Section: Research Paper

Hybrid Meta-Heuristic Based Clustering Model for Colossal Pattern Mining from High-Dimensional Patterns

T. Sreenivasula Reddy [1*] **R. Sathya** [2]

[1*] Research Scholar, Department of Computer Science and Engineering, Faculty of Engineering and Technology, Annamalai University, Annamalai Nagar, Chidambaram, Tamil Nadu-608002, India.
[1*]seenu4linux@gmail.com

[2]Assistant Professor, Department of Computer Science and Engineering, Faculty of Engineering and Technology, Annamalai University, Annamalai Nagar, Chidambaram, Tamil Nadu-608002, India
[2]sathya_vai@yahoo.com

Abstract

High-dimensional datasets were made possible by the growing popularity of bioinformatics as a topic of study and the abundance of data in many other fields. High-dimensional datasets have a small number of rows and a large sum of characteristics. Datasets with high dimensionality make it difficult to extract useful information from them. In this research, we introduce a hybrid meta-heuristic-based clustering tactic for efficiently discovering massive common patterns in high dimensional datasets. In this research, we introduce the Harris Hawks Optimizer with Arithmetic Optimization Procedure, a novel hybrid metaheuristic method (HHO-AOA). The idea is to use it to group the massive pattern. The HHO and AOA are used in a coordinated fashion to create the suggested hybrid algorithm. The devised approach is supposed to improve solution accuracy during optimisation by expanding the pool of possible solutions. The assessment criteria and standard statistical tests are used to confirm the results. Using a hybrid model shortens the time it takes to look for results in a database. At last, a massive cluster is built, and from it, massive patterns emerge. We conduct the tests on many high-dimensional datasets and utilise a wide range of efficiency measures. The results of the studies demonstrate that the suggested approach yields notable and efficient mining outcomes.

Keywords: Arithmetic Optimization Algorithm; Harris Hawks Optimizer; Information extraction; High dimensionality; Colossal Pattern.

Introduction

One of the cornerstones of data mining is the identification of recurrent patterns, or "frequent patterns," that exist in a given dataset, in this case a series of financial transactions [1]. Applications reaching from market basket analysis and reference systems to spam detection have inspired several iterations of the issue to investigate various patterns [2] to sequential patterns [3] to subgroups [4] to graphlets [5]). Some real-world requests involve

investigating a pattern within the context of a series of datasets, where the sequence is provided by, for instance, the collecting of data at varying times. In market basket investigation, for instance, it makes sense to look at the similarities and differences (item sets) across datasets collected from sales that occurred on various days of the week or month. Statistically sound pattern mining [6] assumes that in nearly all cases, each dataset is created through a reproductive process on transactions, which creates transactions according to some probability distribution. Let's say we're collecting data on customer takes at a supermarket through a series of n surveys conducted at n various time intervals. As it is plainly impossible to gather the receipts of the whole population, the purpose of such surveys is to infer info on how the behaviour of the entire consumer community changes. As a result, all that our datasets really are is a smattering of the entire population.

Zhu et al. (2007) [7] introduced the first technique for mining very large sets of objects. The Pattern Fusion technique approximates the huge dataset in order to mine it. Due to the approximation approach used, the Pattern Fusion procedure will not be talented to mine all of the enormous item sets. The decision-making process is hampered by the inability to develop all possible association rules. The BVBUC algorithm will not mine any extremely large or extremely closed itemsets [8]. The decision-making process is hampered by the inability to develop all possible association rules. The majority of BVBUC-mined, closed-items-set support data is incorrect. Because of this, flawed association rules are produced, which in turn impacts judgement. In order to mine FCCI [10-12] from a dataset, the reference [9] presented a DisClose technique that makes use of Compact Row trees (CR-trees). While mining FCCI from a high-dimensional dataset is possible, state-of-the-art techniques do not first remove irrelevant rows and features. This makes the technique inefficient since the search space for row enumerated mining grows exponentially with the number of rows. Neither an effective trimming strategy nor a closure mechanism for testing the closeness of the rowset are included in the current FCCI mining methods.

The authors also suggested the bottom-up, transaction-based BVBUC [13] algorithm. When the number of transactions in a set approaches minSup (which indicates that the designs in the set of dealings are common), the BVBUC links together the 1-transactions to create 2-transactions, and so on [14, 15]. To further condense the search area, the authors also suggested a method to remove twigs that cannot grow to minSup.

The following are some of the drawbacks of BVBUC, despite the fact that it is quicker than CMP and Pattern-Fusion:

1. BVBUC is inefficient since it must repeatedly calculate the patterns of a series of transactions.

2.BVBUC use the descending closure stuff to remove data that does not meet minSup, but it does not delete transactions that do not meet the same criteria (after eliminating rare items, some dealings do not comprise slightly items).

3. Due to the high volume of duplicates produced by BVBUC, verification processes take longer.

4.BVBUC uses a dataset of transactions (tidset) to identify patterns by computing their intersections. If two datasets are related as parents and children, then the only difference between them is a single transaction. Tidsets 1 2 3 and 1 2 3 4 are the same except for the fourth transaction. By calculating the intersection between the pattern X of tidset [1, 2], and the pattern 4 of transaction [1, 2], we may obtain the pattern of tidset [1, 2], 3, 4 in this example.

5. The use of bit vectors increases the required capacity and the time required to join high dimensional databases.

In this research, we apply a data-conversation model and a metaheuristic approach to clustering extensively large patterns. Listed below are the most important results of this study.:

- This is the first known use of a combined HHO algorithm with AOA.
- The Arithmetic Optimization Process (HHO-AOA) is a novel hybrid metaheuristic method introduced to the literature.
- An original scenario is created and given for optimal utilisation of the suggested hybrid HHO-AOA.

Part 2 introduces many models already in use for extracting information from the massive pattern. Part 3 details the suggested approach for mining datasets, whereas Section 4 presents the results of an experimental research. In Section 5 we wrap up the study.

2. Related works

According to Vanahalli and Patil [16], the (FCCI) have a significant amount of weight in a variety of fields, one of which being bioinformatics. The FCCI is responsible for a substantial portion of the decision-making process. When working with datasets that have a high dimensionality, it might be challenging to extract information that is helpful. The most cutting-edge algorithms available today do not completely get rid of useless rows and attributes. The work that is being suggested details an effective pruning strategy for reducing the search space for row enumeration mining as well as a closure method for assessing whether or not a particular rowset is sufficiently close. Both of these methods are specified in the work. An efficient row enumeration strategy that includes the rowset technique and the trimming approach has been created so that mining the whole FCCI set may be done in an effective and efficient manner. Experiments have proven that it is beneficial to get rid of all of the unnecessary data and columns in a database.

