minkowski's Distance is represented by:

$$\boldsymbol{Dist(X_k, Y_k, p)} = \left(\textstyle\sum_{k=1}^{n}|X_k - Y_k|^p\right)^{1/p} \qquad (3)$$

Where n is the number of input attributes.

Varying the value of p in equation (3): when p = 1, the Manhattan Distance is obtained and represented by equation (3). When the value of p = 2 in equation (1), the Euclidean Distance is obtained.

While the Minkowski similarity measure and its variants are based on distance between case attributes, some other form of similarity measures, e.g. the cosine similarity and Pearson correlation coefficient are based on the correlation of attributes.

3.1.4 Cosine Similarity

Cosine similarity is suitable for text or high-dimensional data [28]. It measures the cosine of the angle between two vectors in a vector space. For two feature vectors X, Y, the formula is given as:

$$\boldsymbol{Cos(\theta) = Cos(X, Y)} = \frac{X\ .\ Y}{||\ X\ ||\ ||\ Y\ ||} \qquad (4)$$

Where:

$$\boldsymbol{X\ .\ Y} = \textstyle\sum_{k=1}^{n} x_k y_k$$

|| ***X*** || is length of the vector ***x*** and represented as || ***X*** || = $\sqrt{\sum_{k=1}^{n} x_k^2}$

While || ***Y*** ||is the length of the vector ***y*** and represented by || ***Y*** || =$\sqrt{\sum_{k=1}^{n} y_k^2}$

θ is the angle between the vectors

x represents values from base case where $x = x_1, x_2, x_3, \dots x_n$

y represents values from target case where $y = y_1, y_2, y_3, \dots y_n$

Thus, || ***X*** ||and || ***Y*** ||represent the Euclidean norms of vector *x* and *y*

A cosine similarity of 1 indicates high similarity between features A and B, while a value of -1 shows dissimilarity [28]. Thus, two vectors aligned identically have a cosine similarity of 1, perpendicular vectors have a similarity of 0 and vectors in opposite directions have a similarity of -1. Greater alignment between vectors results in a higher ***Cos(θ)*** value.

3.1.5 Jaccard Similarity

It is used for binary or categorical attributes. The Jaccard index measures similarity between two sets based on their intersection, regardless of element types [30]. The similarity measure for the Jaccard coefficient is represented as:

$$\boldsymbol{S(X, Y)} = \frac{|\ x \cap y|}{|\ x \cup y|} \qquad (5)$$

where | ***x***| is the number of elements in ***X***, | ***y*** | is the number of elements in ***Y*** and | $\boldsymbol{x \cap y}$ | is the number of elements appearing jointly in ***X*** and ***Y***. The inner product is, in principle, unbounded. It calculates the intersection over the union of binary feature sets. And suitable for comparing text documents based on the presence or absence of words or terms.

3.1.6 Pearson Correlation Coefficient

Pearson’s correlation coefficient measures the linear relationship between two continuous variables, with values from -1 (perfect negative correlation) to +1 (perfect positive correlation) and 0 indicating no correlation [31]. And is defined as:

$$\textbf{Pearson(x, y)} = \frac{\sum_{i=1}^{k}(x_i - u_x)(y_i - u_y)}{\sqrt{\sum_{i=1}^{k}(x_i - u_x)^2}\sqrt{\sum_{i=1}^{k}(y_i - u_y)^2}} \tag{6}$$

where u_x and u_y are the means of all features in vectors *x* and *y*.

Each of these measures is suited for specific data types and domain conditions. However, their effectiveness may decline in heterogeneous or high-dimensional data environments.

3.2 Advanced Measures

To overcome limitations of classical methods, several advanced similarity measures have been proposed:

3.2.1 Classification Mahalanobis Distance

This distance measure accounts for correlations between features and variances within the data, making it suitable for multivariate analysis [32]. Mahalanobis distance was introduced by P. C. Mahalanobis in 1936. It is computed using the inverse of the variance-covariance matrix of data sets. It is useful to determine the similarity of an unknown sample set to a known one.

