Computers, Materials & Continua
DOI: 10.32604/cmc.2022.025994
Article

Tech Science Press

An Efficient Scheme for Data Pattern Matching in IoT Networks

Ashraf Ali* and Omar A. Saraereh

Department of Electrical Engineering, Engineering Faculty, The Hashemite University, Zarqa, 13133, Jordan
*Corresponding Author: Ashraf Ali. Email: ashraf@hu.edu.jo

Received: 12 December 2021; Accepted: 24 January 2022

Abstract: The Internet has become an unavoidable trend of all things due to the rapid growth of networking technology, smart home technology encompasses a variety of sectors, including intelligent transportation, allowing users to communicate with anybody or any device at any time and from anywhere. However, most things are different now. Background: Structured data is a form of separated storage that slows down the rate at which everything is connected. Data pattern matching is commonly used in data connectivity and can help with the issues mentioned above. Aim: The present pattern matching system is ineffective due to the heterogeneity and rapid expansion of large IoT data. The method requires a lot of manual work and has a poor match with real-world applications. In the modern IoT context, solving the challenge of automatic pattern matching is complex. Methodology: A three-layer mapping matching is proposed for heterogeneous data from the IoT, and a hierarchical pattern matching technique. The feature classification matching, relational feature clustering matching, and mixed element matching are all examples of feature classification matching. Through layer-by-layer matching, the algorithm gradually narrows the matching space, improving matching quality, reducing the number of matching between components and the degree of manual participation, and producing a better automatic mode matching. Results: The algorithm's efficiency and performance are tested using a large number of data samples, and the results show that the technique is practical and effective. Conclusion: the proposed algorithm utilizes the instance information of the data pattern. It deploys three-layer mapping matching approach and mixed element matching and realizes the automatic pattern matching of heterogeneous data which reduces the matching space between elements in complex patterns. It improves the efficiency and accuracy of automatic matching.

Keywords: Internet of things; distributed computing; optimization; feature classification

1 Introduction

According to statistics from the Internet of Things (IoT) industry, the scale of industry reached 580 billion yuan in 2014, an increase of 18.46% year-on-year. In 2015, the global IoT market reached USD 62.4 billion, a year-on-year increase of 29%. By 2018, the global IoT equipment market size is expected to reach 103.6 billion USD, and the number of new IoT device accesses in 2019 will increase from 1.691 billion in 2015 to 3.054 billion [1–5].

With the rapid development of the IoT and the gradual maturity of the IoT platform, ubiquitous terminal equipment and facilities, such as smart sensors, mobile terminals, etc. are connected through the IoT, and the Internet of Everything (IoE) has become an inevitable trend [6–10]. The IoT technology that has emerged has been widely used in various fields and has produced a large number of differences structure data. The concept of "data gravity" is proposed for comparing data, software applications, interface services, etc. to stars, which contain their respective masses and densities [11–14]. The quality of data continues to increase and is much larger than other "stars", and other "stars" will be attracted by huge gravity and centered on the data "stars". It can be seen that the development of processing technology for heterogeneous data in the IoT will directly affect the development of the entire IoT technology progress [15–18].

At present, most of the heterogeneous data are stored independently and dispersedly in various regions, forming a large number of information islands, consisting of structured data (such as relational databases), semi-structured data such as extensible markup language (XML), hypertext markup language (HTML) and non-structured data. Structured data such as not-only structured query language (NoSQL) databases, pictures, videos and other forms of composition. How to interconnect these islands of information through "two collections" has attracted people's attention [19,20].

Pattern matching technology plays a vital role in the process of data interconnection [21]. Although heterogeneous data in the IoT can be interconnected through the development of a unified data interface standard, it is difficult to achieve in practical applications because it is difficult to achieve The transformation cost of the existing massive data storage model is huge. It can be said that the problem of data heterogeneity is inevitable. Data pattern matching technology provides a good solution to solve the above problems [22].

Pattern matching is a construction source model and goal The process of mapping relationships between elements in the pattern, and traditional pattern matching operations are mostly done manually by information technology (IT) technicians. With the expansion of data scale and the increase of pattern complexity, manual matching will consume huge manpower and material resources, and is easy to destroy data integrity and accuracy [23]. At present, the existing research results use element information, semantic information, data instance information and structural information to mine the correct element mapping relationship.

The heterogeneous data conversion model of the IoT realizes fast and efficient interconnection of heterogeneous data in the IoT. However, the manual participation in the pattern matching process in the interconnection process is too high and the matching efficiency is low [24]. In order to solve this problem, we first analyzed the heterogeneous IoT characteristics of the data, that is, the time series characteristics and using this as the starting point. Through the analysis of time series data to classify and recognize the heterogeneous data of the IoT, it provides the possibility for automatic pattern matching [25].

Based on the above research results, this paper proposes a hierarchical data matching algorithm for heterogeneous data in the IoT. The proposed algorithm is based on the instance information of the data pattern and uses three-layer mapping matching: feature classification matching, relational feature clustering matching, and mixed element matching. It realizes the automatic pattern matching of heterogeneous data in the IoT, reduces the matching space between elements in complex patterns, and improves the efficiency and accuracy of automatic matching.

2 Literature Review and Problem Analysis

2.1 Literature Review

At present, the more typical and widely used matching algorithms include auto-match, Clio, computer operations management associations (COMA), internet messagge access protocol (iMAP), communications provider identifier (Cupid), large-scale distributed (LSD), graphical user interface logic user engineering (GLUE), etc. Reference [26] uses pattern information such as attribute names and data types to calculate the semantic similarity between attributes, Combine semantic similarity and structural similarity for matching. Reference [27] uses a variety of matchers to work together to improve the accuracy and efficiency of matching results through filtering and screening. Reference [28] applies machine learning algorithms to achieve matching and automatically integrate the matching results, and find 1:1 and 1:*n* matches well. Reference [29] deploys the iMAP method of pattern matching based on data instance information is a comprehensive utilization of multiple types of information in the pattern and simultaneously obtains the pattern between the patterns. The method of simple and complex mapping relationship can find the match of 1:1 and 1:*n* very well.