In a setting with a sequence of datasets generated by potentially different fundamental generative processes, Tonon and Vandin [17] travel the problem of mining statistically robust patterns, which are patterns whose likelihoods of appearing in transactions drawn from such generative processes respect well-defined circumstances. There are well-defined requirements that statistically robust patterns must meet in order to exist in transactions taken from such generating processes. These limits specify the desired patterns and describe the sequence of datasets in which those patterns are most likely to occur. It's possible that these

probability will go up, down, or stay the same. Because data is stochastic, analysing a sequence of samples (the datasets) created using generative processes can only provide approximations rather than the actual set of statistically robust patterns. This is because approximations are more easily approximated. The authors next describe gRosSo, an algorithm for obtaining accurate approximations of statistically reliable patterns that are devoid of highly implausible false positives and false negatives. gRosSo was developed in order to find correct approximations of patterns. It is a framework that may be used to extract trustworthy sequential patterns and item sets from data. The extensive evaluation performed on both synthetic and actual data demonstrates that gRosSo offers approximations of a high itemsets.

Singh and Jindal [18] describe an original approach for dynamically recognising fraudulent transactions based on historical information. Their method is touted as an innovation. Hence, we propose sequential pattern mining as a method for detecting database intrusions based on an analysis of user behaviour according to several trust variables (TFUBID). In order to model trust factor–based behavioural patterns of employees who access the database, we must first cluster user behaviour vectors using fuzzy clustering. After that, we must define a class of Integral Data Attributes mining, giving more weight to critical elements and Directly Correlated Attributes. Finally, we can begin modelling trust factor–based behavioural patterns of employees who access the database. Extensive experimental evaluations conducted on the synthetic dataset in accordance with the TPC-C standard benchmark demonstrated that TFUBID outperformed competing state-of-the-art algorithms on a variety of performance criteria, and it achieved an accuracy of 94% when identifying fraudulent transactions.

The procedure that was proposed by Barrientos et al. [19] takes as its input a collection of values (or measurements), and it returns the K values that are lowest within that collection as its output. Measures extracted from metric and spaces, in addition to high-dimensional databases, are all compatible with the method that has been described. In this work, a brand new method that is exhaustive and based on GPUs is presented for addressing kNN issues. The algorithm is broken down into two distinct steps. The first utilises pivots in order to condense the search space, and the second makes use of a collection of heaps in order to deliver the final output. Both of these structures are considered intermediate structures. The method has been improved such that it is now compatible with multi-GPU as well as multi-node/multi-GPU setups. This study is the most recent and cutting-edge example of its kind found in the technical literature. It does this by using the database with the most data volume to carry out a kNN query on up to elements, and it achieves a speed-up of up to 1843 when utilising a platform with 5 nodes and 20 GPUs. The study also uses the largest database in terms of data volume.

Liu et al. [20] propose a fresh strategy for knowledge detection that is centred on double-evolving frequent pattern trees as a means of keeping up with data that is in a state of constant flux. The first tree is used to maintain track of recurring themes in the data that has been gathered, and the second tree is used to keep track of fresh instances of those themes. Sliding windows are used to do regular revisions on both the double frequent pattern trees

and the connections that exist between them whenever new information comes to light and is made available. The incremental data is mined for new frequent patterns, and insights may be derived from the alterations that are produced as a result of this process. The findings of the assessments indicate that the algorithm is efficient at finding new information hidden inside dynamic datasets.

Rambabu and Govardhan [21] provide a novel Data Replication system. To do so, they make use of data mining techniques. In this scenario, the process of replicating the data is carried out by locating data patterns that are utilised often within the vast database of a node. In order to do this, we will implement an approach for mining common patterns that is helped by optimisation, and a novel hybrid algorithm will be responsible for determining the optimal threshold value. The hybrid technique that is offered is called Greywolves Updating Exploration and Exploitation with Sealion Behaviour. It incorporates concepts from both the Sealion Optimization Model (SLnO) and the (GWO) algorithms (GUEES). In addition, the mining work shall be carried out in a manner that is that have been set about the priority and the cost. Data with high and low priorities can be placed in a queue; the cost is decided by an estimate of the amount of data storage required. The GUEES approach enhances the effectiveness of the queues that are the most significant. A comparative validation is performed in order to ascertain, in the end, whether or not the model that was chosen is effective. The network utilisation of the suggested model is 35.07 percentage points higher than that of SMO, LA, ROA, GWO, SLnO, PSO, and HCS when the number of requests is set to 1000, and it is 34.9 percentage points higher than that of ROA, GWO, SLnO, PSO, and HCS when the number of requests is set to 1000.

ITUFP is an acronym that stands for Davashi's rapid method, which he offers in [22] for the interactive mining of Top-K UFPs. The projected method takes use of an innovative data structure known as an IMCUP-List in order to store information about patterns in a manner that is both effective and efficient. It constructs UP-Lists from a single database scan after first generating IMCUP-Lists to extract patterns, saving every list it creates, and making a copy of every list it saves. The approach that is being proposed does not produce new IMCUP-Lists in response to a change in K; rather, it only updates the ones that are already present. The UP-Lists and IMCUP-Lists are only made once, and then they are reused in mining with variable K values. As a result, ITUFP is compatible with the notion of "build once, mine many," as these lists are only utilised once. This is the first investigation into the interactive mining of Top-K UFPs, which is now taking place. The proposed method is very useful for interactive mining of Top-K UFPs, as demonstrated by comprehensive experimental consequences with sparse and dense uncertain data. This is particularly the case.

3. Proposed system

3.1. Database preparation

We present an expressed data matrix (EDM) that has m rows and n columns. The purpose of this matrix is to construct experimental conditions on rows and attributes on columns for a defined set of characteristics in the Transactional dataset, which is displayed in Table 1. Table 2 is an example of an EDM bit matrix, which is the TD equivalent. In this matrix, the

value binary 1 denotes "over-expressed," and the value binary 0 denotes "under-expressed." There is a link between the information that is over-expressed and a transaction in TD.

