IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 17, 2020, accepted June 2, 2020, date of publication June 4, 2020, date of current version June 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3000139

A Novel Optimized Case-Based Reasoning Approach With K-Means Clustering and Genetic Algorithm for Predicting Multi-Class Workload Characterization in Autonomic Database and Data Warehouse System

NUSRAT SHAHEEN[1], BASIT RAZA[1], AHMAD RAZA SHAHID[1], AND HANI ALQUHAYZ[2]
[1]Department of Computer Science, COMSATS University Islamabad (CUI), Islamabad 45550, Pakistan
[2]Department of Computer Science and Information, College of Science in Zulfi, Majmaah University, Al-Majmaah 11952, Saudi Arabia

Corresponding author: Basit Raza (basit.raza@comsats.edu.pk)

ABSTRACT Data management systems are essential elements for any organization which is dealing with large volume of data now a days. Due to increase in data volume, and its complexities, it has become more challenging job for workload management system to maintain its performance. So, there is a need of such a system that can autonomically deal with such complexities with less or without human involvement. Performance of these systems can be improved by making the systems well-aware about the workload entering into the system. The workload of a prevalent typical database and data warehouse system can be characterized into three types that is Online Transaction Processing (OLTP), Decision Support Systems (DSS) and Mixed type of workload. Currently, autonomic characterization of workload into a binary class such as OLTP and DSS is being carried out as reported in the literature, however, characterizing the workload into three types that refers to a multi-class classification problem is relatively a more challenging task. In this study, we propose a novel optimized Case-based Reasoning (CBR) approach based on clustering for autonomically characterizing the workload into multi-class types before entering into the system. We implement four phases of CBR along with case-base generation and map it to the elements of autonomic MAPE-K model. In *Retrieve* phase, k-means clustering is used for enhancing retrieval efficiency and workload types predictions are made in *Reuse* phase. Genetic Algorithm is used in *Revise* and *Adapt* phase of CBR. Few autonomic self_* characteristics are incorporated to make it autonomic. We performed various experiments and results show that the proposed model outperforms in prediction as compared to existing approaches. We performed post-hoc test for the validation of results in comparison with other machine learning classifiers using the Friedman test that show that the proposed model stands out as the best classifier.

INDEX TERMS Workload management system, workload characterization, case-based reasoning, autonomic systems, Genetic Algorithm.

I. INTRODUCTION

With the increase in data volume, the complexity of data also increases, which results in the increase of difficulties regarding the data management. So, data management is getting beyond the human capability and encourages the development of intelligent systems. The role of Database Administrator (DBA) is to manage the database activities which also includes the handling of data. Due to dynamic and complex nature of data, humans cannot handle it in an efficient way. So, there is a need to develop intelligent systems with self-management capabilities for data handling.

The workload entering a data management system is non-deterministic in nature, so such systems are designed to use default settings for all types of workloads. If a system can know all characteristics of incoming data, all settings can be

The associate editor coordinating the review of this manuscript and approving it for publication was Rashid Mehmood.

optimized to achieve better performance and better utilization of resources.

Workload entering into the system can be categorized into two distinct types; in the first, case similar queries are involved which target day-to-day business operations and mostly consist of update, write, and delete operations; the second, involves decision making requiring more read and lesser number of write operations. The former is known as Online Transaction Processing (OLTP), and the send one is known as Decision Support Systems (DSS). OLTP has large number of target users while DSS targets a small number of users. A third type of workload can also be defined which carries properties of both the first two types and can be referred to as a Mixed workload type.

The type of workload cannot be predicted due to two shortcomings: one, DBA's inability to foresee the mix of OLTP and DSS in a workload, and the other is that as workload patterns shift, the Mixed workload is not automatically detected by the workload. As a result, this can cause a problem in handling Mixed type of workload. The effect of these causes leads to issues related to the inability of handling Mixed-workload type. Workload characterization has a significant role and importance in Autonomic Database Management System (ADBMS) and Data Warehouse (DWH) workload management, as well as in achieving improved performance tuning. Given its importance, workload characterization has received much attention from researchers over the past almost four decades [1].

Characterization of workload is a workload partitioning mechanism focused on specific characteristics and similarities. We can categorize the database workload into three different types: DSS, OLTP and Mixed. Workload detection is an important step prior to workload characterization. Workload detection involves the observation of any change occurring in the incoming workload. For all such scenarios where a change is observed in application, there is a requirement of re-analysis of DBMS configuration [2]. After the execution of workload in DBMS the values of some status variables change which are according to the previously executed workload. Hence, for every workload execution the value of status variable of DBMS changes from time to time.

The amount of change in the value of status variables before the workload entry and after the execution of workload is the actual cost of workload. Precisely detecting the workload of a database insures the detection of this change [3]. For characterization, an accumulative workload is transferred to the classifier at the detection stage with its corresponding resultant status variable values. Then, classifier classifies this workload into either DSS or OLTP. If results of earlier group of transactions contain results similar to DSS type workload, and if the results of later transactions represent like OLTP type workload, then this represents the workload shift from one type to another. Such characterization of the workload is the persuasive factor to finalize the values of a DBMS configuration parameter before a significant workload change is detected and granted to the classifier to perform required configuration again.

Knowing about the type of incoming workload before entering the system can be beneficial for handling the workload in a better way. However, existing studies provide workload characterization for only two types which are OLTP and DSS, and that too with less accuracy [4]. The system can optimize itself for the upcoming workload and can achieve better resource utilization. Different studies have used different workload features for the workload characterization in different DBMSs such as IBM, DB2, and MySQL [4]–[7].

Autonomic computing (AC) is the technology which promotes the concept of integrating the system with intelligence and self-management. AC technologies offer a number of characteristics for supporting the workload management. Some of these characteristics are: self-configuration, self-optimization, self-prediction, and self-optimization which can be blended to achieve autonomic workload management [8], [9]. Self-inspection monitors the incoming workload for workload feature extraction. Self-prediction forecasts the type of incoming workload for the system. And self-adaptation is achieved by adapting to the change in the behavior of incoming workload. AC technologies are based on MAPE-K architecture. The components of this architecture are *Monitor*, *Analyze*, *plan* and *Execute*. These components are linked to the knowledge base through feedback loop. In [10]–[13] authors reveal the benefits and impact of autonomic effects for database systems like SQL server, DB2, and Oracle. For autonomic system, the autonomic workload characterization can be achieved with the support of AC characteristics.

To cope with the varied environments, the system must be able to self-configure autonomically [1]. Availability of hundreds of configuration options in different systems make configuration a complex and time-consuming task in response to the changes in environment. For the management of such systems, there is need of highly knowledgeable and expert administrators, aware of all system configuration parameters and range of possible values for configuration parameters.

On the other hand, an autonomic system performs parameter tuning by itself along with the implementations of self-monitoring. System will upgrade its functionality by identifying and characterizing in accordance with new updates [14]. With the passage of time the database workloads are increasingly more varied and voluminous, so continuous monitoring of DBMS is required to check whether workload is configured optimally or not. So regular screening and analysis of workload is very important for the recognition of an autonomic DBMS [2]. Memory allocation for the workload varies according to the workload type such as DSS, OLTP and Mixed. A DBMS which recognizes its context whilst its operation and activities as well as has the ability to automatically configure itself for efficient processing of the workload, constitutes an autonomic DBMS [7].

Machine learning Lazy learning approaches are being used in building of autonomic data management systems either by characterizing the workload or by predicting the performance of the system [15]–[18]. Evolutionary algorithms are playing important part in optimizing the searching problems. Existing studies are using and building different evolutionary algorithms for problems from different domains [18], [19]. Due to inability of DBA to handle the multi-class classification problems which can be better handled by autonomic systems, there is a need of autonomic system multi-class characterization in optimized way.

The objectives of study are summarized as below:

- To propose a mechanism that can manage the workload autonomically by predicting three types of database and data warehouse workload.
- To develop an approach that solves the retrieval efficiency in workload management as the existing studies are using traditional approaches.
- To find the optimized solution by tuning the parameter using optimization techniques.
- To incorporate the autonomic characteristics that leads towards autonomic DBMS and DWH.