2.2 Analysis

Reference [30] proposed a database pattern matching method based on entity classification, which extracts features according to the semantic information of the data instance information, uses the naive Bayes algorithm to classify the features, and finally performs corresponding pattern matching on the sub-patterns. The massive and heterogeneous nature of IoT data, no matter how standardized the database design is, the definition of its attribute names and meanings varies from person to person. It is not advisable to use only the attribute names and other semantic information for fuzzy classification, because this classification is only based on human understanding rather than the essential characteristics of the source data. Reference [31] uses the statistical characteristics of the probability distribution of the data to extract the data characteristics from another angle, and uses the BP neural network algorithm to improve the efficiency and accuracy of pattern matching. However, the algorithm is only for the matching between 1:1 column, and is not suitable for complex pattern matching. As the data size continues to increase, the matching space and the number of matching will also increase sharply, resulting in low matching efficiency. As compared with above literature, the proposed algorithm can better solve the problem of the difficulty in automatically obtaining and analyzing data source pattern information in real applications. By analyzing massive heterogeneous raw data, according to the time series of IoT data characteristics classify the analyzed characteristics to represent the pattern characteristics of the data source. Secondly, the relational feature clustering method in the (hierarchical sequence matching attributes) HSMA algorithm is innovative, especially for the feature clustering of the sensor data of the IoT. The impact of the difference of different data types between the data set data. Finally, it uses a layered method to gradually reduce the matching space, reduce the number of matching, and improve the efficiency of matching.

3 Proposed Framework

In order to realize the intelligentization of the pattern matching of heterogeneous data (structured data) of the IoT, we designed a pattern matching algorithm HSMA. For the input of an unknown source data, we first compare the characteristics of time series and all types data sets are classified to obtain sub-pattern collections. At the same time, through the previous research results from IoT field recognition algorithm based on time-series data (IFRAT) algorithm, the domain feature information of unknown data sources is obtained, and the corresponding domain standard set and standardized database are initialized according to the different domains for relational feature aggregation. Class matching is performed to establish the mapping relationship between the source data set and the target database in various types (1st level matching). Then according to the extracted data set characteristics, use machine learning algorithms to cluster the data sets in each sub-pattern, to further reduce the matching space and scope, calculate the similarity between the clustering results and the corresponding standard set data set, and establish a matching mapping (second-level matching). Finally, the elements in the matching clusters are mixed and the similarity calculation is performed (3rd layer matching), which produces pattern matching results between the source data and the target database. Through the above three layers of matching, the proposed algorithm gradually reduces the matching space, reduces the number of matches, and thus improves the matching efficiency. The overall architecture of the proposed algorithm is shown in Fig. 1.

3.1 Source Data Normalization

There are multiple types of data in each data set. In the past, the pattern matching algorithm used to process different types of elements in the same data set separately. There are various processing methods. Although it can retain the characteristics of a single element, it destroys the association between different types of elements and losing the characteristics of the entire data set. Therefore, for the common data formats (such as values, characters, dates, etc.) in the data set, it is necessary to find a unified data processing method to ensure the integrity of its characteristics. First, the structured data set is regarded as a two-dimensional matrix with a row-column relationship. The difference change result of adjacent row elements in the matrix is used as the row of the new matrix. We turn the new matrix into a relational matrix. The difference change result is used -, 0, + means that through this method, we replace the original content value with the relationship value, convert different types of data into the same standard, and finally get the relationship matrix. As a standardized form of the data set, the relationship matrix can be better ground is used for feature mining of the data set.

In order to better understand the proposed algorithm, we define some terms used in this paper.

Definition 1: Relation matrix. For any common structured data $\boldsymbol{D}_{m\times n}$, the difference can be done through adjacent rows. The difference result is represented by the three symbols -, 0, +. We define this matrix relationship matrix with M expressed.

$$\boldsymbol{D}_{m\times n} = \begin{vmatrix} D_{11} & \cdots & D_{1n} \\ \vdots & \vdots & \vdots \\ D_{m1} & \cdots & D_{nm} \end{vmatrix} \begin{vmatrix} D_{21} - D_{11} & \cdots & D_{2n} - D_{1n} \\ \vdots & \vdots & \vdots \\ D_{m1} - D_{(m-1)1} & \cdots & D_{nm} - D_{(m-1)n} \end{vmatrix} \tag{1}$$

$$\boldsymbol{M} = \begin{vmatrix} + & \cdots & - \\ \vdots & \vdots & \vdots \\ 0 & \cdots & - \end{vmatrix} \tag{2}$$

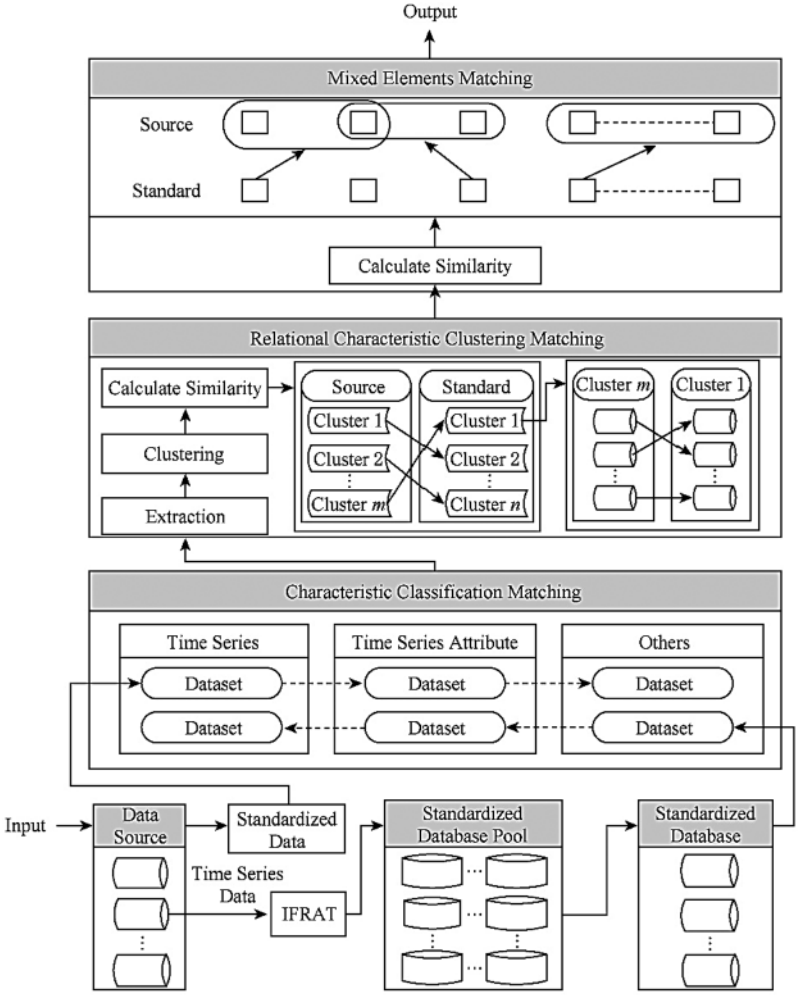


Figure 1: Proposed architecture

Definition 2: Relational key pair. The key pair formed by the pairwise combination of elements in each row of the relational matrix $\boldsymbol{M}$.