The Mahalanobis distance of a multivariate vector $x = (x_1, x_2, x_3, \ldots, x_N)^T$ from the values of a group with mean $\mu = (\mu_1, \mu_2, \mu_3, \ldots, \mu_N)^T$ and covariance matrix S, is defined as [32]:

$$\boldsymbol{Dist_M(x)} = \sqrt{(\boldsymbol{x} - \mu)^{\boldsymbol{T}}\boldsymbol{S}^{-1}(\boldsymbol{x} - \mu)} \tag{7}$$

The key difference between the Mahalanobis distance and the Euclidean distance lies in the consideration of correlations within the data set. While Euclidean distance only considers the individual variances of each variable ignoring potential relationships between them, Mahalanobis distance takes into account both the variances and correlations between variables and also uses the covariance matrix to weight the distances based on the underlying relationships within the data.

3.2.2 Fractional Function-Based Similarity

Another measure, the fractional function-based similarity measure is defined in equation (9). It offers more flexibility in similarity scaling by using fractional functions in distance computation [26].

$$\mathbf{SIM_{i0}^F} = \sum_{j=1}^{m} w_j \left(1 + \frac{|x_{ij} - x_{oj}|}{x_j^{max} - x_j^{min}}\right)^{-1} \mathbf{X\ 100} \tag{8}$$

$$\mathbf{SIM_{i0}^F} = \sum_{j=1}^{m} w_j \left(\frac{x_j^{max} - x_j^{min}}{x_j^{max} - x_j^{min} - |x_{ij} - x_{oj}|}\right) \mathbf{X\ 100} \tag{9}$$

Both Mahalanobis distance and fractional function-based similarity measures can be used with interval/ratio data, 0–1 data and ordinal data, as long as it is treated as interval data [32]. Note that $SIM_{i0}^F \geq SIM_{i0}^A$.

The potential of these measures warrants further investigation, especially for CBR applications.

3.2.3 Arithmetic Summation Similarity

It computes similarity using simple summation formulas. The formula for calculating the similarity between a new case (x_{oj}) and the base case (x_{ij}) for the j-th attribute as follows:

$$\mathbf{SIM_{i0}^A} = \sum_{j=1}^{m} w_j \left(1 - \frac{|x_{ij} - x_{oj}|}{x_j^{max} - x_j^{min}}\right) \mathbf{X\ 100} \tag{10}$$

$$\mathbf{SIM_{i0}^A} = \sum_{j=1}^{m} w_j \left(\frac{x_j^{max} - x_j^{min} - |x_{ij} - x_{oj}|}{x_j^{max} - x_j^{min}}\right) \mathbf{X\ 100} \tag{11}$$

The similarity is easily computed by using equation (11). While less complex, it can be adapted with weights for more nuanced retrieval [33].

3.2.4 Fuzzy-Based Similarity

This measure is useful in cases involving uncertainty or imprecise information. This is common in domains like medicine or decision support [34].

3.2.5 Semantic Similarity

Semantic similarity uses ontologies or concept hierarchies to compute similarity based on conceptual closeness rather than syntactic matching. Commonly applied in NLP and biomedical applications [7].

These advanced techniques provide better performance in complex environments, particularly those with noise, diverse data types, or nuanced semantics.

3.3 Hybrid Similarity Measurement

Traditional similarity measures (Euclidean, Manhattan, Minkowski, Mahalanobis, Hamming, Jaccard, Cosine and Levenshtein distances) generally focus on a single data type, such as numerical or categorical data. However, several real-world problems most times involve mixture of different data types, thereby making it challenging to accurately measure similarity using traditional methods.

This challenge is addressed by hybrid similarity measurement by combining multiple similarity measures to capture the complex relationships between cases with different data types.

For example, a hybrid model may use Euclidean distance for numeric attributes, Jaccard index for categorical features and cosine similarity for textual elements. Such hybridization allows for comprehensive comparison in heterogeneous domains such as healthcare, disaster response and finance [5], [13]. Hybrid models are often weighted, normalized or learned dynamically using optimization techniques.

The hybrid similarity measurement provides a powerful tool for improving the accuracy and applicability of CBR systems to real-world problems. Combining multiple similarity measures has helped CBR systems to effectively handle heterogeneous data, incorporate domain knowledge and improve retrieval accuracy in a wide range of domains [13]. As CBR continues to evolve, hybrid similarity measurement will play an increasingly important role in developing more effective and versatile CBR systems.