The following are the contributions of this study:

i. **Database and DWH Multi-class prediction and classification:** We propose a novel optimized case-based reasoning with clustering approach for workload characterization, that predicts the type of incoming workload as OLTP, DSS and Mixed. The proposed model deals with the prediction of these three types of classes. Hence, it addresses a multi-class classification problem.

ii. **Improved retrieval efficiency:** By incorporating the clustering approach into the traditional CBR approach, the proposed model achieved an improved retrieval efficiency. The cluster-based CBR approach predicts the solution for test cases and newly arriving cases. For prediction, already known similar solutions are retrieved from case-base which consist of clusters, by searching from matching cluster instead of whole case-base, which increases the efficiency of solution retrieval. When a new case arrives, to find the solution for that, matching case is searched within the relevant cluster instead of searching through all cases in the case-base. If a matching case is found with the similarity greater than the pre-defined threshold, then this solution will be reused, and prediction is done.

iii. **GA based Revise and Adaptation:** In case, for a new case if no match is found with respect to matching threshold, then a fresh attempt is made to find the solution in the *Revise* phase and also performs adaptation to the changes in workload. For the revision and adaptation phase, this study uses the GA for finding the optimized solution and to store in the case-base for future use.

iv. **Incorporation of AC characteristics:** Towards the development of autonomic database and data warehouse systems, this study incorporates five AC characteristics, which are: self-inspection, self-configuration, self-prediction, self-adaptation and self-optimization. By using these AC characteristics, our cluster-based optimized CBR autonomically performed all phases with rather less human involvement; however, DBA is required to be involved initially in the preparation of testing and training data.

Organization of the paper as shown in Table 1.

TABLE 1. Organization of paper.

Section	Section Title
II	RELATED WORK
III	METHODOLOGY
IV	PROPOSED OPTIMIZED CBR APPROACH
V	RESULTS AND DISCUSSION
VI	CONCLUSION AND FUTURE WORK

II. RELATED WORK

To achieve autonomic workload management, the research area is heading towards the concept of developing autonomic DBMS and DWH systems. Since decades, a large number of researches have been carried out to improve the DBMS and DWH performance to make systems autonomic. The study [20] provides the literature survey for autonomic workload management in all databases dealing with large volume of data such as DBMS and DWH. This study discusses the importance of autonomic systems aspects including classification, adaptation and performance prediction.

The studies [21], [22] offer a comprehensive survey on workload characterization. These highlight the issues of increase in workload concentration and its dynamic nature along with change in user's options to address the data. To address these issues, developing an autonomic system according to the demand and nature of data is an active research area. A taxonomy of workload management techniques is presented in [23], which classify workload management techniques into four major classes. It also stresses that for effective workload management, building autonomic systems is an important research gap. The study [24] proposed an autonomic framework which predicts the query arrival rate, it also uses clustering for classification of workload. This study doesn't consider other workload features which can affect the system performance.

Autonomic configuration and tuning of systems mainly depends upon workload characterization for the adjustment of parameters. There are many research studies which attempted to achieve autonomic characterization [7], [14], [25]–[27] but they autonomically classified the workload into only two type DSS and OLTP. A Mixed-workload type, blend of DSS and OLTP workload, may also be detected in the DBMS and DWH [5], [14], [26], which is also handled by DBA. With the exposure of Mixed workload type autonomic workload characterization should also detect the mix data type as separate workload type. The literature suggests that the database workload has a non-deterministic behavior [8]. Hence, we cannot achieve optimal performance by using the same DBMS configuration for Mixed-workload type [28]. An approach based on n-gram model is proposed in [2] to examine the workload shifts for autonomic databases. It detects long-term patterns as workload shifts, however these detected workload patterns cannot provide a suitable base for efficient workload characterization. In [3], authors used database status variables for workload characterization and prove that some status variables are better at capturing workload behavior in comparison with others. This study discusses the classification of workload as binary class classification problem and doesn't considers the workload shifts. The studies [14], [29] proposed an approach for autonomic workload management by considering characterization of workload, scheduling, and the idleness detection. An automated characterization of the workload is implemented in [4] by analyzing the system log. Workload characterization is also an important research aspect in cloud computing systems where it helps in many tasks like resource allocation, scheduling and self-management [30], [31].

A fuzzy rule based adaptive system is presented in [32] by considering two workload types and using three workload performance parameters which are *Buffer-Hit-Ratio, Number of Users* and *Database size*. This system is also based on binary classification of workload. Existing studies are also using Lazy learning approaches to develop autonomic systems. In [15], an AWPP framework is presented to predict the performance of workload before entering into the system based on workload characterization. This study uses the CBR approach for performance prediction and specifies that this performance prediction requires identification of workload type. This framework predicts performance on binary classification of workload but doesn't discuss the Mixed type workload. An autonomic performance prediction framework in DWH is presented in [16] using traditional CBR approach and for characterization considered only binary workload types. A CBR based workload characterization model is presented in [17], which considered the classification as binary classification and characterized the workload into OLTP and DSS. Authors in [33], presented a multistore database system ICARUS to handle the storage issues of different data types. It considered three types of workload viz. OLTP, DSS and Mixed, but only with data storage aspect.

Studies are also incorporating the clustering techniques for large scale data repositories for better handling the large volume of data. The study [16] presented clustering in their proposed performance prediction framework in DWH. For optimizing the solutions of searching problems, implementation of evolutionary algorithms can help in a better way. In [18], authors designed new differential evolutionary algorithm in CBR adaptation phase, and the study [19] presented a GA-based multi-objective optimization framework. This leads towards building such an autonomic model that predicts for multi-class characterization of DBMS and DWH workload and efficient prediction in search space in an optimized way.

III. METHODOLOGY

This section illustrates the methodology employed to conduct this study.

A. DATA PREPARATION

We used standard benchmark databases and workload provided by the TPC organization. For OLTP, it used TPCE database as well as TPCE standard workload, whereas for DSS, TPCH database and TPCH standard workload were used. It also involves designing of OLTP queries with the help of OLTP and DSS standard queries. During the execution of each query, it stores the values of all MySQL status variables, which are basically system performance parameters, 506 in number. After careful observation, it eliminated all those variables which don't show any significant change or no change at all during the entire workload execution. For the rest of variables, it also calculated information gain using Weka tool and selected performance parameters with higher ranked values. After feature selection, we executed large number of designed queries to obtain the values for selected performance parameters. The workload Feature Vector (WFV) contains the features which represent the characteristics of incoming workload. The values of these features vary according to the type of incoming workload and can be used for workload type detection.

In this study, we used the Workload Feature Vector (WFV) that can best represent any workload [17]. The WFV consists of features such as *number of sub-queries (Sq), number of equality predicates (Eq), number of selection predicates (Sp), number of non-equality predicates (Neq), number of joins (J), number of equijoin predicates (EqJ), number of non-equijoin predicates (NeJ), number of aggregation columns (Ag), number of sort columns (Sc).* These features helped to prepare the data of WFV and CFV used in our experiments.

B. FEATURE ENGINEERING

Feature selection and limited availability of data are the two common and crucial issues with machine learning methods which hinder their efficiency. To overcome these problems, "*autofeat*," a python framework for automatic features generation, is used that automatically generates non-linear features from input and then selects important ones from generated features [34]. This library uses combination of two or more features to generate new ones. For construction of non-linear features, it applies non-linear transformation to the features (for example $\log x$, $\sqrt{x}$, x^2, x^3 etc.), and then combines these pair of features using different operators (+, x, −). As a result, we obtained growing feature space. After that, different transformations are applied on the features to expand the size of features.

After the generation of new features, selection of important features that contribute meaningful information to the model is performed. Initially, the features having high correlation with the original features are removed and then Lasso LARS regression model, and Logistic regression models are used.

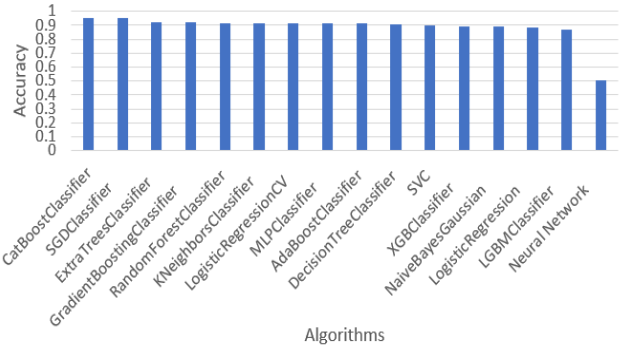

FIGURE 1. **Comparison of various algorithms w.r.t accuracies.**

These methods rank the features based on the training of model and selection of relevant ones. Only the relevant features are selected that contribute to the performance of model and other are discarded. After automatic features engineering, the selected feature space is trained and validated using 10-Fold cross-validation technique on different number of classifiers; among them *CatBoostClassifier* shows best results, with 0.95 accuracy. Whereas, Neural Network shows worst performance, with an accuracy value of 0.50. Figure 1 shows comparison of a number of classifiers with respect to their accuracies.

