An Evolutionary Model to Mine High Expected Utility Patterns from Uncertain Databases

Usman Ahmed, Jerry Chun-Wei Lin*, Gautam Srivastava, Rizwan Yasin, and Youcef Djenouri

***Abstract*—In recent decades, mobile or the Internet of Thing (IoT) devices are dramatically increasing in many domains and applications. Thus, a massive amount of data is generated and produced. Those collected data contain a large amount of interesting information (i.e., interestingness, weight, frequency, or uncertainty), and most of the existing and generic algorithms in pattern mining only consider the single object and precise data to discover the required information. Meanwhile, since the collected information is huge, and it is necessary to discover meaningful and up-to-date information in a limit and particular time. In this paper, we consider both utility and uncertainty as the majority objects to efficiently mine the interesting high expected utility patterns (HEUPs) in a limit time based on the multi-objective evolutionary framework. The benefits of the designed model (called MOEA-HEUPM) can discover the valuable HEUPs without pre-defined threshold values (i.e., minimum utility and minimum uncertainty) in the uncertain environment. Two encoding methodologies are also considered in the developed MOEA-HEUPM to show its effectiveness. Based on the developed MOEA-HEUPM model, the set of non-dominated HEUPs can be discovered in a limit time for decision-making. Experiments are then conducted to show the effectiveness and efficiency of the designed MOEA-HEUPM model in terms of convergence, hypervolume and number of the discovered patterns compared to the generic approaches.**

***Index Terms*—High expected utility pattern mining, data mining, multi-objective optimization, evolutionary computation.**

I. Introduction

Over the past few decades, pattern mining algorithms [1] showed the effectiveness to discover valuable information for decision-making. Apriori [2] is the most fundamental algorithm, which is used to find the association rules among the items in the databases. It first uses the minimum support threshold to discover the set of frequent itemsets. After that, the satisfied frequent itemsets are then combined, forming a set of rules and if the confidence of a rule is no less than a minimum confidence threshold, it is then defined as an association rule. The association-rule mining (ARM) has been widely utilized in many domains and applications to show its effectiveness in knowledge discovery. However, ARM only considers the binary database but ignores other factors, such as weight, importantness, interestingness, or quantity. An obvious example is for the basket-market analysis. Any items with their purchase quantities in a transaction are not considered in ARM. Another limitation of ARM is that all items are considered equally and treated as the same importances. Although ARM has been used to recognize the relationships of the items/sets in a transactional database, the valuable and useful patterns can thus be ignored due to the limitation of ARM; the incomplete information may produce the wrong strategies in decision-making.

High-utility itemset mining (HUIM) [3], [4] is an emerging topic, which considers both unit profit of the items and the quantity of the items to retrieve the set of high-utility itemsets (HUIs) from the quantitative databases. The purpose of HUIM is to reveal patterns that having high utility to users. If the utility of an item/set is no less than a pre-defined minimum utility threshold, it is then considered as a HUI. Based on the HUIM, more profitable products or relationships can thus be revealed, which can be used to make more efficient strategies in decision-making. Also, most existing methods [1] in pattern mining rely on a priori threshold value to discover the required knowledge, which is a non-trivial task since it needs the domain and expert knowledge to set an appropriate threshold to avoid the "rare-item" and "combinational explosion" problems.

Motivation: In a complex industrial environment, data encounters many challenges, i.e., the uncertainty about the data sources and the processing environment factors. Due to the uncertainty factor existing in many resources (e.g., Wifi system, RFID, wireless sensor network and GPS) [5], traditional data mining methods cannot be utilized to mine all the required knowledge from the uncertain databases. The reason is that both factors (i.e., the utility and uncertainty) are two different measures that bring the semantic and objective value for each pattern. Traditional data mining approaches use the utility (a semantic mechanism) as a measure to assess the value of the pattern, i.e., a pattern is available in the relevant data and is useful too. The uncertainty measure can be considered as an objective measure that assesses the reliability and existence of the pattern in terms of probability. Both factors are different from each other. It is a non-trivial task to both consider the utility and uncertainty factors for mining the HUIs in the uncertain databases. For most utility-based approaches, data sources are considered as precise, while data uncertainty factor is not taken into consideration. If the uncertainty factor is not considered, the extracted patterns may become useless, unreliable and lack of essential information