3.4 Adaptation of Similarity Measures

In many real-world applications, each problem domain has unique characteristics that affect case retrieval and reasoning accuracy, hence, similarity measures in CBR systems must be adaptable to different problem domains. A static similarity measure may not be suitable for all applications, necessitating adaptive mechanisms [33].

One commonly used approach to adaptation is dynamic similarity adjustment. This measure adjusts the similarity computations based on evolving data distributions or contextual factors such as attribute importance, domain-specific constraints (e.g., patient history in medical diagnosis) [35].

Learning-based adaptation is another approach. In this case, machine learning techniques such as neural networks or reinforcement learning are employed to optimize weights or parameters of similarity functions optimized on feedback from previous retrievals [36]. Additionally, fuzzy logic-based similarity measures can adjust dynamically to handle uncertainty and imprecise data, particularly useful in diagnostic systems domains like medical diagnosis where symptoms are often subjective [34].

The use of context-aware similarity measures can also enhance adaptability in CBR systems. This approach considers situational parameters or environmental factors in computing similarity. For example, in predictive maintenance, similarity measures must change based on the aging patterns of industrial equipment, as components degrade at different rates [37].

These adaptive mechanisms enhance CBR performance by continuously improving the relevance of retrieved cases.

3.5 Challenges in Similarity Measures

Despite the importance of similarity measures in CBR systems, they are hindered by some challenges. One major challenge is data type compatibility. Different domains utilize various data types, including numerical, categorical, textual, fuzzy, image, multimedia data etc. Standard similarity measures, such as Euclidean distance, perform well on numerical data but may fail to capture relationships in categorical or text-based data [38]. Hence, hybrid similarity measures that integrate multiple data representations are often required to improve retrieval accuracy [39].

Another challenge is data quality. In many real-world applications, case data may be incomplete, noisy or inconsistent which may lead to inaccurate similarity computations [6]. When there are missing values or incorrect attribute weights, similarity assessments can be distorted, making case retrieval unreliable. These issues can be mitigated by using techniques such as data pre-processing, imputation strategies and robust feature selection.

Many similarity measures, particularly those involving high-dimensional feature spaces or complex weighting schemes, can be computationally expensive [40].

Furthermore, domain-specific similarity requirements pose challenges as well. The effectiveness of a similarity measure depends heavily on the specific domain and application. There is need to address the choice of similarity measure for a specific domain. Some applications, such as legal reasoning, require semantic or logic-based similarity, which are not easily implemented with generic measures [41].

A lot of work has been put into improving similarity measures so as to ultimately enhance the efficiency and accuracy of CBR systems across diverse application areas [6, 26]. Addressing these challenges is very important to building scalable, adaptable and domain-relevant CBR systems.

3. Comparative Evaluation of the Interplay of Representation and Similarity Measures

The effectiveness of a CBR system depends significantly on the interplay between its case representation and similarity measurement components. To offer a comprehensive understanding, Table 1 presents a head-to-head comparison across major case representation techniques and similarity measures. This comparative evaluation is crucial for selecting suitable methods in specific application domains.

Table 1: Comparative Analysis of Case Representation Techniques in CBR

Representation Type	**Interpretability**	**Scalability**	**Domain Adaptability**	**Computational Complexity**	**Accuracy/Retrieval Effectiveness**	**Use Cases**
Feature Vector	High	High	Low	Low	Moderate	Recommender systems, IoT [10], [42]
Frame-Based	Medium	Medium	Medium	Medium	Moderate	Knowledge engineering [8], [10]
Object-Oriented	Medium	Medium	High	Medium	High	CAD, engineering design [11]
Predicate-Based	High	Low	High	High	High	Legal reasoning, medical diagnosis [10]
Semantic Nets / Ontology	High	Medium	Very High	High	High	Bioinformatics, NLP [7], [43]
Rule-Based	High	Low	High	Medium–High	Moderate	Expert systems, policy compliance [41]