Algorithm 1 show the pseudo code for automatic features selection technique used by feature engineering. The input is the feature space that consists of CFV and output is the selected features.

With the application of feature engineering and selection method, it provides the following variables which are more helpful in predicting workload type: *Com_ratio (Cr), Innodb_log_writes (Ilw), Innodb_pages_read (Ipr), Innodb_rows_inserted (Iri), Questions* (*Q*) *and Sort_scan (Ss)*. Prediction of data type is made using the values of these features, as for different type of data these feature variables contains different values.

C. EVALUATION METRICS

Table 2 presents the metric of Actual vs Predicted. Where O, D and M refers to the OLTP, DSS and Mixed class respectively. TP_O, TP_D and TP_M refers to as true positive classes for OLTP, DSS and Mixed class respectively. NP_{OD} and NP_{OM} represents the negatively predicted classes, when actual class was OLTP, but the classifier predicted as DSS and Mixed respectively. NP_{DO} and NP_{DM} represents the negatively predicted classes, when actual class was DSS, but classifier predicted as OLTP and Mixed respectively. NP_{MO} and NP_{MD} represents the negatively predicted classes, when actual class was Mixed, but the classifier predicted as OLTP and DSS, respectively.

$$\text{Accuracy} = \frac{TP_O + TP_D + TP_M}{TP_O + NP_{OD} + NP_{OM} + NP_{DO} + TP_D + NP_{DM} + NP_{MO} + NP_{MD} + TP_M} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Precision\ of\ O = \frac{TP_O}{TP_O + NP_{DO} + NP_{MO}} \tag{3}$$

$$Precision\ of\ D = \frac{TP_D}{TP_D + NP_{OD} + NP_{MD}} \tag{4}$$

$$Precision\ of\ M = \frac{TP_M}{TP_M + NP_{OM} + NP_{DM}} \tag{5}$$

$$Precision = \frac{Precision\ (O) + Precision\ (D) + Precision(M)}{3} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$Recall\ of\ O = \frac{TP_O}{TP_O + NP_{OD} + NP_{OM}} \tag{8}$$

$$Recall\ of\ D = \frac{TP_D}{TP_D + NP_{DO} + NP_{DM}} \tag{9}$$

$$Recall\ of\ M = \frac{TP_M}{TP_M + NP_{MO} + NP_{MD}} \tag{10}$$

$$Recall = \frac{Recall\ (O) + Recall(D) + Recall(M)}{3} \tag{11}$$

$$F1 - score = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{12}$$

$$Error\ rate = \frac{NP_{OD} + NP_{OM} + NP_{DO} + NP_{DM} + NP_{MO} + NP_{MD}}{TP_O + NP_{OD} + NP_{OM} + NP_{DO} + TP_D + NP_{DM} + NP_{MO} + NP_{MD} + TP_M} \tag{13}$$

Algorithm 1 Feature Engineering

```
1: Input: A Feature space consisting of characterization Feature Vector
2: Output: Selected Features
3: Procedure:
4: for eachquery = 1, 2, ..., M do
5:    for features = 1, 2, ..., N do
6:       Combine features $feature_i$ and $feature_j$ to make $feature_k$
7:       Apply Transformation on $feature_k$
8:       for EachGeneratedFeature do
9:          Compare Generated Featurewith features to find correlation
10:         Remove Highly Corelated Features
11:         Find all features importance using Lasso LARS Regression and Logistic Regression Model and save them is a list
12:         Select most relevant features from list
13:      end for
14:   end for
15: Train These Fetures Space on different classification algorithms
16: end for = 0
```

TABLE 2. Actual vs Predicted.

		Predicted		
		O	D	M
Actual	O	TP_O	NP_{OD}	NP_{OM}
	D	NP_{DO}	TP_D	NP_{DM}
	M	NP_{MO}	NP_{MD}	TP_M

The Equations 1 to 13, as shown at the bottom of the previous page, were used for the calculations of Accuracy, precision, recall, f1-score, and error rate.

The similarity measures are presented by Equations 14 to 18. Five different types of similarity measures have been used to find the similarity between the test case and the cases stored in the clustered case-base. Following is the detail about each similarity measure:

$$Jaccard\ Distance(JD_{ij}) = \left|\frac{x_i \cap x_j}{x_i \cup x_j}\right| \tag{14}$$

Jaccard distance similarity metric finds the dissimilarity between two vectors. To this study, it is the measure of dissimilarity between test case and any case stored in case-base by the size of the intersection divided by the size of the union of the set of performance features, as given in Equation 14. In the equation, JD_{ij} represents the similarity between case i and case j.

$$cosine\left(c_{xy}\right) = \frac{\sum_{i=1}^{n} x_i \times y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \times \sqrt{\sum_{i=1}^{n} y_i^2}} \tag{15}$$

Cosine distance similarity measures the cosine angle between any case from case-base and the test case. The value of Cosine function is 1 for the 0 value of angle, and Cosine value is less than 1 for all other values. For two similar vectors the value of cosine similarity is 1. The greater the cosine angle, the lesser the similarity. The formula for cosine distance is given in Equation 15, c_{xy} represents the cosine distance between case x and case y.

$$Euclidean\ Distance(ED_{xy}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{16}$$

Euclidean distance similarity is used to measure the similarity between any existing case from the case-base and new input case for searching matching solution. The formula for calculating the Euclidean distance is given in Equation 16, where ED_{xy} represents the Euclidean distance between case x and case y.

Where $\sum_{i=1}^{n} (x_i - y_i)^2 = (x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2$

$$Canberra\ Distance(CD_{xy}) = \sum_{i=1}^{k} \frac{|x_i - y_i|}{|x_i| + |y_i|} \tag{17}$$

Canberra distance similarity is used to measure the similarity between test case and the stored case from case-base. The formula for Canberra distance is given in Equation 17, where CD_{xy} represents the Canberra distance between test case x and any case y from case-base.

$$Bray - Curtis(BD_{ab}) = \frac{\sum_{i=1}^{k} |x_{ai} - x_{bi}|}{\sum_{i=1}^{k} (x_{ai} + x_{bi})} \tag{18}$$

The Bray-Curtis dissimilarity provides robust and reliable dissimilarity results for various applications. In this study, this similarity measure is used to find the matching case from case-base for the test case based on the similarity value. The formula of Bray-Curtis measure is given in Equation 18, which calculates the similarity between case a and case b.

IV. PROPOSED OPTIMIZED CASE-BASED REASONING (CBR) APPROACH

In this section the proposed optimized case-based reasoning approach with k-means clustering and GA optimization is presented in detail. Figure 2 shows the autonomic perspective of the proposed framework. The workload is treated as a managed element. The sensors and effectors act as workload query features and workload performance metrics, respectively. It mapps four autonomic elements such as Monitor, Analyze, Plan, and Execute to the phases of CBR such as *Retrieve, Reuse, Revise* and *Retain (Adapt)*. The workload is monitored in the first step when new workload is tested in the system through WFV, and retrieval is performed and could be reused for prediction. It is followed by the workload analysis through the similarity matching. It aims at finding the best solution that can be the best match or could be optimized using heuristics and could perform revision when finding non-matching cases that have a value less than the specified threshold. Finally, the Execute acts to predict the type of workload.

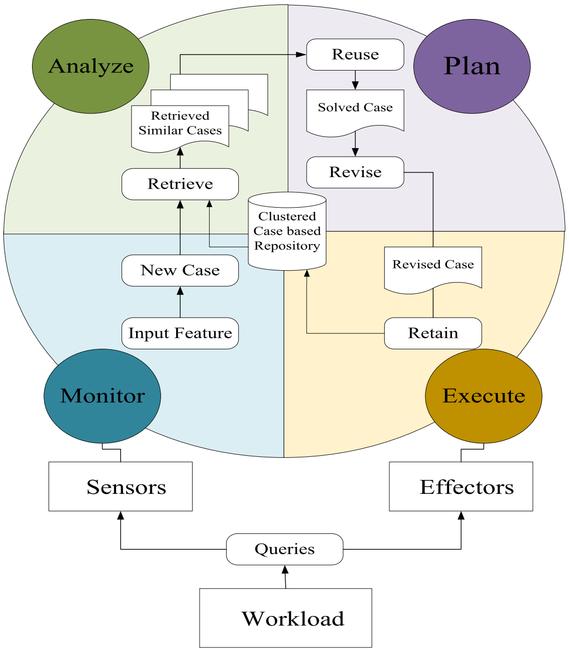

FIGURE 2. Autonomic perspective of CBR.

We are incorporating CBR for the prediction of three types of database workload.