Definition 3: Characteristic column. For any column of the relational matrix $\boldsymbol{M}$, if and only if two conditions are met at the same time: 1) The value type of the column element is a date type or other data types that can be converted into a date type; 2) the column The element "+" or element "-" in the frequency of occurrence is greater than the threshold θ (we take $\theta = 90\%$), such a column is called a time series feature column. If only condition 2 is met, the column is called the main feature column.

Definition 4: Time series data. The raw data generated by IoT sensors are stored in a structured form in chronological order. Such a data set is called time series data.

3.2 Three-Layer Map Matching

This section gives a detailed description of proposed algorithm, which is divided into three layers. The first layer is feature classification matching (time series features and data type features). The second layer is relational feature cluster matching. The third The layer is mixed element matching. Based on the idea of layering, we divide the complex heterogeneous data pattern matching process into three steps, and gradually narrow the matching space. For each step of matching, it is based on the characteristics of the heterogeneous data of the IoT. The similarity between the source data and the standard data, build the mapping relationship, and finally realize the automatic matching of the heterogeneous data pattern of the IoT.

3.2.1 Feature Classification Matching

The proposed algorithm provides two classification methods in the feature classification matching process:

1) Time series feature classification

Time series is a hallmark feature of IoT heterogeneous data, which can be used to classify IoT heterogeneous data. First, we divide IoT heterogeneous data into three categories according to the time series characteristics:

a) Time series data category P. All data sets containing time series feature columns belong to the time series data category (such as sensor network sensor data).
b) Time series attribute class Q. The data set containing the main feature column belongs to the time series data attribute class (such as sensor network monitoring object metadata), such as related information describing time series attributes.
c) Non-time series R. Contains data sets that have nothing to do with time series, and generally consist of many relevant auxiliary information.

The proposed algorithm classifies the heterogeneous data of the IoT according to time series characteristics, and finally obtains the data classification set {time series data class, time series attribute class, non-time series class}, and the specific steps are shown in Algorithm 1.

2) Data type feature classification

In the heterogeneous IoT data, the source data collections in different fields have different data type distribution characteristics. According to this feature, the heterogeneous IoT data is classified. The data types are divided into five categories according to the common types of heterogeneous IoT data: Numerical value Type (value), character type (char), time type (date), character-numerical type (char-value), character-time type (char-date). Considering that the same data may be assigned under complex pattern matching for different data types, the mutual conversion between data types will cause their distribution characteristics to deviate. Therefore, the conditions that can be converted between string and numeric values are included in the data type characteristics. Use the frequency value of each data type to construct a 5 dimensional feature vector, and then classify the heterogeneous data of the IoT according to the feature vector:

a) Time-dominant category E. Due to the particularity of the time type, as long as the data set containing the time type belongs to the time-dominant category.
b) Numerical-dominant category F. Data sets with significant numerical types (such as int, long, float, double, etc.) belong to the numeric-dominant category.
c) Character-dominant class G. Data sets with a significant proportion of character types (such as char, varchar, nvarchar, etc.) belong to the character-oriented class.

The proposed algorithm classifies the heterogeneous data of the IoT according to the characteristics of the data type, and finally obtains the data classification set $\{E,\ F,\ G\}$, and the specific classification process is shown in Algorithm 2.

Algorithm 1: Time series feature classification algorithm

Input: IoT heterogeneous data collection $X = \{x_1, x_2, \ldots, x_n\}$ (structured data)

Output: Time series feature classified set $\{P,\ Q,\ R\}$

```
Obtain the structured data
Normalize $X$ to obtain the corresponding relationship matrix set $Y = \{\boldsymbol{M}_1,\ \boldsymbol{M}_2, \ldots, \boldsymbol{M}_n\}$
Traverse each column of $x_1$; $\forall x_i \in X$ according to the time series feature classification
for each $x_i \in X$ do
    for each $col$ belonging $\boldsymbol{M}_i$ set of columns do
        $n = Column\_num(\boldsymbol{M}_i)$
        $\theta \leftarrow \max \left| \frac{Count(+)}{\sum_{i=1}^{n} i}, \frac{Count(-)}{\sum_{i=1}^{n} i} \right|$
        if $\theta \geq 90\%$
            if $Col \in Type.date$
                $x_i \in P$
            else
                $x_i \in Q$
            end if
        else
            if all the columns of $x_i$ are traversed to end
                $x_i \in R$
            end if
        end if
    end for
end for
return $\{P,\ Q,\ R\}$
```

3.2.2 Feature Cluster Matching

Through the feature classification matching in Section 3.2.1, the time series feature classification and data type feature classification are performed on the source data S and the data in the standardized database T respectively, and the result sets obtained are collectively referred to as S_{cla} and T_{cla}. In this section, we use the SOM clustering method to classify the data sets in S_{cla} and T_{cla} to further reduce the matching space. In the clustering process, we use the relation matrix $\boldsymbol{M}$ as the feature matrix of the data set, because the relation matrix $\boldsymbol{M}$ has the following characteristics : 1) $\boldsymbol{M}$ can convert different types of data in the data set into a unified symbolic representation; 2) $\boldsymbol{M}$ can preserve the interrelationship between elements, but traditional matching methods ignore this; 3) $\boldsymbol{M}$ solves the problem of elements in complex pattern matching The problem of variable arrangement order can better represent the characteristics of the data set. In the relationship matrix $\boldsymbol{M}$, all items in each row are combined to form a relationship key pair. The proposed algorithm traverses the data set to extract the relationship key pair of the data set frequency distribution feature vector, as the data set $\boldsymbol{v}_i$, clustering feature vectors obtained based on matching algorithm as detailed Algorithm 3.