Table 2: Comparative Analysis of Similarity Measures in CBR

Similarity Measure	**Interpretability**	**Scalability**	**Domain Adaptability**	**Computational Complexity**	**Accuracy/Retrieval Effectiveness**	**Use Cases**
Euclidean Distance	High	High	Low	Low	Moderate	Structured data, sensor input [28]
Manhattan Distance	High	High	Low	Low	Moderate	High-dimensional sparse data [29]
Cosine Similarity	Medium	High	Low	Low	High (for text)	NLP, text retrieval [28]
Jaccard Index	High	Medium	Low	Low	Moderate	Categorical, binary data [30], [41]
Pearson Correlation	Medium	Medium	Medium	Medium	High (linear domains)	Health analytics, finance [31]
Mahalanobis Distance	Low	Low	Medium	High	High	Anomaly detection, multivariate problems [32]
Fuzzy/Semantic	High	Low	High	Medium–High	High	Medicine, legal, complex domain matching [34], [44]
Fractional Function	Medium	Medium	Medium	Medium	Moderate–High	Cost estimation, discrete scoring [26]
Arithmetic Similarity	Medium	Medium	Low	Low	Moderate	Simple numeric comparisons [33]
Hybrid Function	Medium	Medium	Very High	High	High	Heterogeneous domains, medical, design [5], [13], [26]

4. Advancements and Applications of Similarity Measures in CBR

CBR has proven to be a versatile methodology across a wide range of domains. Over the years, the evolution of similarity measures has greatly enhanced the efficiency, accuracy and adaptability of CBR across various domains, including healthcare, finance, disaster management and industrial processes [2]. These significant advancements have transformed similarity measures from basic geometric calculations to sophisticated hybrid and adaptive models, thereby greatly improving accuracy, adaptability and domain-specific relevance.

Early CBR systems relied heavily on traditional metrics like Euclidean distance and cosine similarity. While computationally efficient, these methods assume equal feature importance and are prone to diminished performance in high-dimensional or heterogeneous datasets. They also fall short in capturing nonlinear or semantic relationships, which are common in real-world data [2].

To overcome these constraints, researchers developed hybrid similarity models that combine different computational strategies. For instance, [45] introduced a hybrid retrieval model using soft likelihood functions, significantly improving decision efficiency under uncertainty. These models, though more powerful, tend to increase computational overhead.

Recent advances incorporate machine learning (ML) into similarity computation to enhance adaptability. In finance, [46] reported improved stock prediction performance using Recurrent Neural Networks (RNNs) embedded within a CBR framework. However, such ML-driven similarity measures dependences on large, high-quality datasets which makes them vulnerable in low-resource domains. Overfitting and poor generalization may arise when datasets are imbalanced or sparse.

In domains requiring deep domain knowledge (such as healthcare, crisis response), ontology-based similarity models have emerged as powerful tools. These models incorporate structured semantic knowledge to assess similarity more contextually. For example, [7] developed a crisis response framework integrating syntactic and semantic similarity, boosting case retrieval in emergencies. While effective, ontology-based models demand well-structured ontologies, which are labour-intensive to build and maintain.

CBR has been widely used in financial decision support. [47] used asymmetrical similarity measures for loan assessment, refining credit risk modelling. [48] introduced a geometric similarity measure for stock forecasting, achieving better market trend recognition. Yet, these systems struggle under volatile economic conditions. Integrating real-time data streams and dynamic feature weighting can improve responsiveness.

CBR models like [9]'s STGA-CBR (Spatiotemporal Trajectory Similarity) improve disaster assessment by combining location and time-based similarity. Similarly, [49] proposed an adaptive model for gas explosion response. Industrial applications have seen tailored similarity measures to aid design and efficiency. [5] applied a hybrid similarity function for CNC turret design, boosting manufacturing performance.

Evidently, similarity computation continues to shape the future of CBR by pushing its applications further into intelligent, adaptive and domain-specific problem-solving. As domains become more dynamic and data-driven, future systems must evolve toward flexible, scalable and context-aware similarity models. This evolution, though complex, promises to expand CBR's relevance across even more critical and emerging fields.

5. Conclusion

CBR has proven to be a powerful and effective methodology for problem-solving across a wide range of domains. The success of a CBR system hinges on two critical factors: the representation of cases and the ability to accurately measure similarity between cases. Hence, this review presented a comprehensive analysis of case representation formats and similarity measures used in CBR systems.