Pseudo code of the methodology of Proposed Optimized Cluster-based CBR is shown below in Algorithm 2. It summarizes the working of the optimized cluster-based CBR approach. It works on two inputs, one is WFV which is extracted from workload and second is Data Set (DS). It identifies the relevant workload characterization features and using these features the type of incoming workload is predicted. First step is to construct the clustered case-base (CCB), once the CCB is developed the next step is Retrieve phase which returns the list of solutions having similarity according to defined threshold. If similarity is not according to defined threshold, then algorithm will start Revise phase along with retain phase. In case of non-empty list, the topmost solution will serve as predicted solution. The output of the algorithm is CFV and the type of workload could be OLTP, DSS and Mixed.

A. CBR PHASES

This study works with all four phases of CBR. Existing studies have implemented a traditional CBR-based approach. Here, first the case-base is developed. The incoming workload is treated as new case, called as test case. For workload characterization and type prediction, a new case is matched with existing cases in case-base and solution is retrieved. The best matching case is reused for prediction. When the case does not match with the set threshold value, it needs to revise. In the next phase, the revised case is retained for future use in the case base to achieve adaptation. Case base is the basic component of CBR system, which contains the already known cases along with their relevant classes.

Algorithm 2 Proposed Optimized Cluster-Based CBR

```
Input: WFV {Sq, Sp, Eq, Neq, J, EqJNeJ, Ag, Sc}, DS
Output: Characterization Feature Vector(CFV),
        Type prediction as OLTP, DSS, Mixed
Procedure
Clustered Case Base construction ( )
    //Algorithm 3 for CCB construction
SMList := Retrieve phase (new case)
//Algorithm 4 returns the list containing similar cases
if Len(SMList) >= 1 then
    Reuse phase ()            // Algorithm 5
else if
    Revise phase ()           // Algorithm 6
    Retain phase
end if
End of procedure
```

In CBR, case-based internal architecture is in the form of matrix; the row of matrix represents the case, whereas a column in the matrix represents the characterization features. A case can be represented as shown in Equation 19.

$$\text{Case}_i = (c_i k_1, c_i k_2, c_i k_3, \ldots\ldots\ldots\ldots c_i k_n; Li) \qquad (19)$$

Any case in case-base contains the values for different features selected to represent the data. In the equation 19, i represents the case number the value of $i = 0, 1, \ldots\ldots\ldots$ N. Where N is the total number of records in the case-base. The variable ck represents the data value of any feature, and n represents the total number of selected features. Li represents the class label of case i. The ability to adapt to changes in workload makes the CBR a suitable approach and it retains the changes, hence no need for retraining afresh when using this approach as compared to other machine learning approaches which require so.

Sample OLTP Workload

```
select S_SYMB
from INDUSTRY, COMPANY, SECURITY
where IN_NAME = 'Air Courier' and CO_IN_ID =
IN_ID and CO_ID between (4300000388 and
4300000308) and S_CO_ID = 4300000310;
```

Table 3 shows the example of a cases stored in the case base along with new case.

After the adaptation, the new items are also added to the case base. So, number of cases keeps increasing along with new adopted results, and hence making searching a more time-consuming step. To increase the retrieval efficiency, this study uses the cluster-based approach. All cases remain in their relevant cluster, as the system performs a search for the similar cases, only relevant cluster would be used for case matching. The pseudo code for clustered case-base generation is presented by Algorithm 3. We defined the similarity > 80. This is a user defined value which can be set

TABLE 3. An example of Case representation in the case base along with the new case.

Characterization Features	Com_ratio	Innodb_log_writes	Innodb_pages_read	Innodb_rows_inserted	Questions	Sort_scan	Workload type
New case	97.2	1459	48416	177703	3225	12	1
Case 0	95.8	876	48416	109238	3166	11	1
Case 1	147	22159	443681	2669835	4557	29	0
Case 2	5.0825397	36657	447667	4360025	7418	30	2

Sample DSS Workload

```
select l_shipmode, sum(case when o_orderpriority
= '1 – URGENT' or o_orderpriority
= '2 – HIGH' then 1 else 0 end) as high_line_count,
sum(case when o_orderpriority <> '1 — URGENT' and
        o_orderpriority <> '2 – HIGH' then 1
else 0 end) as low_line_count
from orders, lineitem
where o_orderkey = l_orderkey and l_commitdate < l_receiptdate
and l_shipdate < l_commitdate
group by l_shipmode
order by l_shipmode;
```

Sample Mixed Workload

```
Update p_retailprice = 5000
from partsupp, part
where p_partkey = ps_partkey
and p_brand <> 'Brand#34'
and p_type not like 'ECONOMY BRUSHED%'
and p_size in(22, 14, 27, 49, 21, 33, 35, 28)
and partsupp.ps_suppkey not in(
select s_suppkey
from supplier where
s_comment like '%Customer%Complaints%'
```

according to the level of similarity a user wants for accepting a solution. This value may vary according to the nature of data and sensitivity.

The four CBR phases employed in this study are detailed as follows.

B. RETRIEVE

This the first phase of CBR approach. As retrieval is important phase, therefore making it efficient is one of our objectives of the study. For a small size of data repositories, retrieval is not a big issue and does not affect the system performance as well as its response time. However, as this is an age of large-scale data repositories, therefore it could be a difficult and challenging task. In order to deal with a large-scale data, this study introduced clustering to make the search efficient for a new case, as it helps to save time when performing a new case search, because the solution matches the relevant cluster only.

Algorithm 3 Clustered Case Base Generation With K-Means Using Elbow and Monte Carlo Methods

```
Input: Case base with m cases, no of iteration n,
Output: Ciustered case base with N dusters
Procedure:
initial Accuracy := 0
for i := 1 to n do //Generate Random Clusters centroid
    N := i
    Accuracy_N := 0
    CCB := MakeCiuster(Case base, N)
    for each c_j in test case do
//Calculate cluster for each case in case base
            cluster = add(c_j, CCB)
            for each c_p ∈ cluster_i do
                simi := findSim(c_i, c_p)
//Bind Similarity betwen the cluster centroid and case p
            end for
            if simi > 80 then
                Accuracy_N := updateaccu()
            end if
        end for
        if Accuracy_N > initial Accuracy then
            Accuracy_N := updateaccu()
            N_opt := N
        end if
end for
CCB := MakeCluster(Case base, N_opt)
```

This phase is initiated with the arrival of a new case for prediction. For finding solution to the new case, the case base gets searched. Being a cluster-based case repository, therefore, depending upon cluster similarity, the relevant cluster would be searched for matching the cases. The cases having similarity greater than the required threshold would be retrieved, and topmost case will be used as solution. Pseudo code for case retrieval in Revise phase is given in Algorithm 4.

C. REUSE

The second phase is *Reuse*. In this, the retrieved case is reused for prediction. The cases retrieved could be more than

Algorithm 4 Retrieve Phase

```
Input: CCB, newCase/s NC, threshold m
Output: List of cases greater than threshold
Procedure:
MinDst := 100
for each ci ∈ CCB do
    Centpti ← centpt(ci)
    jDi = JD(NC, Centpti)
    if JDi < MinDst then
        MinDst := JDi
        cselec = ci
    end if
end for
for each case CJ ∈ ClusSelect do
    //ClusSelect is selected cluster for case matching
    JDJ = ]D(NCi,CJ)
    If JDJ > m then
        Append List with sim value
    end if
and for
return List
```

one. It uses the best match with the high similarity value or as per defined threshold. The prediction is performed here autonomically. It enables the self-prediction characteristics of autonomic computing. Algorithm 5 provides the pseudo code of Reuse phase as follows.

Algorithm 5 Reuse Phase

```
Input: List of selected cases
Output: location of best matching case from list
Procedure:
max := 0
for i := 1 to len (List) do
    If simi > max then
        max := simi
        loc = i
    end if
end for
Return location of best matching case
```

D. REVISE

Revise, the third phase, is performed when no match is found in the case-base on the required threshold. It tunes the solution from the case-base and revises it. This revised solution is used for prediction. In future, the revised case could be employed for reuse. Existing studies are based on top best match case for reuse, that are only based on similarity of new case with the case of the case-base; however, this study optimizes the search by using the Genetic Algorithm (GA) that selects the best case based on fitness values computed through fitness functions. We considered GA for optimization because it is a state-of-the-art optimization technique [19], [35], and it performed the best in this study as well. Figure 3 shows how GA based optimization is performed in this CBR-based approach. It works on heuristic case-base search using Crossover operator and Mutation operator.