U. Ahmed and J. C. W Lin are with the Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, 5063, Bergen, Norway. Email: usman.ahmed@hvl.no, jerrylin@ieee.org. Website: http://ikelab.net. (*Corresponding author: Jerry Chun-Wei Lin)

G.Srivastava is with the Department of Mathematics & Computer Science, Brandon University, Canada. Email: srivastavag@brandonu.ca

R. Yasin is with the Department of Computer Science, Centre for Research in Data Science, Semantics & Scientometrics (CRDS), Capital University of Science and Technology, Islamabad, Pakistan. Email: rykhan2003@gmail.com

Y. Djenouri is with the SINTEF Digital, Oslo, Norway. Email: youcef.djenouri@sintef.no

This is the author accepted version of an article published in
IEEE Transactions on Emerging Topics in Computational Intelligence
http://dx.doi.org/10.1109/TETCI.2020.3000224

with low probability. Furthermore, more than two factors are considered together for decision-making, and they could have the trade-off and non-dominated relationships (i.e., price and distance to the downtown for booking a hotel room). To make an efficient decision in a limit time, it is necessary to have a robust approach that considers both uncertainty and utility factors together to extract the information.

In this paper, we first take the uncertainty and utility as the two objects for the consideration to find the non-dominated solutions based on evolutionary computation. The contributions of this paper are then listed below.

- In this study, we examine the problem by considering both utility and uncertainty objects (multi-objectives) simultaneously to discover the qualified non-dominated high expected utility patterns (HEUPs) from the uncertain databases based on the evolutionary computation.
- An **M**ulti-**O**bjective **E**volutionary **A**pproach to mine **H**igh **E**xpected **U**tility **P**attern **M**ining (MOEA-HEUPM) model is first developed, which can be used to find the required HEUPs in a limit time for the uncertain environment.
- The designed MOEA-HEUPM does not require the priori knowledge, i.e., minimum utility threshold or minimum uncertain threshold, for knowledge discovery, but the non-dominated patterns can be mined, which is much more unique and meaningful for decision-making.
- The weight-based *(Tchebycheff)* algorithm is utilized here for quickly obtaining the non-dominated solutions based on the multi-objective *(utility and uncertainty)* mechanism.
- Two encoding schemas are then developed and utilized in the designed MOEA-HEUPM model to show its effectiveness, and the experiments showed that the developed MOEA-HEUPM outperforms the generic pattern-mining algorithms in terms of convergence, hypervolume and number of the discovered patterns.

II. Literature Review

Association rule mining (ARM) is the most fundamental knowledge in Knowledge Discovery in Database (KDD), and the first algorithm is called Apriori [2], which was proposed by Agrawal and Srikant to find the relationships of the items in the databases. Apriori uses two phases to first find the set of frequent itemsets in the database based on the minimum support threshold. After that, the combinations of the frequent itemsets are formed to generate the set of association rules based on the minimum confidence threshold.

Since the occurrence frequency does not show the insight of the discovered patterns, for example, the amount of the purchased diamond in a shopping mall is much less than that of the number of clothes; however, the obtained profit of the diamond for the retailer is much more than that of the clothes. High-utility itemset mining (HUIM) [4], [6] was proposed to find the profitable items/sets from the databases, and it has been widely developed for the last decades. The HUIM takes both unit profit of the items and quantity of items as the consideration to discover the set of high-utility itemsets (HUIs) in databases. Since the original HUIM does not hold the downward closure property, thus the two-phase model called transaction-weighted utilization (TWU) [6] was designed to build the downward closure property by high transaction-weighted utilization itemsets (HTWUIs) for maintaining the correctness and completeness of the derived HUIs. Many extensions were presented and studied [7], and most of the pattern-mining algorithms including whether ARM or HUIM require to set a minimum threshold value to verify whether an item/set is considered as a valuable, important pattern. Some algorithms were studied to mine the HUIs without the threshold setting, for example, top-k HUIM [8]. Instead of using the minimum threshold as the standard metric to find the set of HUIs, the evolutionary computation was also involved in finding useful and meaningful information from databases. Kannimuthu *et al.* [9] proposed a GA-based model with ranked mutation operator to mine HUIs. However, the GA-based model requires an amount of computational cost, Lin *et al.* [10] then presented the PSO-based model for mining the set of HUIs. An effective OR/NOR tree [11] was also designed to verify the valid solutions in the evolutionary progress, which can provide more accurate HUIs. The ACO-based algorithm called HUIM-ACS [12] was then also developed to find the set of HUIs efficiently. The above studies focused, however, either on mining frequent itemsets or mining high utility itemsets individually using the priori parametric values (i.e., minimum support threshold or minimum utility threshold). The above approaches cannot be used to mine the interesting patterns with more than one object, i.e., uncertainty and utility together. As the rapid growth of Internet of Things (IoTs), varied data is collected from the uncertain environment, and it is necessary to design the robust approach to handle this situation.