The study categorized case representation techniques into feature vector representation and structured representation. Each offers distinct advantages and trade-offs in terms of interpretability, domain adaptability and computational cost. Likewise, similarity measures were grouped into classical, advanced and hybrid methods. Their effectiveness varies significantly depending on the data type, application domain and retrieval objectives.

Comparative evaluations across multiple criteria such as scalability, interpretability, computational complexity, applicability and adaptability revealed that no single method universally outperforms others. Instead, domain-specific requirements must guide the selection or combination of techniques. For example, medical and legal applications benefit from semantic and fuzzy similarity measures due to the ambiguity and structured knowledge involved, whereas recommender systems rely on scalable and efficient vector-based models.

The study suggests that hybrid approaches which combines multiple representation and similarity techniques can offer improved adaptability to real-world heterogeneous datasets. Also, semantic and ontology-based representations should be prioritized in domains requiring deep contextual reasoning, such as medicine and legal systems.

While significant progress has been made in the development of efficient case representation and similarity measures in CBR systems, further research should focus on adaptive similarity models that learn from system feedback and evolve with the domain. Additionally, there is a need for the development of benchmark datasets and standardized evaluation protocols to objectively compare CBR systems across domains.

It is important to establish synergy between robust representation and context-sensitive similarity computation. This is central to advancing the next generation of intelligent, interpretable and domain-aware CBR systems.

References

[1] Watson, I. (1998). Applying Case-Based Reasoning: Techniques for Enterprise Systems (1st ed.). Morgan Kaufmann Publishers Inc.

[2] Aamodt, A., & Plaza, E. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7(1), 39–59.

[3] Yan, A., & Cheng, Z. (2023). A Case Weighted Similarity Deep Measurement Method Based on a Self-Attention Siamese Neural Network. *Industrial Artificial Intelligence*, 1(1), 1–10.

[4] Bach, K., & Mork, P. J. (2020). On the Explanation of Similarity for Developing and Deploying CBR Systems. In Proceedings of the Thirty-Third International FLAIRS Conference (FLAIRS-33).

[5] Wang, H. Q., Sun, B. B., & Shen, X. F. (2018). Hybrid Similarity Measure for Retrieval in Case-Based Reasoning Systems and its Applications for Computer Numerical Control Turret Design. Proceedings of the Institute of Mechanical Engineers, *Journal of Engineering Manufacture*, 232(5), 918–927.

[6] Feuillâtre, H., Auffret, V., Castro, M., Lalys, F., Le Breton, H., Garreau, M., & Haigron, P. (2020). Similarity Measures and Attribute Selection for Case-Based Reasoning in Transcatheter Aortic Valve Implantation. *PLOS One*, 15(9), 1-21.

[7] Bannour, W., Maalel, A., & Ghezala, H. H. B. (2020). Case-Based Reasoning for Crisis Response: Case Representation and Case Retrieval, *Procedia Computer Science*, 176, 1063–1072.

[8] Kolodner, J. (1993). Case-Based Reasoning [eBook].

[9] Zhai, Z., Martínez, J. F., Martínez, N.L., & Díaz, V. H. (2020). Applying Case-Based Reasoning and a Learning-Based Adaptation Strategy to Irrigation Scheduling in Grape Farming. *Computers and Electronics in Agriculture*, 178(2020), 1-14.

[10] Yan, A., & Cheng, Z. (2023). A Review of the Development and Future Challenges of Case-Based Reasoning, *Applied Science*, 14(16), 1-22.

[11] Akhmadulin, R. K., Gluhih, I. N., & Karyakin, I. Y. (2016). An Object-Oriented Model of Case-Based Reasoning System Using Situations Tree. In 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT) (pp. 1-4). IEEE.

[12] López, B. (2022). Case-Based Reasoning: A Concise Introduction, Springer Nature.

[13] Geryk, M. (2023). Artificial Intelligence in Higher Education Industry: Just a Brief Introduction to Complexity of an Issue of Future Challenge. *Zeszyty Naukowe-Politechnika Śląska. Organizacja i Zarządzanie*, 172, 201–217.