Table 1: EDM of the sample database (TD)

Tid	A	B	C	D	E	F	G	H	I	J	K
1	1	1	1	0	1	0	1	1	1	0	0
2	1	0	1	1	1	1	0	1	0	1	0
3	0	1	0	0	1	1	1	1	0	0	0
4	1	1	1	1	1	1	1	0	0	0	1
5	1	1	0	1	0	1	1	0	0	0	0
6	0	1	0	0	1	0	1	1	1	1	1
Attribute support	4	5	3	3	5	4	5	4	2	2	2

Table 2: EDM with minimal support 3 after pruning.

Tid	A	B	C	D	E	F	G	H
1	1	1	1	0	1	0	1	1
2	1	0	1	1	1	1	0	1
3	0	1	0	0	1	1	1	1
4	1	1	1	1	1	1	1	0
5	1	1	0	1	0	1	1	0
6	0	1	0	0	1	0	1	1
Attribute support	4	5	3	3	5	4	5	4

Pruning will be performed on characteristics whose cumulative frequency is lower than the minimum support on EDM. Table 1 displays the speech information matrix with a minimum support of 3 after the trimming algorithm has been applied.

3.2 Data Cleaning

The reliability of the transactional data being mined is crucial to the precision of frequent item mining outcomes. Unfortunately, it is difficult to be confident that the mining result is legitimate because the transactional data acquired in this method is typically inadequate, noisy, and inconsistent. The suggested design includes a step for cleaning the data using the K-mean cluster technique to address the aforementioned issues. It completes missing data, quiets down noisy data, fixes up inaccurate data, gets rid of redundant or unnecessary information, and turns inconsistent information into a uniform format. Algorithm 1 is the fundamental data cleaning algorithm, and its purpose is to cleanse data sets by excluding whitespace, special characters, and irrelevant symbols.

Procedure 1 : Primary Data Cleaning

$$Input: Data\ Set\ of\ [item1, item2, \ldots\ldots item n]$$
$$Output: Cleaned\ Data\ Sets\ Values$$
$$count = 0$$
$$reader = args[0]$$
$$writer = args[1]$$
$$new\_line = null$$
$$while\ new\_line = reader.readLine() != NULL\ do$$
$$tokens[] = new\_line.split[,]$$
$$count++$$
$$if\ count > 0\ then$$
$$if\ isAlpha(tokens[6])\ then$$
$$writer.write(count.tokens[7].tokens[8])$$
$$else$$
$$writer.write(count.tokens[6].tokens[7])$$
$$end$$
$$end$$
$$end$$
$$isAlpha(Stringname)$$
$$Chars[] = name.toCharArray$$
$$specialCharacters = "|| - ||space||(||)$$
$$flag = false$$
$$while\ c \in chars\ do$$
$$if\ (!Charactre.isLetter(c))\ then$$
$$if\ !(c == specialCharacters)\ then$$
$$flag = false$$
$$return\ flag$$
$$end$$

3.3. Data Conversion

During the transformation of data, the considered objects' names are transformed into identifying numbers. This is a diagram of the data conversion algorithm.

Algorithm 2: Data Conversion Algorithm

$$Input: List\ of\ items$$
$$Output: Item\ names\ are\ converted\ to\ Unique\_ID$$
$$Data\_Conversion()$$
$$tdata = read\_file("input.txt")$$
$$n = total\ number\ of\ items$$
$$foreach\ items\ do$$
$$assign\ unique\_id$$
$$end$$

$write\ item.name\ and\ unique\_id\ into\ file\_1$
$context.write(key, centroid)$
$return\ context$
$foreach\ items\ tdata\ do$
$convert\ item\ name\ to\ unique\_id$
$refer\ file\_1\ to\ return\ each\ tdata\ item\ with\ its$
$unique\ id$
end
$write\ converted\ values\ into\ file\_2$

3.4. Clustering the Colossal Pattern using Hybrid Optimization Model

The massive pattern is clustered using the one-of-a-kind ID provided as an input from Algorithm 2. The hybrid algorithm is based on a novel cooperation between the HHO and AOA algorithms. In the beginning of the development procedure, we break down each procedure and describe their particular structure and how they go about searching for solutions.

3.4.1. Harris hawk's optimization algorithm

1) In 2019, Ali Asghar Heidari et al. created the HHO algorithm. The behaviour of Harris' hawks served as inspiration for this population-based metaheuristic algorithm. The cooperative hunting strategy of ' hawks provided the inspiration for the HHO algorithm. The next sections provide an explanation of these processes and their mathematical modelling according to three distinct phases: phase I (exploration), phase II (transitioning from exploration to exploitation), and phase III (exploitation) (phase II).**PHASE I – EXPLORATION**

Harris' hawks are the potential solutions considered during the hunting phase of the HHO algorithm. Harris' hawks have exceptional vision that allows them to swiftly and accurately find their prey. It may take a long time in a desert to wait for, observe, and track a prey. Harris' hawks represent the process of perching at a random position and finding their prey during a hunting event individually depending on the status of q, as shown in equation (1), which is either q 0:5 or q 0:5. Harris' hawks are assumed to sit close to their family and their prey (a rabbit) if q 0:5, and to perch on random tall trees within their range if q > 0. The next iterated hawk locations are given by X(t+1) in equation (1). The average position is utilised for the position update inside the search range to mimic the hawks' activity as closely as possible. At iteration t, the average hawk location is given by the equation (2).

$$X(t+1) = \begin{cases} \left(X_{rabbit}(t) - X_m(t)\right) - r3_{hho}\left(LB + r4_{hho}(UB - LB)\right) & q < 0.5 \\ X_{rand}(t) - r1_{hho}.\left|X_{rand}(t) - 2r2_{hho}X(t)\right| & q \geq 0.5 \end{cases} \quad (1)$$

$$X_m(t) = \frac{1}{N_{hho}}.\sum_{i=1}^{Nhho} X_i(t) \quad (2)$$

2) TRANSITION FROM PHASE I TO PHASE II

The Harris' hawks' shift from the discovery phase to the misuse phase is tied to the vigour of the prey they pursue. Equation models the time-dependent decline in the kinetic energy of

prey escaping from Harris' hawks (3). Prey energy, denoted as E0, begins each cycle with a value between -1 and 1 determined at random. The prey's (the rabbit's) strength improves as E0 rises from 0 to 1, but its weakness worsens as E0 falls from 0 to -1. The iterative evaluation of the escape energy E reveals a diminishing trend.

$$E = 2E_0\left(1 - \frac{t}{T}\right) \text{ (3)}$$

3) PHASE II - EXPLOITATION

Once Harris' hawks locate their prey during the investigation phase, the exploitation phase begins with a surprise assault (surprise pounce). Surprise pounce or seven kills techniques by the hawks are accomplished in a variety of ways, much as in the real world where the hunter hunts the prey and the animal flees. The parameter r is linked to the prey's escape probability, as the animal is always attempting to outrun the hunter. If r is less than zero, the prey was caught off guard and killed, whereas a value of zero indicates that the prey evaded capture. Harris' hawks use four distinct tactics during the exploitation stage: a soft besiege, a hard besiege, a soft besiege combined with progressive fast dives, and a hard besiege combined with progressive rapid dives soft besiege where r = 0.5 and e = 0.5. Each tactic is described in detail below.