The generic evolutionary computation (EC) [13] is a meta-heuristic approach, which is used to solve the NP-hard and optimization problems efficiently based on the single-objective fitness function in evolutionary progress. In evolutionary computation, multi-objective optimization is one of the most research areas and used in many domains and applications [14], [13]. MOEA/D is a generic framework that integrates the multi-objective evolutionary problem into small multi-objective optimization subproblems [15]. The MOEA/D uses the population-based approach to optimize these subproblems [15]. Based on the MOEA/D framework, it can produce a set of equally disseminated solutions and has a great convergence as compared to the other MOEA algorithms such as NSGA-II [16] and SPEA-II [17]. In the field of data mining, the multi-objective evolutionary algorithms (MOEAs) are commonly utilized to solve the classification [13], and clustering and feature selection [13], [18] problem since the multi-criteria are considered into those problems that have to be optimized [13]. For example, the interesting pattern in the database depends upon the multiple measures, i.e., interestingness, support, confidence comprehensibility, and lift.

Zhang *et al.* [19] developed the evolutionary progress to mine the frequent and utility patterns. This model does not require to give the priori parameters but discovers the non-dominated patterns regarding utility and frequency factors. However, this approach fails to find sufficient patterns for

decision-making. Djenouri *et al.* [20] developed several meta-heuristic algorithms for efficiently mining the high frequent and utility patterns from the databases. Up to now, non-existing models focus on mining the HUIs from the uncertain database, which is the major consideration in this paper.

Recently, many uncertain pattern mining algorithms mine useful patterns from the database including uncertain high-utility itemsets [21], frequent uncertain patterns (UFPs) - [22], [23], uncertain sequential patterns [24], [25], uncertain weighted frequent itemsets [26] and interesting uncertain patterns [27]. Liu *et al.* [24] proposed the uncertain sequential pattern mining algorithm based on the candidate generation approach and applied it to the sensor data set of the pollution monitoring network. Palacios *et al.* [25] applied fuzzy uncertain mining technique to mine the events in the uncertain health data of aero-engine condition monitoring. Lin *et al.* [21] proposed the novel framework named potential high-utility itemsets mining (PHUIM) in uncertain databases. PHUIM mines high utility uncertain patterns using the uncertainty tuple model. Lin *et al.* [26] applied Apriori-like two-phase approach to mine weighted frequent itemsets and proposed high upper bound to reduce the search space and irrelevant itemsets. Bui *et al.* [28] introduced the uncertain high utility closed itemsets that is based on Lin *et al.* [21] work. The model prunes non-closed potential high utility itemsets using the downward closure property and depth-first search to discover the required information.

Ahmed *et al.* [27] proposed a weighted uncertain interesting patterns approach that uses a tree-based structure for computing the prefix values. Lee *et al.* [23] proposed weight-based distinct uncertain mining approach and selectively mined the meaningful itemsets. Lee *et al.* [22] designed the list-based data structure that mined the frequent uncertain patterns. Yun *et al.* [29] highlighted the performance issue with the current incremental high utility mining algorithm that has a high false discovery rate and generates many patterns. The list-based mining technique was then presented to mine high utility patterns without candidate generations in incremental mining. Furthermore, Yun *et al.* [30] proposed a window-based method to mine utility patterns from data streams. The algorithm avoids the generate-and-test approach for many unpromising candidates and it can be efficiently performed in the complex dynamic systems.

In summary, most studies focused on using the uncertain database to mine high frequent or high utility patterns by two individual measures, i.e., support and utility factors. Fewer approaches used those two measures to mine the required information. The TKQ-Miner [31] considered two factors and used the bound estimation to mine the top-k high utility patterns. However, TKQ-Miner still required the priori parameters, i.e., the minimum support value and a minimum utility value. In real applications, users require domain knowledge to set the appropriate parameters to mine the high utility patterns.