[14] Atanassov, A., & Tomova, F. (2021). Application of Case-Based Reasoning in Biomedical Research, Science, and Education. *Science, Engineering and Education*, 6(1), 32–48.

[15] Liu, Q., Shi, S., Zhu, H., & Xiao, J. (2014). A Mutual Information-Based Hybrid Feature Selection Method for Software Cost Estimation Using Feature Clustering. In 2014 IEEE 38th Annual Computer Software and Applications Conference (pp. 27-32). IEEE.

[16] Khosravani, M. R., & Nasiri, S. (2020). Injection Molding Manufacturing Process: Review of Case-Based Reasoning Applications. *Journal of Intelligent Manufacturing*, 31(2020), 847–864.

[17] Ji, A. M., Zhu, K., & Huang, Q. S. (2012). Methods Determining the Weights of Characteristics in Mechanical Products Design on Case-Base Reasoning. *Applied Mechanics and Materials*, 138, 315-320.

[18] Kar, D., Chakraborti, S., & Ravindran, B. (2012). Feature Weighting and Confidence Based Prediction for Case Based Reasoning Systems. In Case-Based Reasoning Research and Development: 20th International Conference, ICCBR 2012, Lyon, France, September 3-6, 2012. Proceedings 20 (pp. 211-225). Springer Berlin Heidelberg.

[19] Sotnik, S., Deineko, Z., & Lyashenko, V. (2022). Key Directions for Development of Modern Expert Systems. *International Journal of Engineering and Information Systems (IJEAIS)*, 6(5), 4–10.

[20] Çelik Ertuğrul, D., & Elçi, A. (2020). A Survey on Semanticized and Personalized Health Recommender Systems. *Expert Systems*, 37(4), e12519.

[21] Li, W., Hacid, H., Almazrouei, E., & Debbah, M. (2023). A Comprehensive Review and a Taxonomy of Edge Machine Learning: Requirements, Paradigms, and Techniques. *Ai*, 4(3), 729-786.

[22] Zhang, Q., Hu, L., Shi, C., Liu, K., & Cao, L. (2022). Supervised Deep Hashing for High-dimensional and Heterogeneous Case-based Reasoning. *arXiv preprint arXiv:2206.14523*.

[23] Faisal, A., Jhanjhi, N. Z., Ashraf, H., Ray, S. K., & Ashfaq, F. (2025). A Comprehensive Review of Machine Learning Models: Principles, Applications, and Optimal Model Selection. *Authorea Preprints*.

[24] Lupiani, E., Juarez, J. M. & Palma, J. (2014). Evaluating Case-Base Maintenance Algorithms. *Knowledge-Based Systems 67*, 180-194.

[25] Burggräf, P., Wagner, J., & Weißer, T. Knowledge-Based Problem Solving in Physical Product Development—A Methodological Review, *Expert Systems with Applications: X*, 5(2020), 2020, 1-14.

[26] Ahn, J., Ji, S. H., Ahn, S. J., Park, M., Lee, H. S., Kwon, N., Lee, E. B., & Kim, Y. (2020). Performance Evaluation of Normalization-Based CBR Models for Improving Construction Cost Estimation. *Automation in Construction*, 119, 1–13.

[27] Zhang, L., Pan, Y., Wu, X., & Skibniewski, M. J. (2021). Artificial Intelligence in Construction Engineering and Management. Lecture Notes in Civil Engineering, 163, 231–256.

[28] Oyelade, O. N. & Ezugwu, A. E. (2020). A Case-Based Reasoning Framework for Early Detection and Diagnosis of Novel Coronavirus. *Informatics in Medicine Unlocked*, 20((2020), 1-22.

[29] Rumuy, A., Delima, R., Saputra, K. P. & Purwadi, J. (2023). Application of the Minkowski Distance Similarity Method in Case-Based Reasoning for Stroke Diagnosis. *JUITA: Jurnal Informatika*, 11(2), 323-332.

[30] Sujo, J. C. M. (2023). BRAIN L: A Book Recommender System. *Natural Language Processing Journal*, arXiv preprint arXiv:2302.00653, 1-24.

[31] Odemerho, J. O. & Odimegwu, T. C. (2023). Correlation Analyses of Particle Size Distribution and California Bearing Ratio of Lateritic Soil in Benin City. *Journal of Innovation and Technology*, 2023(10), 1-11.