We say the rabbit has enough energy to get away from the hawks when r is greater than 0.5 and |E| is greater than 0.5. Yet, the Harris' hawks' gentle besiege methods render the rabbit's ostensibly deceptive movements and jumps futile. The hawks swoop down on him quickly. Equations (4)–(6) yield the current position of the hawks, the difference between the rabbit's position and the current location, and the random leap strength of the rabbit..

$$X(t+1) = \Delta X(t) - E|(J.X_{rabbit}(t) - X(t))| \text{ (4)}$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \text{ (5)}$$

$$J = 2(1 - r_{5_{hho}}) \text{ (6)}$$

$b: r \geq 0{:}5\ AND\ |E| < 0{:}5 - HARD\ BESIEGE$

If r is less than 0:5 and |E| is less than 0:5, the rabbit is assumed to have insufficient energy to outrun the Harris' hawks. The hawks had surrounded the rabbit with a strong besiege technique and ambushed it. In a severe siege scenario, the current hawks' location is determined by the equation (7).

$$X(t+1) = X_{rabbit}(t) - E.|\Delta X(t)| \text{ (7)}$$

c: r < 0:5 AND $|E| \geq 0{:}5$-SOFT BESIEGE WITH PROGRESSIVE RAPID DIVES

The rabbit has enough energy to get away from the Harris' hawks if r > 0:5 and |E| > 0. Levy flight now represents the rabbits' method of evasion, which consists of a series of zigzags and other deceptive manoeuvres (LF). Hawks engage in a "soft besiege" prior to launching a surprise attack, and they evaluate and decide on their next move using an equation (8). If they have made poor diving judgements in the past, they may attack their prey with sudden and

erratic dives. The hawks trained by Harris will continue diving until they catch the rabbit. For this, we use the LF pattern from equation (9). Formulation of the LF function (10). In this scenario, the hawks' positions are updated using the equation (11).

$$Y = X_{rabbit}(t) - E.|J.X_{rabbit}(t) - X(t)| \quad (8)$$

$$Z = Y + S.LF(D) \quad (9)$$

$$LF(x) = (0.01)\left(\frac{u\sigma}{|v|^{\frac{1}{\beta}}}\right),$$

$$\sigma = \left(\frac{\Gamma(1+\beta)sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)\beta 2^{\left(\frac{\beta-1}{2}\right)}}\right)^{\frac{1}{\beta}} \quad (10)$$

$$X(t+1) = \begin{cases} Y & if\ F(Y) < F(X(t)) \\ Z & if\ F(Z) < F(X(t)) \end{cases} \quad (11)$$

d: r < 0:5 AND $|E|$ < 0:5-HARD SURROUND WITH LIBERAL RAPID DIVES

The rabbit is helpless against the Harris' hawks if r is less than 0:5 and |E| is less than 0. Hawks, like cats, undertake hard besiege before launching a surprise attack on their victim, which is called a "soft besiege." This can only occur if the regular position of the hawks is closer to the target. Equations (12)-(14) provide the laws that apply in this scenario (14).

$$X(t+1) = \begin{cases} Y & ifF(Y) < F(X(t)) \\ Z & if\ F(Z) < F(X(t)) \end{cases} \quad (12)$$

$$Y = X_{rabbit}(t) - E.|J.X_{rabbit}(t) - X_m(t)| \quad (13)$$

$$Z = Y + S.LF(D) \quad (14)$$

3.4.2. Arithmetic optimization algorithm

In 2021, Laith Abualigah et al. introduced the AOA algorithm. Mathematical arithmetic operations (add, subtract, multiply, and divide) served as inspiration for this population-based metaheuristic method. The AOA method divides the application of mathematical operators to a problem into an initiation phase, an exploration phase, and an exploitation phase. In the sections that follow, we'll break down each stage of the AOA.

1) INITIALIZATION

Using the equation as a starting point, the AOA method optimises the set of potential solutions X. (15). Starting with a pool of randomly generated candidates, the optimisation process keeps on until either the optimal solution is found or the stopping requirement is met. Then, a maths optimizer accelerated function (MOA) is used to make decisions about the exploration and exploitation phases using the rule established by equation (17), which expresses a coefficient (17). The r1aoa shown include only integers between 0 and 1. The system operates in the exploratory mode when r1 AOA is less than MOA, and in the exploitation mode when r1 aoa is more than MOA.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & x_{2,j} & \cdots & x_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & \cdots & \cdots & x_{N-1,n} \\ x_{N,1} & \cdots & \cdots & x_{N,n-1} & x_{N,n} \end{bmatrix} \quad (15)$$

$$MOA(t) = MOA_{min} + t.\left(\frac{MOA_{max} - MOA_{min}}{T}\right) \quad (16)$$

$$PhaseSelection = \begin{Bmatrix} Exploration & r1_{aoa} \geq MOA \\ Exploration & r1_{aoa} < MOA \end{Bmatrix} \quad (17)$$

2) PHASE I - EXPLORATION

Multiplication and division, both mathematical operations, have a significant dispersion in the numbers or conclusions they produce. As a result, they will have a hard time achieving convergence. In the discovery phase, we utilise the multiplication and division operators to randomly probe various locations in search of optimal solutions. As stated at the bottom of the next page, the position update equations given in equation (18) are utilised if r2aoa>0:5, else the multiplication operator is employed. Equation (19), displayed at the bottom of the next page, represents the maths optimizer probability function (MOP), a coefficient.