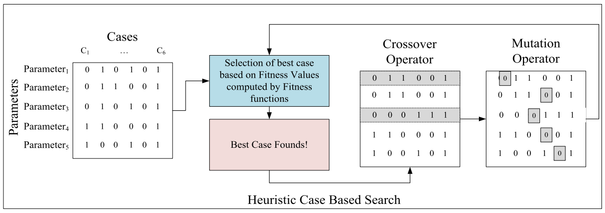

FIGURE 3. GA based optimization.

Algorithm 6 Automatic Case Generation using Genetic Algorithm: Revise and Retain Phase

```
Input: Clustered case base, Number of Iterations, Number of Individuals
Output: best revise matching case
Procedure:
Initialize the chromosome
Generate random individuals based on datasets
while Iteration <= No of iterations do
    Calculate fitness of individual
    Capture best individual having maximum fitness
    Apply Crossover on individuals
    Apply Mutation on individuals
end while
return best case
```

Algorithm 6 below provides the pseudo code of *Revise* Phase that generate cases after revision through GA.

GA is heuristic based searching technique to find accurate solution for optimization and searching problems. Here, GA is adopted for searching the optimized solution for the test case in CBR's revised phase. *Revise* phase gets functional if matching solution for the test case, according to predefined threshold, doesn't exist in case-base. GA starts with the generation of the population, randomly. Population consists of n number of individuals and each individual is represented by chromosome. Chromosome consists of set of parameters which represents the object of any system. In this scheme chromosome is the collection of performance features and is represented by the Equation 20.

$$\text{Chromosome}\,(X_i) = (x_1, x_2, x_3, \ldots\ldots\ldots, x_m)$$
$$where\ m = 6 \quad and\ i = 1, 2, \ldots\ldots.n \tag{20}$$

For random population generation, the values of parameters in the chromosomes are populated based on already available range of values from case-base. After random population generation, the fitness of every individual is tested to find the best optimized solution. In this scenario, the fitness function is shown in Equation 21 as follows.

$$fitness\,(f) = Sim\left(Case_t, Case_p\right) \geq th \tag{21}$$

where $Case_t$ represents the test case and $Case_p$ is any individual p from randomly generated population. And, th is the threshold, which can be set during implementation. The entire population would be searched to find the optimal solution. From all qualifying solutions, the solution with greater similarity would replace the previously selected solution. During the search for the best solution, the algorithm undergoes the mutation and crossover steps to produce variation in the next generation.

E. RETAIN

Retain, the fourth phase, is used in the proposed model and has a great impact in improving the efficiency. As in large-scale data repositories, there is an undetermined challenge to be dealt with arising from the dynamic behavior of workload. So, adaptiveness is a critical element of the proposed model. CBR itself provides adaptiveness due to its characteristic nature, therefore, this approach best suits to the workload management problem having non-deterministic changing behavior. The cases generated as a result of revise operation could be used in future. To overcome the issue of re-computation, the revised cases are adapted in the case-base. It incorporates the self-adaptation characteristics of autonomic computing in this phase.

V. RESULTS AND DISCUSSION

In this section we are providing the results that are obtained by performing extensive experiments and analyze the results of our proposed approach with discussion.

A. EXPRIMENTS AND RESULTS

The experiments were carried out on a Windows 10 on a 64-bit system, with a 4 GB RAM. MySQL 5.7 with InnoDB was employed to execute the workload. To validate the experiments, TPC benchmarks, introduced by [36], have been used. TPC-H benchmark supports the DSS workload, whereas TPC-E benchmark supports the OLTP workload. Firstly, we created both the databases along with data population and executed different workloads. In the initial phase, some queries were executed to get values to select the characterization features. MySQL database contains more than 500 variables which records system performance, but all are not equally important, so this study is based upon six features selected provided by the feature selection method. These selected features are Com_ratio, Innodb_log_writes, Innodb_pages_read, Innodb_rows_inserted, Questions and Sort_scan.

After features selection, it executed more workload and captured the results in the form of snapshots for DSS and OLTP workloads. For the Mixed workload, it executed the combination of OLTP and DSS workloads.

This model was implemented using Python 3.7.4 environment, and for the case-base it used clustering approach to increase the retrieval efficiency in the *Retrieve* phase. We incorporated GA in the clustered case-base for optimized search in the *Revise* and *Adapt* phase. Initially, CBR searches for solution for new case in the *Retrieve* phase inside a clustered case-base. If no match is found according to defined threshold, then solution is revised in the Revise phase. GA-incorporated approach produced better results.

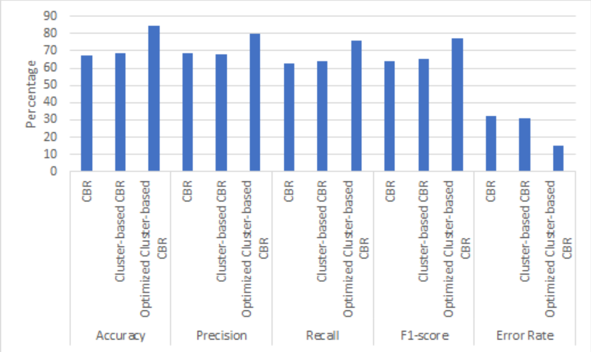

FIGURE 4. Comparison of optimized CBR with clustered CBR and traditional CBR using different similarity measures.

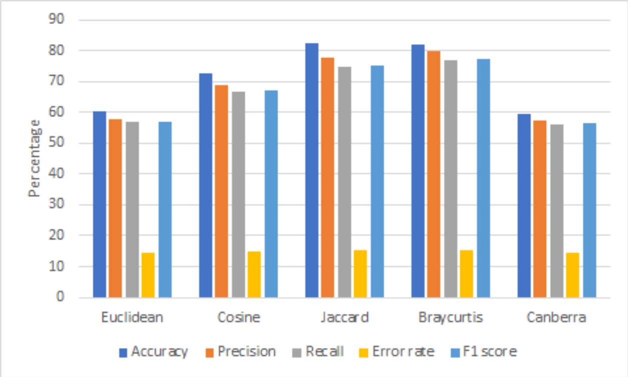

FIGURE 5. Comparison of different similarity measures.

For comparison of our model with traditional CBR, precision, recall, accuracy, and F1- score have been computed as performance measures. Figure 4 presents the comparison of all approaches, experimented in this study, using performance measures. The results demonstrate that optimized cluster-based CBR exhibits higher performance with 85% accuracy in comparison with 69% and 67% for CBR with clustering and traditional CBR respectively and low error rate as compared to the clustered and traditional CBR.

We have also implemented the proposed approach using different similarity measures to determine which similarity measure performed the best. Figure 5 presents the results for different similarity measures using different performance parameters.

The motivation behind the integration of clustering in the case-based was to make the retrieval of similar cases more efficient. During *Retrieve* phase, the system searches for the matching cases from relevant cluster inside the case-base data repository, rather than searching from entire data repository. As CBR is adaptive in nature, so it retains revise solutions for the potential future use, but it also increases the data volume resulting in more time to search from entire

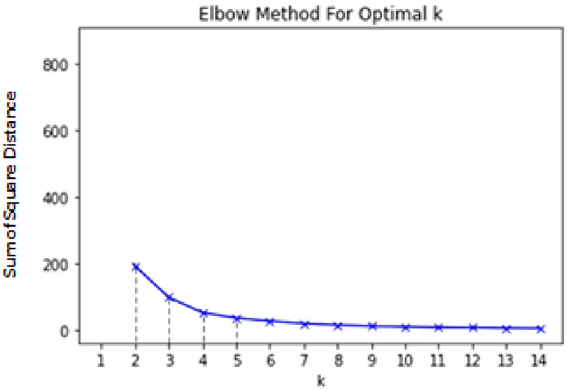

FIGURE 6. Finding optimal number of clusters k = 3 using elbow method k-means clustering.

case-base. The number of clusters represents the categories in which data can be differentiated. Our propose model engages k-means technique for achieving clustering. Existing studies employing clustering [17] refers to k-means as a reliable and robust technique for clustering. K-means clustering divides the entire data into k sets. Number of possible sets depends on the nature of data or on the plausible possibility of number of sets a data can have. For finding the optimal number of clusters, we calculate it dynamically using elbow method by setting the range of possible number of clusters from 2 to 10. K-means is an unsupervised learning technique; we automatically selected the value of k after performing clustering with different values of k using elbow method. In elbow method, the elbow bend shows the best value of k. It can be seen from Figure 6 the elbow bend is at k = 3. Therefore, we selected automatically k = 3 in this study and because of the nature of our data, this value of k creates three good clusters that belong to OLTP, DSS and Mixed workload. In contrast, K-Nearest Neighbors (KNN) algorithm is not considered in this study because, first of all, we need to fix the value of k. Further the computation cost is high as we need to compute distance of each query, every time for all training sets to obtain the specified number of neighbors for finding relevant class. The generalization of the algorithm is also not good due to its learning incapability from training data. As our data can be categorized into three basic types, at the maximum, hence it is equivalent to the number of optimized clusters.