Mai *et al.* [32] addressed the issue of mining a small set of non-redundant high-utility itemsets for better efficiency. The proposed NR-HARs algorithm mines a smaller set of HUIM using lattice structure. This model helps make timely decisions by mining a limited set of high utility itemsets. Baek *et al.* [33] proposed the uncertain itemset mining algorithm that uses the list-based method. The method extracts the commodities with large values that can help to find the non-defective products in the manufacturing plants [33]. Lee *et al.* [34] proposed a tree structure based on the uncertain frequent pattern mining approach. The method allows to examine the uncertain data and overcome the limitations of traditional approaches that is not able to concern the probabilities of items for pattern-mining progress [34]. Lin *et al.* [35] addressed the uncertain HUIM by considering the existence of probabilistic of the transactions based on a list structure. The designed method can avoid the multiple database scans and the huge amount of unpromising itemsets can be early removed by applying efficient pruning strategies. Gan *et al.* then proposed the HUPNU (mining High-Utility itemsets with both Positive and Negative unit profits from Uncertain databases) [36] to bother consider the positive and negative HUIs in the uncertain databases. The above methods required pre-defined threshold values (that can be varied in different datasets) to mine the HUIM. However, the proposed algorithm does not require any threshold values and mine the high utility itemset from the uncertain database by considering both uncertain and utility as multi-objectives in the pattern-mining progress.

III. Preliminaries and Problem Statements

Let $I = \{\imath_1, i_2, \ldots, i_m\}$ be a set of items, and let the uncertain transactional database be a set of transactions such as $D = \{T_1, T_2, \ldots, T_n\}$ where each transaction is a set of items belonging to I and has a unique identifier called T_{id}. Each item in a transaction has an uncertain value (probability of existence) such as $uv(i_k, T_c)$. Table I shows a simple example for the quantitative and uncertain database. Furthermore, Table II shows the unit profits of the items in the database.

TABLE I: The quantitative and uncertain database

TID	**Item: quantity, probability**
T_1	(*a*:5, 0.3); (*b*:3, 0.40); (*c*:6, 0.9)
T_2	(*c*:4, 0.75); (*d*:2, 0.9)
T_3	(*a*:7, 1.0); (*b*:8, 1.0); (*e*:2, 0.75)
T_4	(*a*:3, 0.9); (*c*:1, 0.9)
T_5	(*b*:2, 1.0); (*c*:4, 0.95); (*e*:4, 1.0)

TABLE II: The unit profit of the items

Item	**Profit**
a	8
b	3
c	8
d	3
e	5

Tables I and II are used as a running example in this paper. It is then described as follows: It has five transactions $(T_1, T_2, T_3, T_4, T_5)$. For example, transaction T_2 showed the items (c) and (d); their purchase quantities respectively are 4 and 2, and their uncertain values are 0.75 and 0.9 in T_2. The unit of profit of each sold item in the database is shown in Table II, i.e., the retailer receives \$8 as the profit of an unit for a sold item (a).

Definition 1: **(Item Utility in a Transaction)** The $u(i_k, T_c)$ is the utility of an item i_k in the transaction T_c, which is defined as:

$$u(i_k, T_c) = pr(i_k) \times q(i_k, T_c) \quad (1)$$

Example 1: For example, the utility of an item (a) in T_1 is calculated as: $u(a, T_1) = 5 \times \$8 = \$40$.

Definition 2: **(Itemset Utility in a Transaction)** The $u(X, T_c)$ is the utility of an itemset X in a transaction T_c, in which $i_k \in X \subseteq T_c$. It is defined as:

$$u(X, T_c) = \sum_{ik \in X} u(i_k, T_c) \quad (2)$$

Example 2: For example, the utility of an itemset (ab) in T_1 is calculated as: $u(ab, T_1) = \$40 + \$9 = \$49$.

Definition 3: **(Itemset Utility in a Database)** The utility of X in a transaction database D is denoted as $u(X)$, which is defined as:

$$u(X) = \sum_{X \subseteq T_c \wedge T_c \epsilon D} u(X, T_c) \quad (3)$$

Example 3: For example, the utility of an itemset (ac) in T_1 is calculated as: $u(ac) = u(a, T_1) + u(c, T_1) + u(a, T_4) + u(c, T_4) = \$40 + \$48 + \$24 + \$8 = \$120$.