$$x_{i,j}(t+1) = \begin{Bmatrix} best\left(x_j\right) \div \left[(MOP + \in).\left(UB_j - LB_j\right)\mu + LB_j\right] & r2_{aoa} > 0.5 \quad Division \\ best\left(x_j\right).MOP.\left(UB_j - LB_j\right)\mu + LB_j & r2_{aoa} \leq 0.5 \quad Multiplication \end{Bmatrix} \quad (18)$$

$$MOP(t) = 1 - \frac{t^{1/a_{aoa}}}{T^{1/a_{aoa}}} \quad (19)$$

$$X_{i,j}(t+1) = \begin{Bmatrix} best\left(x_j\right) - MOP.\left(\left(UB_j - LB_j\right)\mu + LB_j\right) & r_{3aoa} > 0.5 \ \ subtraction \\ best\left(x_j\right).MOP.\left(\left(UB_j - LB_j\right)\mu + LB_j\right) & r_{3aoa} \leq 0.5 \quad\quad Addition \end{Bmatrix} \quad (20)$$

3) PHASE II - EXPLOITATION

Due to their poor dispersion qualities, addition and subtraction are particularly prone to producing highly dense values or judgements. This means they can readily converge to the goal, unlike the multiplication and division operations. So, during the exploration phase, the multiplication and division operators are used to randomly investigate various locations, while the addition and subtraction operators are employed during the exploitation phase to conduct in-depth investigations. Equation (20), listed at page's bottom, is employed throughout the exploitation phase to calculate position updates. In the first example, the addition operator is used, and in the second case, the subtraction operator is used when r3aoa > 0:5.

3.4.3. Development and design of proposed hybrid algorithm

Thus, we have created a novel hybrid algorithm within the scope of this study by fusing the HHO and the AOA algorithms with a novel approach that accomplishes functioning collaboratively in an optimal method. We hope to get higher solution accuracy by employing the hybrid HHO-AOA method that we have presented. Our goal is to show that by combining several algorithms, we can achieve better results than with just one of them. Each stage is described in depth below.

- Initial random populations are generated when the HHO algorithm's parameters (the total number of populations and iterations) are set.
- In the meanwhile, the AOA algorithm's parameters (the total number of solutions and iterations) are set, and the starting positions of the random solutions are determined.
- For this many loops, the HHO algorithm is executed. In each cycle, the starting vitality, jumping power, and escape vitality are revised. Exploration and exploitation are analysed based on the principles that form the backbone of the HHO algorithm. After calculating the hawks' fitness levels, the optimal position is identified.
- When the number of possible solutions decreases, a flattening pattern emerges in the convergence graph. By starting with a large pool of potential solutions, a substantial proportion of them will eventually prove to be optimum. Yet, as the solution process nears completion, variety and the likelihood of arriving at the best possible solution both decline. The primary focus here should be on including sufficient variation and diversity in the optimisation procedure. So, this research creates and employs a novel hybridization technique to guarantee the best and broadest possible variety. The HHO algorithm's fitness values have been favoured and employed to achieve this goal. Thus, we switch from HHO to AOA when the convergence graph approaches a threshold number of recurring fitness values. AOA is executed for 3, 50, and 200 iterations for HHO fitness levels of 20, 30, and 40, respectively.
- The AOA should be able to pick up where the HHO left off in its quest for answers. It is recommended that the best solution value of the AOA algorithm be set to the HHO's most recent best solution (location) throughout this transition. So, the AOA algorithm will pick up where the HHO left off in its search for a new solution.
- The AOA algorithm is executed for the given number of cycles. Each iteration brings new versions of the MOA and MOP. The AOA algorithm's criteria for exploring and exploiting a resource are the basis for this research. The optimal solution is then determined by comparing the solutions' fitness values.
- The best solution value is then transferred to the best location value of the HHO algorithm, and the solution search process is handed off to the HHO algorithm after the number of iterations set for the AOA method has been achieved. Therefore, the HHO and AOA algorithms cooperate throughout the longest possible HHO iteration.

3.5. Extracting colossal patterns

In order to find the massive patterns, we can potentially expand each core pattern cluster acquired by the hybrid method into an unrooted tree. Un-rooted trees are formed from the top down, listing all core patterns in a cluster that share attribute A. This is done for each core pattern cluster. In a vertical top-down search tree, the size of the core pattern of a node A is never smaller than the size of any of the nodes that make up A's children. For each level of the tree, the values of the support count are listed from highest to lowest, starting with level 1 for the ABCD core pattern. Figure 1 depicts the tree in its unrooted state (a). Similarly, the unrooted tree for the ACD core pattern is seen in Figure 1. (b). Figure 1 shows the unrooted

tree for the ultimate result of merging the core patterns ABCD and ACD into a single cluster (c)

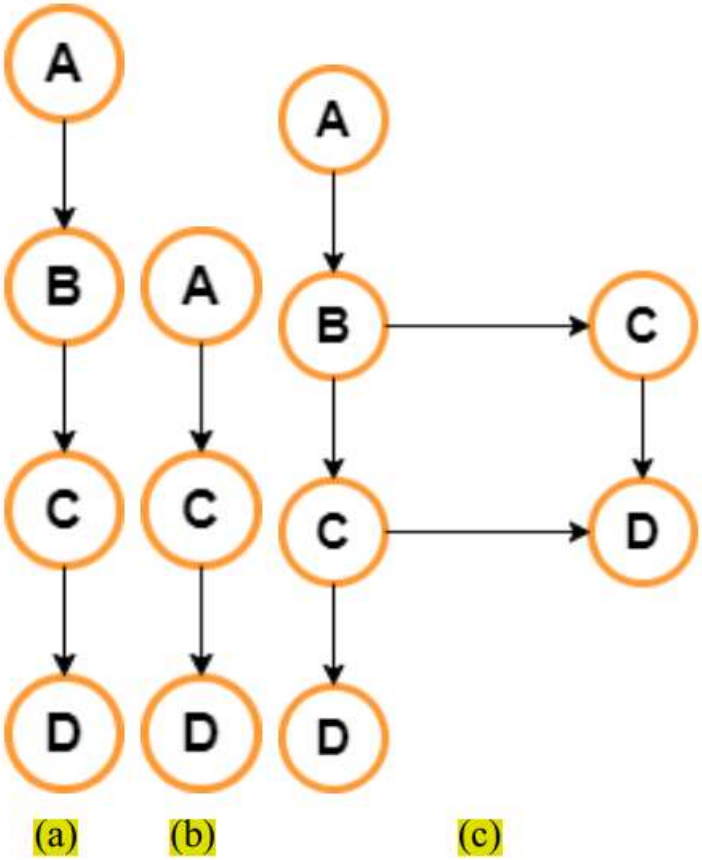

Figure 1: (a) Un-rooted tree for ABCD (b) Un-rooted tree for ACD (c) Merged un-rooted tree