Algorithm 2: Data type feature classification algorithm

Input: IoT heterogeneous data collection $X = \{x_1, x_2, \ldots, x_n\}$ (structured data)

Output: Data type feature classified set $\{E,\ F,\ G\}$

```
1: Obtain the structured data
2: Standardize the data collection of all data sets $x_i$ in $X$
3: $\forall x_i \in X$, traversing $X$ each column, based on distribution data type construct eigenvectors $v_i$, to give corresponding feature matrix $\boldsymbol{V} = (v_1, v_2, \ldots, v_n)$
4: for each $x_i \in X$ do
5:     for each $col \in x_i$ do
6:         Determine the number of occurrences of each data type in the statistical process in step 1
7:     end for
8:     Build data type feature vector $\boldsymbol{v}_i$
9:     $n = Column\_num(x_i)$
10:    $\boldsymbol{v}_i \leftarrow \left| \frac{Count(\text{“value”})}{\sum_{i=1}^{n} i}, \frac{Count(\text{“char”})}{\sum_{i=1}^{n} i}, \frac{Count(\text{“date”})}{\sum_{i=1}^{n} i}, \frac{Count(\text{“char − value”})}{\sum_{i=1}^{n} i}, \frac{Count(\text{“char − date”})}{\sum_{i=1}^{n} i} \right|$
11:    if $\frac{Count(\text{“date”})}{\sum_{i=1}^{n} i} \neq$ or $\frac{Count(\text{“char − date”})}{\sum_{i=1}^{n} i} \neq 0$
12:        $x_i \in E$
13:    else
14:        if $\left| \frac{Count(\text{“value”})}{\sum_{i=1}^{n} i} + \frac{Count(\text{“char − value”})}{\sum_{i=1}^{n} i} \right| > \left| \frac{Count(\text{“char”})}{\sum_{i=1}^{n} i} + \frac{Count(\text{“char − date”})}{\sum_{i=1}^{n} i} \right|$
15:            $x_i \in F$
16:        else $x_i \in G$
17:        end if
18:    end if
19: end for
20: return $\{E,\ F,\ G\}$
```

Algorithm 3: Feature cluster matching algorithm

Input: S_{cla}, T_{cla}

Output: Cluster set of S_{cla} and T_{cla}

```
1: Get the classification set in $S_{\text{cla}}$ and $T_{\text{cla}}$: $S_{\text{cla}}$ or $T_{\text{cla}} = \{\text{Classification}_1, \text{Classification}_2, \text{Classi fication}_3\}$
2: dataset $\in \{E, F, G\}$
3: for each classification $\in S_{\text{cla}}$ do
4:     for each $ds \in dataset$ do
5:         $ds \rightarrow \boldsymbol{M}$
6:         for each $row$ belongs to the set of $\boldsymbol{M}$
7:             Relation $Key \leftarrow \{(-,-), (-,0), (-,+), (0,0),\ (0,+), (+,+)\}$
8:             Count the number of occurrences of each relationship $Key$
9:         end for
10:        $n \leftarrow$ Count the total number of occurrences of all relation $Key$
11:        $\boldsymbol{v}_i \leftarrow \left| \frac{Count(-,-)}{\sum_{i=1}^{n} i}, \frac{Count(-,0)}{\sum_{i=1}^{n} i}, \frac{Count(-,+)}{\sum_{i=1}^{n} i}, \frac{Count(0,0)}{\sum_{i=1}^{n} i}, \frac{Count(0,+)}{\sum_{i=1}^{n} i}, \frac{Count(+,+)}{\sum_{i=1}^{n} i} \right|$
```

(Continued)

Algorithm 3: Continued

```
12: end for
13: end for
14: Obtain the eigenvector matrix $\boldsymbol{V}_s$ of all datasets $S_{cla}$ using steps 2 to 11
15: Obtain the eigenvector matrix $\boldsymbol{V}_T$ of $T_{cla}$ using steps 2 to 11
16: Cluster the eigenvectors belonging to $\boldsymbol{V}_s$ via SOM, calculates its cluster centers (average method), and obtains the cluster center set $C = \{\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_p\}, \{\boldsymbol{c}_{p+1}, \boldsymbol{c}_{p+2}, \ldots, \boldsymbol{c}_q\}, \{\boldsymbol{c}_{q+1}, \boldsymbol{c}_{q+2}, \ldots\}\}$
17: For $\boldsymbol{V}_T$ to any one feature vector $\boldsymbol{v}_i$, which is calculated with the Euclidean distance from the cluster center $C$ similarity
18: for each $\boldsymbol{v}_i \in \boldsymbol{V}_T$ do
19: for each $c_j \in C$ do
20: $sim(\boldsymbol{v}_i, \boldsymbol{c}_j)$
21: end for
22: The max: $sim(\boldsymbol{v}_i, \boldsymbol{c}_j)$ in $\boldsymbol{v}_i$ corresponding to the data set into $\boldsymbol{c}_j$ corresponding cluster
23: end for
24: return cluster set
```

In this section, we cluster all the data sets in S_{cla} based on the feature vector extracted from the relation matrix, and obtain the cluster set $D = \{d_1, d_2, \ldots, d_l\}$, by calculating the Euclidean distance. Determine the most similar cluster set in S_{cla} for each data set in T_{cla} , so as to further reduce the matching space. We assume and assume that each cluster in S corresponds to the only data set in T, that is, the data set is only a 1:n relationship between.

3.2.3 Mixed Element Matching

Based on the obtained set of clusters D. $\forall d_i \in D$, all d_i elements are mixed together. In [32], the algorithm for a single cluster matching elements, the method can quickly and effectively find the clustering center in a short time, the clustering process can be completed in a short time and the process is streamlined, and finally the matching result set $R = \{r_1, r_2, \ldots, r_l\}$. Considering the complexity of the actual situation, it is difficult for us to deploy 1:1 for precise matching elements, so that $\forall r_i \in R$, we require r_i that contains only a standardized data elements and φ source data elements (where φ is artificially set, select herein $\varphi =. 1, 2, 3$), these φ elements of source data are recommended to users as the most similar elements, as shown in Algorithm 4.

Algorithm 4: Mixed element matching algorithm

```
Input: Cluster set $D \leftarrow \{d_1,\ d_2, \ldots, d_l\}$
Output: Matching result set $R \leftarrow \{r_1,\ r_2, \ldots, r_l\}$
Get the cluster set $D$
for each $\forall d_i \in D$ do
    Perform element matching
    Set the value of $\varphi$
    $r_i \leftarrow d_i$
end for
Return $R$
```

4 Experimental Analysis

4.1 Data Selector

In order to prove the feasibility and effectiveness of the proposed algorithm, we selected 30 databases from 13 different manufacturers in the household air-conditioning performance test in the field of industrial Internet of Things appliance product testing as the source data, and used the air-conditioning test database based on the international standard IEEE 1851 as the standardized database (using Oracle), the data details are shown in Tab. 1.

Table 1: Source data

S. No.	DataSource	Number	Data size (GB)
1	Sqlserver2005	15	12
2	Txlfiles	8	10
3	Access	7	9.2

We believe that there are generally two common heterogeneous forms: rank conversion and splitting.