Definition 4: **(Itemset Uncertainty in a Transaction)** The uncertain value of an itemset X in a transaction is denoted as $uc(X, T_c)$, which is defined as:

$$p(X, T_c) = \Pi_{x_i \epsilon X} p(x_i, T_c) \quad (4)$$

Example 4: For example, the probability of (a) in T_1 is calculated as: $p(a, T_1) = 0.3$. The probability of the itemset (ac) in T_1 is calculated as $p(ac, T_1) = p(a, T_1) \times p(c, T_1) = 0.3 \times 0.9 = 0.27$.

For the generic pattern-mining approaches [37], [1], most of the existing algorithms focus on evaluating the patterns by the individual thresholds, i.e., support, confidence, uncertainty, or utility. Also, it is not a trivial task to set the appropriate threshold for pattern evaluation since it can easily cause the "rare-item" and "combinational explosion" problems [19]. For example, when the threshold is set higher, the low number of patterns is then discovered. However, when the threshold is set higher, a huge amount of knowledge is then mined; the priori domain knowledge is required for the pattern mining to define the suitable threshold. It is not an easy task to find the most valuable patterns for decision-making. Moreover, most algorithms in KDD focus on mining the information from the precise database. In the recent IoT environment, the collected information brings, however, the uncertainty value in the databases. To solve the above limitations, the problem statement of this paper is defined below.

Problem Statement: For the quantitative and uncertain database, the purpose of this paper is to address both of the utility and uncertainty factors to discover the set of the non-dominated high expected utility patterns without the priori threshold settings for knowledge discovery. To reduce the computational cost for handling the big dataset and discovering the most up-to-date information, the multi-objective evolutionary computation (MOEA) is then utilized here to find the required information in a limit time.

IV. Designed MOEA-HEUPM Model

In this section, we first present the MOEA-based model called MOEA-HEUPM for mining the non-dominated high expected utility patterns (HEUPs) from the quantitative and uncertain databases. The proposed model is then described in Fig. 1.

The objective of the developed MOEA-HEUPM considers both utility and uncertainty factors to simultaneously discover the set of the non-dominated high expected utility patterns (HEUPs) from the uncertain databases. Both two measures somehow have a conflict to each other. In some cases, the patterns with higher utility do not have higher uncertainty while lower utility will not lead to higher utility. To both consider those two factors together without the priori knowledge, it can thus be considered as the two-objective optimization problem. Compared to the traditional pattern-mining algorithms, the developed model only illustrates fewer and valuable patterns for decision-making. Since both the utility and uncertainty are considered in the designed MOEA-HEUPM model, two objectives are then considered to be maximized and shown in equation 5. Details of the designed framework are then described below.

$$Max\, F(X) = \left\{ max\left(utility(X), uncertainty\,(X)^T\right\}\right. \quad (5)$$

A. Initialization

The population initialization is an essential step in the multi-objective evolutionary approach as the population may lead to a weak solution or take large computational requirements to converge [13]. In general, the initial population was randomly generated from the databases based on random item-selection. However, the random item-selection mechanism always leads to generate invalid patterns. i.e., the pattern does not exist in the transactions. In the first step of the developed MOEA-HEUPM, problem-specific initialization strategy was proposed. We use two strategies to select the population, which are named as **meta-itemset-selection** and **transaction-itemset-selection**. For the **transaction-itemset-selection**, the 50% individuals in a population are selected from the transactions in the database based on the utility probability. This method ensures that all the selected individuals are the ultimate solutions in the database. For the resting 50% individuals in a population of the **meta-itemset-selection**, only one item in a transaction is encoded and selected based on the uncertainty probability. The transaction-itemset is useful to quickly obtain the optimal solutions in the evolutionary progress (i.e., used in the crossover and mutation operations). The reason is that the transaction itemset is selected using the utility probability, in which it can quickly achieve the stable convergence of the solutions. In contrast, meta-itemset is useful to generate new offsprings from parents with higher diversity. The reason to obtain higher diversity of the derived solutions is that meta-itemset contains single items and has higher possibility to generate a new offspring by operating the crossover operator. Thus in this step, we first calculate the uncertain probability of the meta-itemset (Algorithm 1, lines 1 to 3). The results are then shown in Table III. After that, we further calculate