Table 6 Colossal pattern discovered from core patterns for the EDM shown in Table 3 uing un-rooted tree

Colossal pattern sequence
{B, E, G, H}
{A, C, E, H}
{A, B, C, E, G}
{A, B, D, F, G}
{A, C, D, E, F}
{B, E, F, G}

In addition, a vertical top-down tree is used to generate the size of a node's matching pattern. Hence, the size of patterns in every fork of this tree is larger than the size of patterns generated at the fork's level i. Due to the presence of minsup gigantic patterns on the initial tree level. The tree can learn all the massive patterns if we only make it as big as minsup. The unrooted tree only goes through its offspring and removes those that aren't part of a minsup tree, which is specifically conditioned on a non-overlapping set. A subset tree consisting simply of the letter A is shown in Figure 1(c). Table 3 displays the massive patterns that were derived from the unrooted sub-pattern trees. In Figure 1, we see a subset tree where the single

node is the letter A. Table 3 displays the massive patterns that were derived from the unrooted sub-pattern trees.

4. Results and Discussion

A close approximation of the whole mining result may be obtained using the massive pattern mining approach we propose. This is helpful when dealing with massive mining results. There must be a new way to measure how well a collection of massive patterns approximates the full mining outcome. The testing environment is a Windows XP machine with a 2.5 GHz Intel Core i5 and 4 GB of RAM. The algorithm's performance is measured in two ways: (1) its runtime, and (2) a representative score.

The total runtime includes both processing time and input/output time. As database sizes might vary widely, relative rather than absolute values for minsup are used in the performance study. Our algorithms are tested on four authentic databases collected from FIMI (Bayardo, 1998). As can be seen in Table 4, several research have been undertaken utilising a dense transaction data set consisting of four data points.

Table 4: Databases used

Database	**Items**	**Transactions**
Accident	468	340,183
Pumbs*	7,117	49,046
Retail	16,469	88,162
Yeast	79	2,467

In terms of response time in seconds, the output results are reported. Evaluated mining findings are analysed in each test and contrasted with those of current algorithms. To analyse the response time of a range of small regular 1-itemsets, the threshold for minimum help is set for each set of data evaluated. The y-axis reflects the response time on four databases in each figure with different minimum support levels on x-axis.

4.1. Evaluation metrics

Performance metrics such as accuracies, precisions, recalls, F1-scores, and specificities were determined. TP, FN, FP, and TP are abbreviations for "True Positive," "False Negative," "False Positive," and "True Negative," respectively (TN). Here, TP and TN stand for the right number of margin-positive and margin-negative photos, whereas FP and FN stand for the number of margin-negative images that were improperly accepted as margin-positive and the number of margin-positive images that were misclassified as margin-negative.

Calculated using equation (21), accuracy scores reveal how often the models correctly predicted outcomes.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (21)$$

Precision: It only indicates "how many of the chosen data pieces are important." To rephrase, precision measures how accurate the method is by counting how many "positive" observations actually are. Accuracy, then, is a measure of how reliably a model predicts events with positive margins. This equation is used to determine how precise a measurement is: (22)

$$Precision = \frac{TP}{TP+FP} \quad (22)$$

Recall: the "number of selected relevant data items" is shown. How many of the good observations may be attributed to the algorithm's predictions is displayed. Recall is calculated as follows: recall = number of true positives / (number of true positives + number of false negatives) A measure of recall is the fraction of test data's Margin Positive pictures that were properly labelled (see equation (23)).

$$Recall = \frac{TP}{TP+FN} \quad (23)$$

Specificity: measures how well it assigns labels to pictures with a Margin Negative (see equation (24)).

$$Specificity = \frac{TN}{TN+FP} \quad (24)$$

F1 score: The In equation (25), accuracy and recall are weighted equally to generate the F1 score.

$$F1 - Score = 2 * \frac{Precision \times Recall}{Precision + Recall} \quad (25)$$

Table 5: Analysis of Proposed Model without Clustering (Hybrid Model)

Dataset	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Specificity**
Accident	85.5	87	86	86	86
Pumbs*	87	91	87	88	87
Retail	91.5	91.5	91.5	91.5	95
Yeast	95.2	95	96	95	96