1) Row and column conversion. Contents that are not completely in the same column are listed in the same column. In order to eliminate the impact of different structures on matching, we need to perform logical row conversion on this type of column and record the corresponding mapping relationship at the same time, as shown in the Fig. 2.
2) Split. Generally, the following merges are either string separated by special characters, or Boolean merged directly, so find the merged field for logical splitting, and record the corresponding mapping relationship at the same time, as shown in Fig. 3.

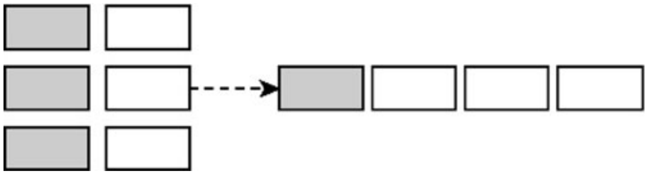


Figure 2: Data conversion flow

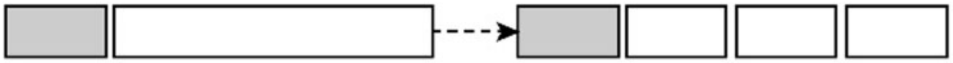


Figure 3: Data splitting

4.2 Algorithms Comparison

We use different pattern matching algorithms based on data instances and database pattern information to compare with the proposed HSMA algorithm. The detailed comparison information is shown in Tab. 2.

1) Shrink Mobile Edge Computing (SMEC). Through naive Bayesian learning, entities are divided into different classes, and the same class is used to match pattern elements between sub-patterns.

2) Sequential matrix diagonalization detection (SMDD). A neural network-based pattern matching algorithm that automatically completes pattern matching by analyzing the distribution law of the data instances contained in the pattern elements.
3) iMAP. A method that comprehensively utilizes multiple types of information in patterns to obtain simple and complex mapping relationships between patterns at the same time. When the matching relationship is obtained, the judgment process of each matching relationship is saved. When the user When the final result is adjusted, the saved judgment process is provided to the user as the basis for the user to make adjustments.

Table 2: Performance comparison of the proposed and existing algorithms

Algorithm	Feature	Classification	Clustering	Machine learning
SMEC	Schema	Yes	No	Naïve bayes
SMDD	Instance	Yes	No	BP-NN
iMAP	Instance	Yes	No	Naïve bayes
Proposed	Instance	Yes	No	SOM and Ref [10]

4.3 Measured Results

In order to evaluate the matching quality of the proposed algorithm, we use three general indicators [33] for evaluation:

1) Precision. The ratio of correct matching results among matching results.

$$\text{Precision} = \frac{T}{P} = \frac{T}{T+F} \tag{3}$$

Among them, T is the matching result that is correctly identified, P is the matching result returned by the matching method, and F is the wrong match [34,35].

2) Recall. The ratio of correct matching results to actual matching results in matching results.

$$\text{Recall} = \frac{T}{R} \tag{4}$$

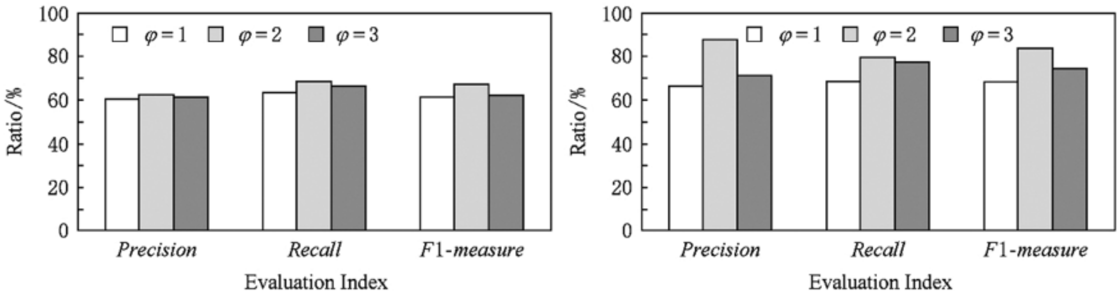
Among them, R is the result of manual matching.

3) F1_measure. Statistics that can comprehensively evaluate the quality of matching.