As our dataset is nominal with three well-defined categories, and elbow method also suggests to set k = 3, hence the K-means runs for three clusters. Figure 7 shows the results of k-means clustering with the value of k = 3. Three distinct clusters, containing closer data points in each cluster, are in evidence. We also performed clustering by setting values of k = 2, k = 4, and k = 5, however the cluster formation did not turn out appropriately. Figures 8 depicts the results for k = 5, which demonstrates that higher k values generate unnecessary data groups.

Figure 9, 10, 11, and 12 present the Actual vs predicted values of the selected characterization features of CFV, for k = 3. Figure 9 shows the Actual vs Predicted of the

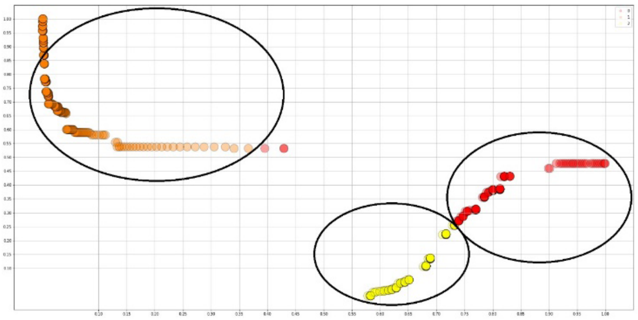

FIGURE 7. Clusters using k-means clustering with k = 3.

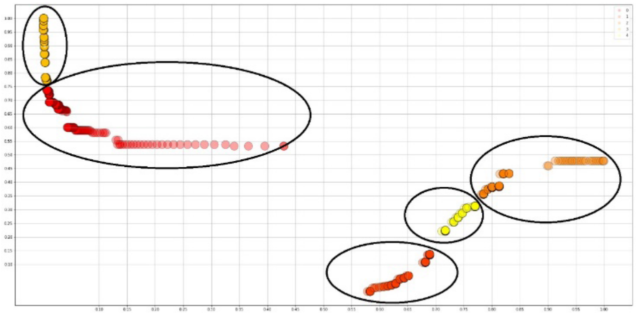

FIGURE 8. Clusters using k-means clustering with k = 5.

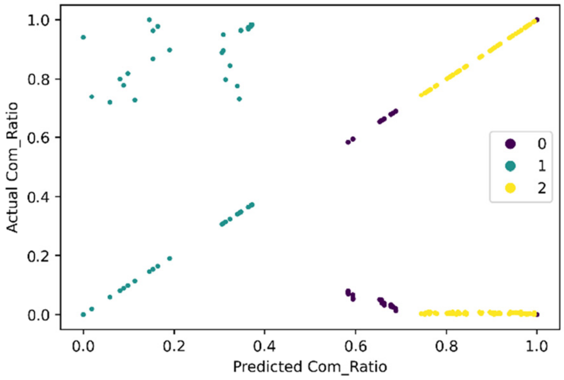

FIGURE 9. Actual v Predicted value for *Com_Ratio*.

characterization feature *Com_Ratio*. It can be seen that for the feature *Com_Ratio*, the proposed optimized cluster-based CBR approach offers better results. For another feature *Innodb_page_read* of CFV, the Actual vs predicted results, for k = 3, also produces better results. Similarly, for Innodb_rows_inserted and Sort_scan the Actual vs predicted results are also robust which shows that our proposed approach performs well in characterizing the workload into its types i.e. OLTP, DSS, and Mixed.

This study extends on the results obtained from the traditional CBR. For performance improvement, we integrated a clustering technique by achieving efficient search processing as well as incorporated GA to increase the ability to find a rather optimized solution. CBR is a lazy learning approach which doesn't require retraining as it calculates

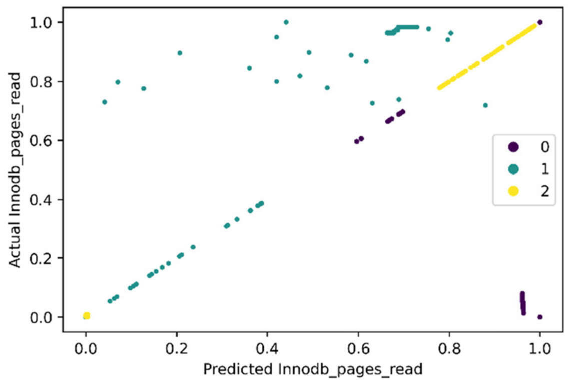

FIGURE 10. Actual vs Predicted value for ***Innodb_page_read***.

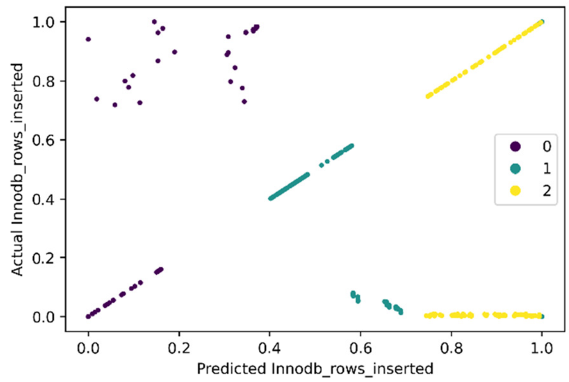

FIGURE 11. Actual vs Predicted value for ***Innodb_rows_inserted***.

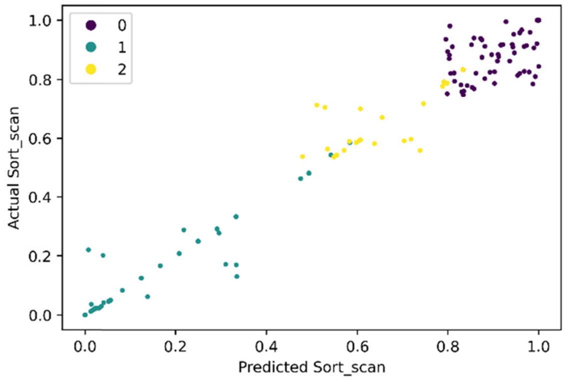

FIGURE 12. Actual v Predicted value for ***Sort_Scan***.

revise solutions for new unknown problems and stores the newly updated solution inside the knowledge repository, which are represented as the case-base. This model incorporates the different phases of autonomic perspective on the CBR to make it autonomous. Firstly, CBR incorporates the monitoring characteristics which inspect the incoming workload referred to as new case. Secondly, analysis is done to find the matching cases in the knowledge base. Thirdly, plan performs its function if it is required to recompute a solution for a new case which doesn't find its match in the existing repository. Finally, execute comes into action to retain the newly found solution in the knowledge repository, which contains the pre-calculated results or possible solutions.

1) VALIDATION THROUGH POST-HOC TESTS

For validation, different tests were performed by using keel software [37]. The Friedman test ***nxn*** is conducted which tests hypotheses in ***nxn*** comparison having logical relation among them. The corresponding probability ***p***-value with confidence α level is computed and returns ***p***-value for $\alpha = \mathbf{0.05}$ and $\alpha = \mathbf{0.01}$. For CFV, the Table 4 shows the Holm and Shaffer procedures for ***p***-value with $\alpha = \mathbf{0.05}$ which shows the ***p***-value and adjustment of α through these procedures. The z value, from the table of the normal distribution, is used to determine the corresponding probability (***p*** -value) that is compared with the level of significance α as well as adjusted for various comparisons.

We formulated and tested the null-hypothesis, and observe that the performance of classifiers is similar, but with a slight difference. There are three defined types of variability: variability among classifiers; error variability; and data sets variability. If the variability between classifiers is greater than the error variability, as well as there are differences between the classifiers, the post-hoc tests are applied to find the actual differences of classifiers. In this work, we have considered the workload features (WFV) and six classifiers i.e. proposed Optimized Cluster-based CBR, SVM, Simple Cart, Naï ve Bayes, Bayes Net, and J48, which are compared and evaluated for the hypothesis rejections and Adjusted p-values (APVs), by using the method provided in [38], [39].