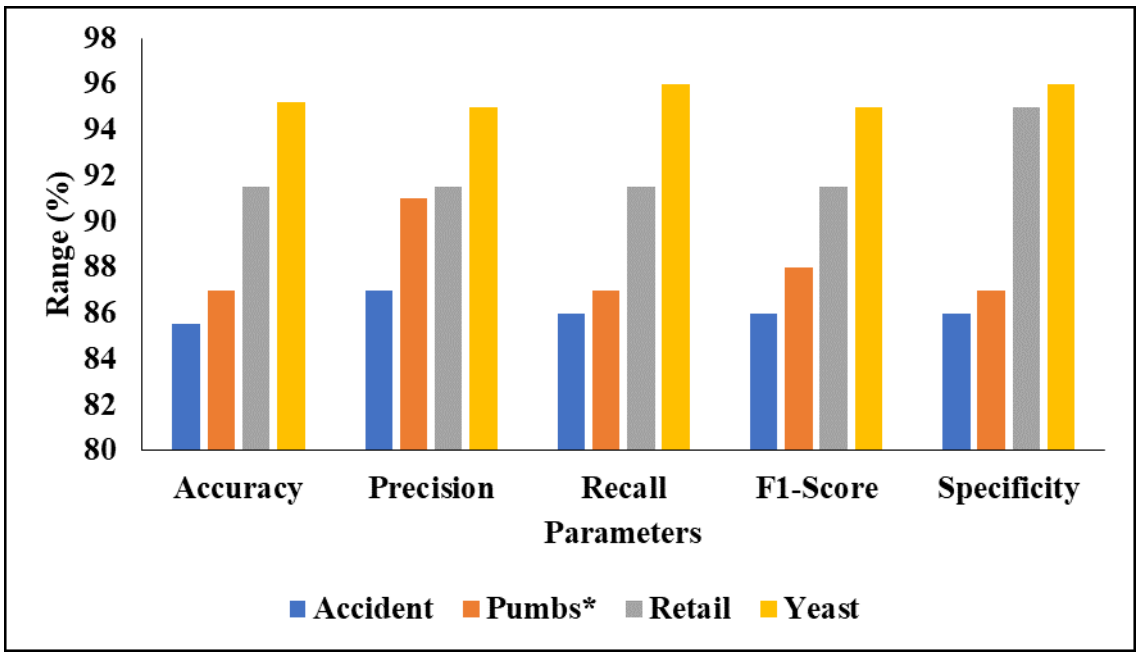

Figure 2: Graphical analysis of Proposed Model

For accident dataset, the proposed model achieved 85% to 87% of accuracy, precision, recall, F1-score and specificity. The model achieved nearly 95% to 96% of accuracy, precision, recall, F1-score and specificity for the Yeast dataset. While comparing with other three dataset, the model achieved better performance in Yeast dataset. When the model is tested with Retail dataset, it achieved 91.5% of accuracy, precision, recall and F1-score, then it achieved 95% of specificity. The reason for poor performance is that the unique id is directly given to extract the colossal pattern, where it consumes high computation time. Therefore, table 6 and Figure 3 presents the analysis of proposed model by considering hybrid optimization for colossal clustering.

Table 6: Analysis of Model with Clustering (Hybrid Model)

Dataset	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Specificity**
Accident	87.30	90.3	88.56	88.50	88.50
Pumbs*	92.67	91.45	91.67	91.39	90.67
Retail	91.34	90.78	91.78	91.39	96.27
Yeast	96.5	97.0	97.0	97.0	95.7

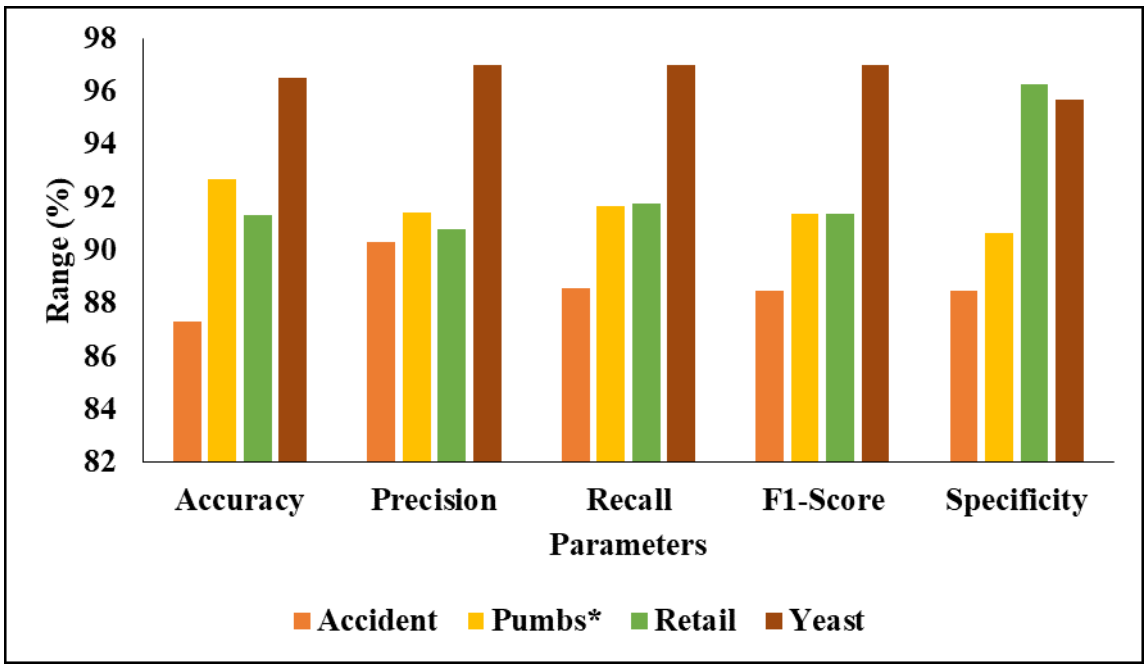

Figure 3: Analysis of Proposed Model with Clustering

When the models are tested with accident, the proposed model achieved 87.30% of accuracy, 90.3% of precision, 88.56% of recall, 88.50% of F1-score and 88.50% of specificity. The models achieved nearly 90% to 92% of accuracy, precision, recall, F1 score and specificity. When comparing with all datasets, the model achieved better performance in Yeast model, i.e., 95.7% of specificity, 96.5% of accuracy and 97% of precision, recall and F1-score. The model achieved 96.27% of specificity, where it achieved nearly 90% to 91% of accuracy, precision, recall and F1-score. The execution time for proposed model is tested and it is achieved in Table 7.

Table 7: Execution Time (S)

Dataset	**Minimum support**		
	3	**4**	**5**
Accident	**83**	**75**	**60**
Pumbs*	**45**	**38**	**36**
Retail	**19**	**18**	**17**
Yeast	**8**	**7**	**6**

For the minimum support 3, the model achieved 83s for accident, 45s for pumbs, 19s for retails and 8s for Yeast. When the minimum support is high, the execution time is less for all datasets. For instance, the model has 75s for accident, 18s for retail and 7s for yeast, where the proposed model achieved 6s for Yeast dataset, 17s for retail and 60s for accident dataset. Table 8 presents the comparative analysis with existing technique HCP-Miner [23], which is implemented, then results are averaged.

Table 8: Comparative analysis of Proposed Model with HCP-Miner

Methods	Dataset	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Specificity**
Proposed Model	**Accident**	87.30	90.3	88.56	88.50	88.50
	Pumbs*	92.67	91.45	91.67	91.39	90.67
	Retail	91.34	90.78	91.78	91.39	96.27
	Yeast	96.5	97.0	97.0	97.0	95.7
HCP-Miner [23]	**Accident**	85.14	87.56	85.91	86.47	85.93
	Pumbs*	90.67	88.42	87.20	87.26	86.48
	Retail	88.17	87.63	87.39	88.69	91.59
	Yeast	93.89	95.37	94.10	95.41	92.06

In the analysis of accuracy for accident dataset, the existing technique achieved 85.14% and proposed model achieved 87.30%. Both models achieved better performance on Yeast dataset in terms of accuracy, precision, recall, F1-score and specificity. For instance, the proposed model achieved 97% of precision, recall, F1-score, 96.5% of accuracy and 95.70% of specificity, where the existing model achieved 93.89% of accuracy, 95.37% of precision, 94.10% of recall, 95.41% of F1-score and 92.06% of specificity on Yeast dataset. The reason for better performance is that the clustering process is carried out by hybrid optimization model, where existing technique uses lattice array for constructing the sub-patterns.