$$\text{F1}_{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

4.3.1 Self-Test of the Proposed Algorithm

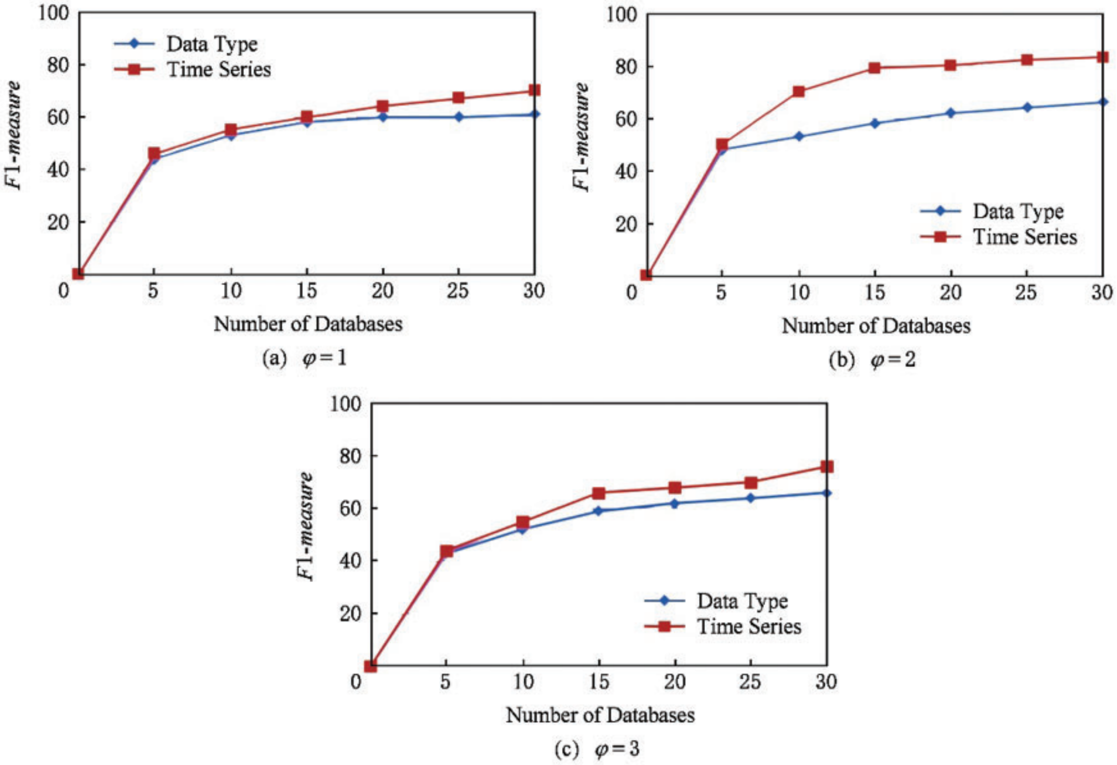
1) The matching quality of the HSMA algorithm is affected by the selection of the parameter φ [36,37]. By setting different values for φ, the precision, recall and F1_measure is analyzed and compared, and the result is shown in Fig. 4. With $\varphi = 1$ in contrast, when $\varphi > 1$, the recall rate, precision rate and comprehensiveness are improved, because the proposed algorithm cannot guarantee accurate 1:1 matching, and the heterogeneity of the pattern leads to multiple similar similarity results. But it is not that the larger the value of φ, the better, the increase of φ will increase the matching interference term and reduce the matching quality. When $\varphi = 3$, it can be seen that the matching quality is greater than $\varphi = 1$ but less than $\varphi = 2$.
2) The matching quality of the proposed algorithm is affected by the number of input databases. The number of input databases will directly affect the effect of relational feature clustering. Sufficient data will make the data features more obvious. As shown in Fig. 5, for different φ values, when the number of input databases is in the range of (0, 15), the matching quality continuously improves with the increase of the number of input databases and the change is more obvious. But when the number of input databases is greater than 15, the matching quality increases slowly.
3) The time efficiency of the proposed algorithm is affected by the selection of the parameter φ. Fig. 6 compares the influence of the number of source data on the time efficiency of the proposed algorithm under different values of φ. As shown in Fig. 6, the larger the φ, the more the greater the matching time, because the increase in the value of φ leads to an increase in the number of matches in the mixed element matching process, which increases the time complexity. When the value of φ is fixed, the matching time of the proposed algorithm using time series feature classification is lower than that of using data type feature classification.
4) The time efficiency of the proposed algorithm is affected by the feature classification. We randomly select 15 data sources as the input of the proposed algorithm, respectively use no classification, time series feature classification and data type feature classification and set different values, analyze and compare the time used for each layer of matching is shown in Fig. 7. The selection of φ only affects the matching time of the mixed element matching. The larger the value of φ, the more time it takes for the mixed element matching process. When the value of φ is fixed, the time is used matching time of the proposed algorithm for sequence classification is relatively small.



(a) Classification based on data type

(b) Classification based on time series

Figure 4: Comparison of matching quality of the proposed algorithm



(a) $\varphi=1$

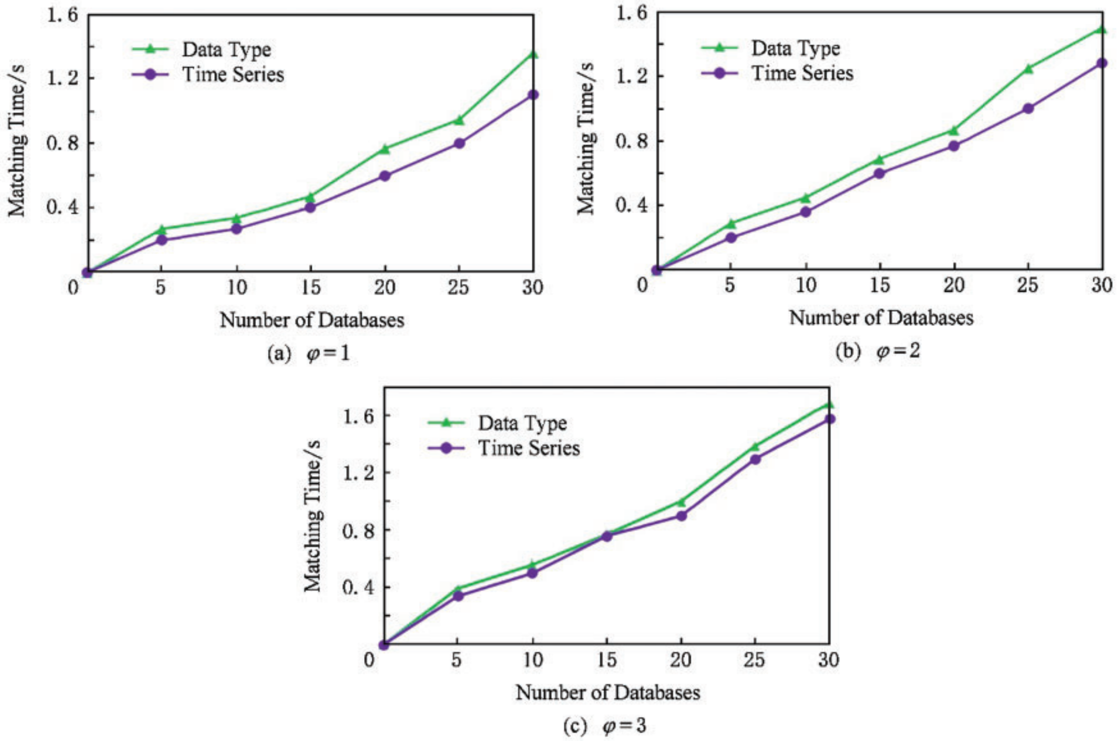
(b) $\varphi=2$

(c) $\varphi=3$

Figure 5: Comparison of data type and time series of the proposed scheme

4.3.2 Performance Comparison with Existing Algorithms

It can be seen from Section 4.3.1 that when $\varphi = 2$ is selected, the proposed algorithm using time series feature classification has the highest performance, so compare it with each pattern matching algorithm in Section 4.2. We have selected 30 heterogeneous databases as the proposed algorithm results of the comparison are shown in Fig. 8. As shown in Fig. 8a, the matching quality of the proposed algorithm is higher than other algorithms. Among them, the heterogeneity of the source database and the incomplete pattern information lead to the most efficient SMEC. As shown in Fig. 8b, because the proposed algorithm uses preprocessing and multiple clustering, the time it takes is significantly higher than other algorithms.