The hypotheses are rejected with the unadjusted p-values 0.0038 and 0.0033, by Holm's and Shaffer's procedures, respectively for $\alpha = \mathbf{0.05}$. The hypotheses SVM vs. Naive Bayes and SVM vs. Simple Cart are rejected by Bergmann's procedure. The hypotheses rejected by Holm's and Shaffer's procedures with unadjusted ***p***-value 0.0077, 0.0067, respectively, for $\alpha = \mathbf{0.01}$. The hypotheses SVM vs. Bayes Net, SVM vs. Naive Bayes, SVM vs. Simple Cart, and SVM vs. J48 are rejected by Bergmann's procedure.

The significance of a statistical hypothesis test can be obtained using the information from its ***p***-value. The evidence against the null hypothesis becomes stronger for the smaller ***p***-values. Within multiple comparisons, a ***p***-value reflects the probability error for some comparisons, leaving the remaining comparisons. However, the APVs consider that the number of tests, according to chosen significance level, are conducted and compared. The APVs obtained through Shaffer Holm, Bergman, and Nemenyi for CFV are shown in Table 5. In this example, power difference among the test procedures can be seen. Table 5 shows the retainment or rejection state of a hypothesis.

The results convincingly establish that the proposed cluster-based CBR approach performs well as its search time has reduced due to the fusion of clustering technique. Case retrieval becomes efficient due the search being conducted from a specific relevant cluster rather than the

TABLE 4. **The P-values for $\alpha = 0.05$ for Holm and Shaffer w.r.t CF.**

I	Hypothesis	z = (R_0- R_i)/SE	p-value	p-Holm	p-Shaffer
1	SVM vs. Simple Cart	3.6080	0.0003	0.0033	0.0033
2	SVM vs. Naive Bayes	3.5412	0.0004	0.0036	0.005
3	SVM vs. Bayes Net	2.6058	0.0092	0.0038	0.005
4	SVM vs. J48	2.5390	0.0111	0.0042	0.005
5	Optimized Cluster-based CBR vs. Simple Cart	2.2717	0.0231	0.0045	0.005
6	Optimized Cluster-based CBR vs. Naive Bayes	2.2049	0.0275	0.0050	0.0050
7	Optimized Cluster-based CBR vs. SVM	1.3363	0.1814	0.0056	0.0056
8	Optimized Cluster-based CBR vs. Bayes Net	1.2695	0.2043	0.0063	0.0063
9	Optimized Cluster-based CBR vs. J48	1.2027	0.2291	0.0071	0.0071
10	Simple Cart vs. J48	1.0690	0.2850	0.0083	0.0083
11	Simple Cart vs. Bayes Net	1.0022	0.3162	0.01	0.01
12	Naive Bayes vs. J48	1.0022	0.3162	0.0125	0.0125
13	Naive Bayes vs. Bayes Net	0.9354	0.3496	0.0167	0.0167
14	Simple Cart vs. Naive Bayes	0.0668	0.9467	0.025	0.025
15	Bayes Net vs. J48	0.0668	0.9467	0.05	0.05

TABLE 5. **APVs obtained in the example by Holm, Nemenyi, Shaffer and Bergman w.r.t CFV.**

I	Hypothesis	p-value	APV-Holm	APV-Nemenyi	APV-Shaffer	APV-Bergman
1	SVM vs. Simple Cart	0.0003	0.0046	0.0046	0.0046	0.0046
2	SVM vs. Naive Bayes	0.0004	0.0056	0.0060	0.0046	0.0046
3	SVM vs. Bayes Net	0.0092	0.1192	0.1375	0.0917	0.0642
4	SVM vs. J48	0.0111	0.1334	0.1668	0.1112	0.0778
5	Optimized Cluster-based CBR vs. Simple Cart	0.0231	0.2541	0.3466	0.2310	0.2310
6	Optimized Cluster-based CBR vs. Naive Bayes	0.0275	0.2746	0.4119	0.2746	0.2310
7	Optimized Cluster-based CBR vs. SVM	0.1814	1.6330	2.7217	1.2701	1.2701
8	Optimized Cluster-based CBR vs. Bayes Net	0.2043	1.6341	3.0640	1.4299	1.2701
9	Optimized Cluster-based CBR vs. J480.229102	0.2291	1.6341	3.4365	1.6037	1.2701
10	Simple Cart vs. J48	0.2850	1.7103	4.2757	1.7103	1.7103
11	Simple Cart vs. Bayes Net	0.3162	1.7103	4.7435	1.7103	1.7103
12	Naive Bayes vs. J48	0.3162	1.7103	4.7435	1.7103	1.7103
13	Naive Bayes vs. Bayes Net	0.3496	1.7103	5.2436	1.7103	1.7103
14	Simple Cart vs. Naive Bayes	0.9467	1.8935	14.2009	1.8935	1.8935
15	Bayes Net vs. J48	0.9467	1.8935	14.2009	1.8934	1.8935

whole case-base. The proposed model is also compared with machine learning techniques and results clearly evidence that CBR performed the best in producing effective and accurate results. The proposed model is evaluated through computing the accuracy, effectiveness, significance, and adaptiveness measures. CBR manifests rather good predictive and adaptive ability which reduces the human intervention, a great advantage for efficient data management.

VI. CONCLUSION AND FUTURE WORK

In this study we proposed a novel optimized CBR approach with k-means clustering and GA optimization for characterizing the workload to autonomically manage database and data warehouse system. It characterizes the workload into three types that includes OLTP, DSS, and Mixed, referring to a multi-class classification problem. Incorporation of k-means clustering in the CBR, to make optimum clusters according to the nature of data has significantly increased the search and retrieval efficiency. The optimization is implemented by employing GA for computing the best optimized solution in its *Revise* and *Adapt* phases. For this model to become an autonomic system, few AC characteristics such as self-inspection, self-configuration, self-prediction, self-adaptation, and self-optimization have also been incorporated. The performance of the proposed model is compared with existing approaches and observed 16% accuracy improvement in comparison with CBR with clustering and traditional CBR approach. For validation, standard post-hoc test has been performed. The results demonstrate that this study achieves all its objectives and offers well defined novel contributions to the body of knowledge.

The future work could be to enhance the CBR model for performance modeling for large-scale data repositories through deep learning. Due to revision and adaptiveness, new cases are appended in the case-base. Therefore, case-base maintenance and efficient storage can be investigated to achieve improved results. Further, the learning scheme can also be improved by calibrating the learning model.

REFERENCES

[1] A. Mateen, "Workload management: A technology perspective with respect to self characteristics," *Int. J. Phys. Sci.*, vol. 7, no. 9, pp. 463–489, Feb. 2012.

[2] M. Holze and N. Ritter, "Autonomic databases: Detection of workload shifts with n-gram-models," in *Proc. East Eur. Conf. Adv. Databases Inf. Syst.* Berlin, Germany: Springer, 2008, pp. 127–142.

[3] Z. Zewdu, M. K. Denko, and M. Libsie, "Workload characterization of autonomic DBMSs using statistical and data mining techniques," in *Proc. Int. Conf. Adv. Inf. Netw. Appl. Workshops*, May 2009, pp. 244–249.

[4] M. Awad and D. A. Menascé, "Automatic workload characterization using system log analysis," in *Proc. Comput. Meas. Group Conf. Perform. Capacity*, San Antonio, TX, USA, 2015, pp. 1–11.

[5] S. Elnaffar, "Towards workload-aware DBMSS: Identifying workload type and predicting its change," Ph.D. dissertation, Queen's Univ., Kingston, ON, Canada, 2004.

[6] N. Huber, J. Walter, M. Bahr, and S. Kounev, "Model-based autonomic and performance-aware system adaptation in heterogeneous resource environments: A case study," in *Proc. Int. Conf. Cloud Autonomic Comput.*, Sep. 2015, pp. 181–191.

[7] S. Elnaffar, P. Martin, and R. Horman, "Automatically classifying database workloads," in *Proc. 11th Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2002, pp. 622–624.

[8] P. Manish and S. Hariri, "Autonomic computing: An overview," in *Proc. Int. Workshop Unconventional Program. Paradigms*. Berlin, Germany: Springer, 2004, pp. 257–269.

[9] J. Cámara, K. L. Bellman, J. O. Kephart, M. Autili, N. Bencomo, A. Diaconescu, H. Giese, S. Götz, P. Inverardi, S. Kounev, and M. Tivoli, "Self-aware computing systems: Related concepts and research areas," in *Self-Aware Computing Systems*. Cham, Switzerland: Springer, 2017, pp. 17–49.

[10] A. Mateen, B. Raza, T. Hussain, and M. M. Awais, "Autonomic computing in SQL server," in *Proc. 7th IEEE/ACIS Int. Conf. Comput. Inf. Sci. (ICIS)*, May 2008, pp. 113–118.