5. Conclusion

This study introduces a method that, when applied to a huge database of sales records, can significantly improve the effectiveness of frequent item mining. This study examines the process of mining gigantic patterns and details the approach we developed to do so. Instead of picking random patterns, we use a meta-heuristics method (HHO-AOA) to choose which will serve as our core patterns. In this study, we provide a novel metaheuristic method that combines the Harris Hawks Optimizer with arithmetic optimisation (HHO-AOA). The hybrid approach is designed to improve the accuracy of clustering the itemset by increasing solution diversity. In this study, we offer a more effective strategy for mining massive patterns. The pumbs*, accident, and retail are near to yeast datasets were used in the performance evaluation. Empirical results show that the suggested model works better than alternative techniques when used to high-dimensional datasets. The evaluations on four different datasets demonstrated that our method outperformed the competition.

References

[1] Sornalakshmi, M., Balamurali, S., Venkatesulu, M., Krishnan, M.N., Ramasamy, L.K., Kadry, S. and Lim, S., 2021. An efficient apriori algorithm for frequent pattern mining using mapreduce in healthcare data. Bulletin of Electrical Engineering and Informatics, 10(1), pp.390-403.

[2] Davashi, R., 2021. ILUNA: Single-pass incremental method for uncertain frequent pattern mining without false positives. Information Sciences, 564, pp.1-26.

[3] Nguyen, T.L., Vo, B. and Snasel, V., 2017. Efficient algorithms for mining colossal patterns in high dimensional databases. Knowledge-Based Systems, 122, pp.75-89.

[4] Sohrabi, M.K. and Barforoush, A.A., 2012. Efficient colossal pattern mining in high dimensional datasets. Knowledge-Based Systems, 33, pp.41-52.

[5] Djenouri, Y., Belhadi, A., Srivastava, G. and Lin, J.C.W., 2023. Advanced Pattern-Mining System for Fake News Analysis. IEEE Transactions on Computational Social Systems.

[6] Leung, C.K., 2023. Big Data Mining and Analytics With MapReduce. In Encyclopedia of Data Science and Machine Learning (pp. 156-172). IGI Global.

[7] Mahardika, F., Alfiah, N. and Sumantri, R.B.B., 2023. Penerapan Metode FP Tree dan Frequent Pattern Growth pada Penerimaan Mahasiswa Baru STMIK. Blend Sains Jurnal Teknik, 1(3), pp.226-234.

[8] Nguyen, H., Le, N., Bui, H. and Le, T., 2023. A new approach for efficiently mining frequent weighted utility patterns. Applied Intelligence, 53(1), pp.121-140.

[9] Djenouri, Y., Belhadi, A., Srivastava, G. and Lin, J.C.W., 2023. A Secure Parallel Pattern Mining System for Medical Internet of Things. IEEE/ACM Transactions on Computational Biology and Bioinformatics.

[10] Chen, Y., Gan, W., Wu, Y. and Philip, S.Y., 2023. Privacy-Preserving Federated Mining of Frequent Itemsets. Information Sciences.

[11] Leung, C.K., 2023. Big Data Visualization of Association Rules and Frequent Patterns. In Encyclopedia of Data Science and Machine Learning (pp. 1284-1298). IGI Global.

[12] Patel, S.B., Shah, S.M. and Patel, M.N., 2023. An Efficient Search Space Exploration Technique for High Utility Itemset Mining. Procedia Computer Science, 218, pp.937-948.

[13] Dhanaseelan, R. and Jeyasutha, M., 2023. A novel fuzzy frequent itemsets mining approach for the detection of breast cancer. In Research Anthology on Medical Informatics in Breast and Cervical Cancer (pp. 511-531). IGI Global.

[14] Rehman, S.U., Khan, M.A., Nabi, H.U., Ali, S., Alnazzawi, N. and Khan, S., 2023. TKIFRPM: A Novel Approach for Topmost-K Identical Frequent Regular Patterns Mining from Incremental Datasets. Applied Sciences, 13(1), p.654.

[15] Yu, L., Gan, W., Chen, Z. and Liu, Y., 2023. IDHUP: Incremental Discovery of High Utility Pattern. Journal of Internet Technology, 24(1), pp.135-147.

[16] Vanahalli, M.K. and Patil, N., 2022. An efficient colossal closed itemset mining algorithm for a dataset with high dimensionality. Journal of King Saud University-Computer and Information Sciences, 34(6), pp.2798-2808.

[17] Tonon, A. and Vandin, F., 2022. gRosSo: mining statistically robust patterns from a sequence of datasets. Knowledge and Information Systems, 64(9), pp.2329-2359.

[18] Singh, I. and Jindal, R., 2023. Trust factor-based analysis of user behavior using sequential pattern mining for detecting intrusive transactions in databases. The Journal of Supercomputing, pp.1-33.

[19] Barrientos, R.J., Riquelme, J.A., Hernández-García, R., Navarro, C.A. and Soto-Silva, W., 2022. Fast kNN query processing over a multi-node GPU environment. The Journal of Supercomputing, pp.1-27.

[20] Liu, X., Zheng, L., Zhang, W., Zhou, J., Cao, S. and Yu, S., 2022. An evolutive frequent pattern tree-based incremental knowledge discovery algorithm. ACM Transactions on Management Information Systems (TMIS), 13(3), pp.1-20.

[21] Rambabu, D. and Govardhan, A., 2023. Optimization assisted frequent pattern mining for data replication in cloud: Combining sealion and grey wolf algorithm. Advances in Engineering Software, p.103401.

[22] Davashi, R., 2023. ITUFP: A fast method for interactive mining of Top-K frequent patterns from uncertain data. Expert Systems with Applications, 214, p.119156.

[23] Reddy, T.S., Sathya, R. and Nuka, M.R. (xxxx) 'HCP miner: an efficient heuristic-based clustering method for discovering colossal frequent patterns from high dimensional databases', Int. J. Engineering Systems Modelling and Simulation, Vol. x, No. x, pp. xxx–xxx.