[32] Ahn, J., Park, M., Lee, H., Ahn, S. J., Ji, S., Song, K., & Son, B. (2017). Covariance Effect Analysis of Similarity Measurement Methods for Early Construction Cost Estimation Using Case-Based Reasoning. *Automation in Construction*, 81, 254–266.

[33] Mathisen, B. M., Aamodt, A., Bach, K., & Langseth, H. (2020). Learning Similarity Measures from Data. *Progress in Artificial Intelligence*, 9(2), 129-143.

[34] Ahmed, M. U. (2010). A Case-Based Multi-Modal Clinical System for Stress Management. Doctoral dissertation, Mälardalen University.

[35] Pasha, S. J., & Mohamed, E. S. (2020). Novel Feature Reduction (NFR) Model with Machine Learning and Data Mining Algorithms for Effective Disease Risk Prediction. *IEEE Access*, 8, 184087-184108.

[36] Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction (2nd ed.). MIT Press.

[37] Xue, B., Xu, H., Huang, X., Zhu, K., Xu, Z., & Pei, H. (2022). Similarity-Based Prediction Method for Machinery Remaining Useful Life: A Review. *The International Journal of Advanced Manufacturing Technology*, 121(3), 1501-1531.

[38] Wani, A. A. (2024). A Review of Challenges and Solutions for Using Machine Learning Approaches for Missing Data. *International Journal of Engineering Applied Sciences and Technology*, 9(5), 36-50, ISSN No. 2455-2143.

[39] Huang, S. C., Shen, L., Lungren, M. P., & Yeung, S. (2021). Gloria: A Multimodal Global-Local Representation Learning Framework for Label-Efficient Medical Image Recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 3942-3951).

[40] Yuan, G., Zhai, Y., Tang, J., & Zhou, X. (2023). CSCIM_FS: Cosine Similarity Coefficient and Information Measurement Criterion-Based Feature Selection Method for High-Dimensional Data. *Neurocomputing*, 552, 126564.

[41] Ye, X. (2019). Similarity Assessment, *Association for the Advancement of Artificial Intelligence Association for the Advancement of Artificial Intelligence*, 1-7.

[42] Gu, W., Moustafa, A., Ito, T., Zhang, M., & Yang, C. (2021). A Case-Based Reasoning Approach for Supporting Facilitation in Online

Discussions, Group Decision and Negotiation, 30(2021), 719-742.

[43] Tang, M., Zhao, T., Hu, Z., & Li, Q. (2023). Research on Risk Prediction and Early Warning of Human Resource Management Based on Machine Learning and Ontology Reasoning. *Tehnički Vjesnik*, 30(6), 2036-2045.

[44] Shojaee-Mend, H., Ayatollahi, H., & Abdolahadi, A. (2024). A fuzzy ontology-based case-based reasoning system for stomach dystemperament in Persian medicine. *PLOS ONE*, 19(10), 1–15.

[45] Wang, Y., Fei, L., Feng, Y., Wang, Y., & Liu, L. (2022). A Hybrid Retrieval Strategy for Case-Based Reasoning Using Soft Likelihood Functions. *Soft Computing*, 26(7), 3489–3501.

[46] Bebarta, D. K., Das, T. K., Chowdhary, C. L., & Gao, X. (2021). An Intelligent Hybrid System for Forecasting Stock and Forex Trading Signals Using Optimized Recurrent FLANN and Case-Based Reasoning. *International Journal of Computational Intelligence Systems*, 14(1), 1763–1772.

[47] Benmessaoud, N., Adla, A., & Bella, A. B. (2019). A CBR Decision Support System for Loan Evaluation. *Journal of Digital Information Management*, 17(2), 75–86.

[48] Chun, S., & Ko, Y. (2020). Geometric Case Based Reasoning for Stock Market Prediction. *Sustainability*, 12(17), 1–11.

[49] Fan Z. P, Li Y. H., Wang X. & Liu Y., Hybrid Similarity Measure for Case Retrieval in CBR and Its Application to Emergency Response Towards Gas Explosion, *Expert Systems with Applications*, 41(5), 2014, 2526–253.