[11] A. Mateen, B. Raza, T. Hussain, and M. M. Awais, "Autonomicity in universal database DB2," in *Proc. 8th IEEE/ACIS Int. Conf. Comput. Inf. Sci.*, 2009, pp. 445–450.

[12] B. Raza, A. Mateen, M. Sher, M. M. Awais, and T. Hussain, "Autonomicity in Oracle database management system," in *Proc. Int. Conf. Data Storage Data Eng.*, Feb. 2010, pp. 296–300.

[13] B. Raza, A. Mateen, T. Hussain, and M. M. Awais, "Autonomic success in database management systems," in *Proc. 8th IEEE/ACIS Int. Conf. Comput. Inf. Sci.*, Jun. 2009, pp. 439–444.

[14] M. Abdul, A. M. Muhammad, N. Mustapha, S. Muhammad, and N. Ahmad, "Database workload management through CBR and fuzzy based characterization," *Appl. Soft Comput.*, vol. 22, pp. 605–621, Sep. 2014.

[15] B. Raza, Y. J. Kumar, A. K. Malik, A. Anjum, and M. Faheem, "Performance prediction and adaptation for database management system workload using case-based reasoning approach," *Inf. Syst.*, vol. 76, pp. 46–58, Jul. 2018.

[16] N. Shaheen, B. Raza, and A. K. Malik, "A CBR model for workload characterization in autonomic database management system," in *Proc. 14th Int. Conf. Emerg. Technol. (ICET)*, Nov. 2018, pp. 1–6.

[17] B. Raza, A. Aslam, A. Sher, A. K. Malik, and M. Faheem, "Autonomic performance prediction framework for data warehouse queries using lazy learning approach," *Appl. Soft Comput.*, vol. 91, Jun. 2020, Art. no. 106216.

[18] X. Yu, C. Li, W.-X. Zhao, and H. Chen, "A novel case adaptation method based on differential evolution algorithm for disaster emergency," *Appl. Soft Comput.*, vol. 92, Jul. 2020, Art. no. 106306.

[19] T. George and T. Amudha, "Genetic algorithm based multi-objective optimization framework to solve traveling salesman problem," in *Proc. Adv. Comput. Intell. Syst.* Singapore: Springer, 2020, pp. 141–151.

[20] B. Raza, A. Sher, S. Afzal, A. K. Malik, A. Anjum, Y. J. Kumar, and M. Faheem, "Autonomic workload performance tuning in large-scale data repositories," *Knowl. Inf. Syst.*, vol. 61, pp. 27–63, Oct. 2019.

[21] S. R. Shishira, A. Kandasamy, and K. Chandrasekaran, "Workload characterization: Survey of current approaches and research challenges," in *Proc. 7th Int. Conf. Comput. Commun. Technol. (ICCCT)*, 2017, pp. 151–156.

[22] M. C. Calzarossa, L. Massari, and D. Tessera, "Workload characterization: A survey revisited," *ACM Comput. Surveys*, vol. 48, no. 3, pp. 1–43, Feb. 2016.

[23] M. Zhang, P. Martin, W. Powley, and J. Chen, "Workload management in database management systems: A taxonomy," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1386–1402, Jul. 2018.

[24] L. Ma, D. Van Aken, A. Hefny, G. Mezerhane, A. Pavlo, and G. J. Gordon, "Query-based workload forecasting for self-driving database management systems," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, 2018, pp. 631–645.

[25] S. Elnaffar and P. Martin, "The psychic-skeptic prediction framework for effective monitoring of DBMS workloads," *Data Knowl. Eng.*, vol. 68, no. 4, pp. 393–414, Apr. 2009.

[26] S. S. Elnaffar and P. Martin, "An intelligent framework for predicting shifts in the workloads of autonomic database management systems," in *Proc. IEEE Int. Conf. Adv. Intell. Syst.-Theory Appl.*, Nov. 2004, pp. 1–8.

[27] Z. Ding, Z. Wei, and H. Chen, "A software cybernetics approach to self-tuning performance of on-line transaction processing systems," *J. Syst. Softw.*, vol. 124, pp. 247–259, Feb. 2017.

[28] S. S. Elnaffar, "A methodology for auto-recognizing DBMS workloads," in *Proc. Conf. Centre Adv. Stud. Collaborative Res.* Indianapolis, IN, USA: IBM Press, 2002, p. 2.

[29] A. Mateen, "Workload management through characterization and idleness detection," Ph.D. dissertation, Int. Islamic Univ., Islamabad, Pakistan, 2012.

[30] S. S. Gill and R. Buyya, "Resource provisioning based scheduling framework for execution of heterogeneous and clustered workloads in clouds: From fundamental to autonomic offering," *J. Grid Comput.*, vol. 17, no. 3, pp. 385–417, Sep. 2019.

[31] C. Liu, C. Liu, Y. Shang, S. Chen, B. Cheng, and J. Chen, "An adaptive prediction approach based on workload pattern discrimination in the cloud," *J. Netw. Comput. Appl.*, vol. 80, pp. 35–44, Feb. 2017.

[32] S. F. Rodd and U. P. Kulkarni, "Adaptive self-tuning techniques for performance tuning of database systems: A fuzzy-based approach with tuning moderation," *Soft Comput.*, vol. 19, no. 7, pp. 2039–2045, Jul. 2015.

[33] M. Vogt, A. Stiemer, and H. Schuldt, "Icarus: Towards a multistore database system," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2017, pp. 2490–2499.

[34] F. Horn, R. Pack, and M. Rieger, "The autofeat Python library for automated feature engineering and selection," 2019, *arXiv:1901.07329*. [Online]. Available: http://arxiv.org/abs/1901.07329

[35] F. Prado, M. C. Minutolo, and W. Kristjanpoller, "Forecasting based on an ensemble autoregressive moving average–adaptive neuro–fuzzy inference system-neural network–Genetic algorithm framework," *Energy*, vol. 197, Apr. 2020, Art. no. 117159.

[36] *Data Set*. Accessed: Apr. 20, 2020. [Online]. Available: http://www.tpc.org/

[37] *Keel Software*. Accessed: Apr. 20, 2020. [Online]. Available: http://www.keel.es./

[38] S. Garcia and F. Herrera, "An extension on 'statistical comparisons of classifiers over multiple data sets' for all pairwise comparisons," *J. Mach. Learn. Res.*, vol. 9, pp. 2677–2694, Dec. 2008.

[39] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

NUSRAT SHAHEEN is currently pursuing the Ph.D. degree with the Department of Computer Science, COMSATS University Islamabad (CUI), Islamabad, Pakistan. She is working as a Lecturer at the Department of Computer Science, COMSATS University Islamabad (CUI), Islamabad, Pakistan. Her areas of research interests include databases, data mining, machine learning, and autonomic workload management.

BASIT RAZA received the master's degree in computer science from the University of Central Punjab, Lahore, Pakistan, and the Ph.D. degree in computer science from International Islamic University Islamabad (IIU), Islamabad, Pakistan, in 2014, and conducted his Ph.D. research at the Faculty of Computing, Universiti Technology Malaysia (UTM), Malaysia. He is currently an Assistant Professor at the Department of Computer Science, COMSATS University Islamabad (CUI), Islamabad. His research interests include ad hoc wireless and sensor networks, database management systems, the IoT, security and privacy, data mining, data warehousing, machine learning, and artificial intelligence. He has authored several articles in refereed journals and serves as a reviewer for prestigious journals, such as the *Journal of Network and Computer Applications*, *Physical Communication*, *Networks*, *Applied Soft Computing*, *Swarm and Evolutionary Computation*, *Swarm Intelligence*, *Applied Intelligence*, and *Future Generation Computer Systems*.

AHMAD RAZA SHAHID received the Ph.D. degree in computer science from York, U.K., in 2012. He is currently working as an Assistant Professor at the COMSATS Institute of Information Technology, Islamabad, Pakistan. Since his Ph.D., he has been working in the areas of computer vision and pattern recognition, machine learning, and natural language processing. During his Ph.D. studies, he worked on automatically building a WordNet for four languages: English, German, French, and Greek. A few of the problems that he has worked on include cancer detection, pedestrian detection, driver fatigue detection, and data mining.

HANI ALQUHAYZ received the bachelor's degree in computer science and the master's degree in information systems management from King Saud University, and the Ph.D. degree in computer science from De Montfort University, U.K. He is currently an Associate Professor with the Computer Science Department, College of Science, Majmaah University, Saudi Arabia. He has authored several articles in high-impact journals, such as IEEE Access, *Sensors*, and *Wireless Communications and Mobile Computing*. His research interests include wireless security, scheduling, image processing, the IoT, security and privacy, data mining, machine learning, and artificial intelligence.