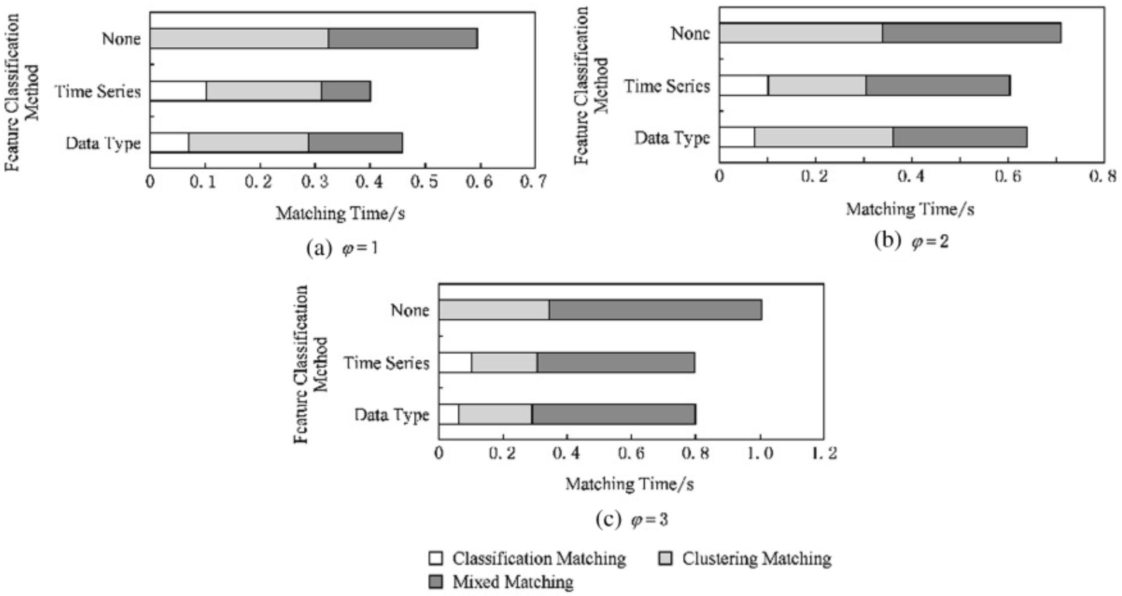


(a) $\varphi = 1$

(b) $\varphi = 2$

(c) $\varphi = 3$

Figure 6: Comparison of matching time of the proposed scheme with increasing number of database

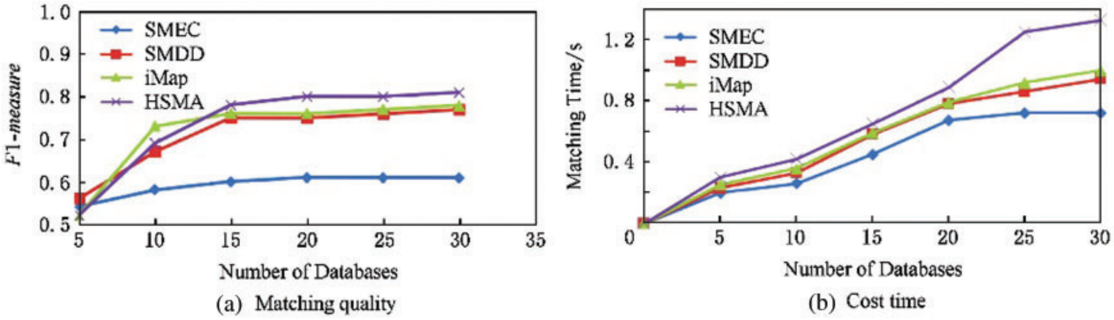


(a) $\varphi = 1$

(b) $\varphi = 2$

(c) $\varphi = 3$

Figure 7: Comparison of matching time of the proposed algorithm



(a) Matching quality (b) Cost time

Figure 8: Evaluation of proposed and existing algorithms *vs.* number of databases

5 Conclusion

We designed a hierarchical pattern matching algorithm. For input of unknown source data, initialize the corresponding field standard set and standardized database according to the field to perform feature classification and matching, and establish the source data set and standard database data set in each category. Then extract the relationship features in the source data set and the target database table, use the SOM clustering algorithm to cluster the data sets in each sub-pattern, and perform similarities between the clustering results and the corresponding standard database data sets the degree to establish a matching mapping. Finally, calculate the similarity of the mixed elements in the matching result set to perform a single element matching. Through the above three levels of matching, the proposed algorithm gradually reduces the matching space and reduces the number of matches, thereby improving the quality and efficiency of matching.

In the process of relation feature cluster matching and mixed element matching, the quality of the clustering algorithm directly determines the final matching quality and overall matching time. Although clustering can reduce the matching space, it may also bring matching errors, resulting in related elements not in the same class. In the future, we will try to use different machine learning algorithms to improve the proposed algorithm, and find the best combination mode to improve the effect of clustering. In addition, this article uses iMAP, SMDD in the experimental verification for comparing the classic algorithms with proposed algorithm has improved the reliability of the comparison results.

Acknowledgement: The authors would like to thanks the editors and reviewers for their review and recommendations.

Funding Statement: The Authors would like to thank as well the Royal Academy of Engineering (UK) for supporting this research under grant Transforming Systems through Partnership Project Number TSP1040.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Doss, J. Paranthaman, S. Gopalakrishnan, A. Duraisamy, S. Pal *et al.*, "Memetic optimization with cryptographic encryption for secure medical data transmission in IoT-based distributed systems," *Computers, Materials & Continua*, vol. 66, no. 2, pp. 1577–1594, 2021.

[2] S. Verma, S. Kaur, D. B. Rawat, C. Xi, L. T. Alex *et al.,* "Intelligent framework using IoT-based WSNs for wildfire detection," *IEEE Access*, vol. 9, pp. 48185–48196, 2021.

[3] A. A. Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[4] M. A. Garadi, A. Mohammed, A. K. Ali, X. Du, I. Ali *et al.,* "A survey of machine and deep learning methods for internet of things (IoT) security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020.

[5] S. Li, L. Li, B. Xu, Y. Feng and H. Zhou, "Research of a reliable constraint algorithm on MIMO signal detection," *International Journal of Embedded Systems*, vol. 12, no. 2, pp. 13–26, 2020.

[6] J. Gojal, E. Monteiro and J. S. Silva, "Security for the internet of things: A survey of existing protocols and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.

[7] K. Tange, M. D. Donno, X. Fafoutis and N. Dragoni, "A systematic survey of industrial internet of things security: Requirements and foq computing opportunities," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2489–2520, 2020.

[8] S. Bashir, M. H. Alsharif, I. Khan, M. A. Albreem, A. Sali *et al.,* "MIMO-terahertz in 6G nano-communications: Channel modeling and analysis," *Computers, Materials & Continua*, vol. 66, no. 1, pp. 263–274, 2020.

[9] F. Jameel, T. Ristaniemi, I. Khan and B. M. Lee, "Simultaneous harvest-and-transmit ambient backscatter communications under Rayleigh fading," *EURASIP Journal on Wireless Communications and Networking*, vol. 19, no. 1, pp. 1–9, 2019.

[10] Q. Alsafasfeh, O. A. Saraereh, A. Ali, L. A. Tarawneh, I. Khan *et al.,* "Efficient power control framework for small-cell heterogeneous networks," *Sensors*, vol. 20, no. 5, pp. 1–14, 2020.

[11] K. M. Awan, M. Nadeem, A. S. Sadiq, A. Alghushami, I. Khan *et al.,* "Smart handoff technique for internet of vehicles communication using dynamic edge-backup node," *Electronics*, vol. 9, no. 3, pp. 1–17, 2020.

[12] W. Shahjehan, S. Bashir, S. L. Mohammed, A. B. Fakhri, A. A. Isaiah *et al.,* "Efficient modulation scheme for intermediate relay-aided IoT networks," *Applied Sciences*, vol. 10, no. 6, pp. 1–12, 2020.

[13] B. M. Lee, M. Patil, P. Hunt and I. Khan, "An easy network onboarding scheme for internet of things network," *IEEE Access*, vol. 7, pp. 8763–8772, 2018.

[14] O. A. Saraereh, A. Alsaraira, I. Khan and B. J. Choi, "A hybrid energy harvesting design for on-body internet-of-things (IoT) networks," *Sensors*, vol. 20, no. 2, pp. 1–14, 2020.

[15] T. Jabeen, Z. Ali, W. U. Khan, F. Jameel, I. Khan *et al.,* "Joint power allocation and link selection for multi-carrier buffer aided relay network," *Electronics*, vol. 8, no. 6, pp. 1–15, 2019.

[16] K. Yiping, J. Hauswald, G. Cao, A. Rovinski, T. Mudge *et al.,* "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," in *Proc. of the ACM 22nd Int. Conf. on Architectural Support for Programming Languages and Operating Systems*, New York, USA, pp. 615–629, 2017.

[17] T. Lawrence and L. Zhang, "IoTNet: An efficient and accurate convolutional neural network for iot devices," *Sensors*, vol. 19, no. 24, pp. 1–17, 2019.

[18] Z. Zhuoran, M. K. Barijough and A. Gertlauer, "Deepthings: Distributed adaptive deep learning inference on resource-constrained iot edge clusters," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2348–2359, 2018.

[19] A. E. Eshratifar, M. S. Abrishami and M. Pedram, "JointDNN: An efficient training and inference engine for intelligent mobile cloud computing services," *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 565–576, 2019.

[20] S. Dey, J. Mondal and A. Mukherjee, "Offload execution of deep learning inference at edge: challenges and insights," in *IEEE Int. Conf. on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Kyoto, Japan, pp. 1–6, 2019.

[21] S. Yao, Y. Zhao, H. Shao, S. Liu, D. Liu *et al.,* "Fastdeepiot: towards understanding and optimizing neural network execution time on mobile and embedded devices," in *Proc. of the ACM Conf. on Embedded Networked Sensor Systems*, New York, USA, pp. 278–291, 2018.

[22] W. Shi, Y. Huo, S. Zhou, Z. Niu, Y. Zhang *et al.,* "Improving device-edge cooperative inference of deep learning via 2-step pruning," in *IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS)*, Paris, France, pp. 1–7, 2019.

[23] C. Hu, W. Bao, D. Wang and F. Liu, "Dynamic adaptive dnn surgery for inference acceleration on the edge," in *IEEE Int. Conf. on Computer Communications (INFOCOM)*, Paris, France, pp. 1–9, 2019.

[24] H. Mao, X. Chen, K. Nixon, C. Krieger and Y. Chen, "MoDNN: Local distributed mobile computing system for deep neural network," in *Design, Automation & Test in European Conf. and Exhibition*, Shanghai, China, pp. 1–6, 2017.

[25] J. Yu, A. Lukefahr, D. Palframan, G. Dasika, R. Das *et al.,* "Scalpel: Customizing dnn pruning to the underlaying hardware parallelism," in *Proc. of the Int. Symp. on Computer Architecture (ISCA)*, Toronto, Canada, pp. 548–560, 2017.

[26] J. Madhavan, P. Bernstein and E. Rahm, "Generic schema matching with cupid," in *Proc. of the 27th VLDB Conf.*, San Francisco, USA, pp. 49–58, 2001.

[27] H. Do, "COMA: A system for flexible combination of schema matching approach," in *Proc. of the 28th VLDB Conf.*, San Francisco, USA, pp. 610–621, 2002.

[28] A. Doan, "Reconciling schemas of disparate data sources: A machine-learning approach," in *Proc. of the 1st ACM SIGMOD Int. Conf. on Management of Data*, New York, USA, pp. 509–520, 2001.

[29] R. Dhamankar, "iMAP: discovering complex semantic matches between database schemas," in *Proc. of the 4th ACM SIGMOD Int. Conf. on Management of Data*, New York, USA, pp. 383–394, 2004.

[30] P. Paakkonen and D. Pakkala, "Reference architecture and classification of technologies, products and services for big data systems," *Big Data Research*, vol. 2, no. 4, pp. 166–186, 2015.

[31] J. Kang and J. Naughton, "Schema matching using interattribute dependencies," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 10, pp. 1393–1407, 2008.

[32] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science Journal*, vol. 334, no. 6191, pp. 10182–10196, 2014.

[33] G. Cyril and G. Eric, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," *International Journal of Radiation Biology*, vol. 51, no. 5, pp. 199–239, 2005.

[34] M. Mustafa and A. Al-Badi, "Role of internet of things (IoT) increasing quality implementation in Oman hospitals during covid-19," *SPAST Abstracts Journal*, vol. 1, no. 1, pp. 878–886, 2021.

[35] T. Kassanuk, M. Mustafa, P. Panse, R. Sivanand, K. Phasiam *et al.,* "An internet of things and cloud based smart irrigation system," *Annals of R.S.C.B*, vol. 25, no. 4, pp. 20010–20016, 2021.

[36] M. Jawarneh and S. Alzubi, "Factors affecting the success of internet of things for enhancing quality and efficiency implementation in hospitals sector in Jordan during the crisis of covid-19," *Internet of Medical Things*, vol. 5, pp. 107–140, 2020.

[37] M. Jawarneh, V. Madhava, R. Selvaraj, V. Rao, S. Kumar *et al.,* "Towards security and privacy concerns in the internet of things in the agriculture," *Journal of Physiotherapy and Rehabilitation*, vol. 32, no. 3, pp. 1063–1071, 2021.