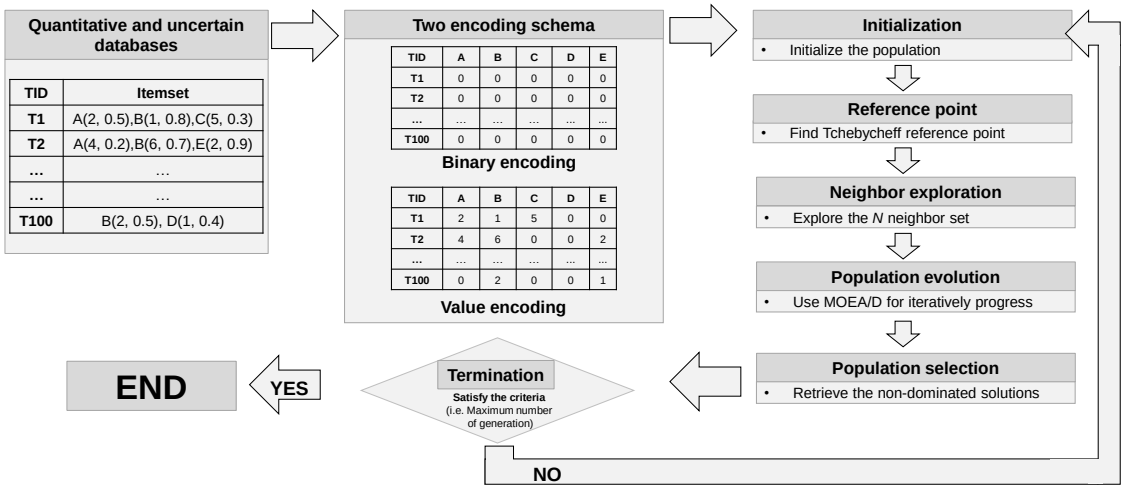

Fig. 1: The framework of the designed MOEA-HEUPM model

the utility probability of the transaction-itemset (Algorithm 1, lines 4 to 6). The results are shown in Table IV. Each 50% individuals of Tables III and IV are then generated forming the population (Algorithm 1, line 7). For example, the population size is initialized as 4. The designed MOEA-HEUPM respectively selects two individuals from Table IV based on the utility probability, and two individuals from Table III based on the uncertain probability. The algorithm for initialization is shown in Algorithm 1.

TABLE III: Meta-itemset-selection strategy

Itemset	Uncertainty	Uncertain Probability
a	1.3	0.16
b	1.4	0.17
c	2.75	0.34
d	0.9	0.11
e	1.75	0.22

TABLE IV: Transaction-itemset-selection strategy

TID	a	b	c	d	e	Utility	Utility Probability
$T1$	5	3	6	0	0	97	0.31
$T2$	0	0	4	2	0	38	0.12
$T3$	7	8	0	0	2	90	0.29
$T4$	3	0	1	0	0	32	0.10
$T5$	0	2	4	0	4	58	0.18

1) Encoding: In evolutionary computation, many encoding mechanisms were respectively studied based on different domains and applications. However, both binary and value encoding method is commonly used in EC since they can help to obtain the higher convergence and diversity of the derived solutions [13]. Thus in the designed model, after the initialization process of the population, encoding is performed on the initialized population (Algorithm 2, line 1). In the designed MOEA-HEUPM model, we first utilize two encoding schemas, i.e., binary and value encodings. For the binary encoding, if an item appears in a transaction, the position of the corresponding item is then defined as 1 for the encoding schema, and vice-versa. For the value encoding, we then set the quantity of the items as the encoding value in the schema.

Algorithm 1 MOEA-HEUPM: initialization

INPUT: The transaction data set D, population pop size, weight vector $\{w_1, w_2, \ldots, w_{pop}\}$, the size of neighbors n_s, crossover probability p_c, mutation probability p_m.
OUTPUT: Initialize individuals in a population.

1: **foreach** $i \in meta_{itemset}$ **do**
2: $\quad M \leftarrow (\frac{uncertain(D_i)}{uncertain(D)})$
3: **end foreach**
4: **foreach** $j \in transaction_{itemset}$ **do**
5: $\quad T \leftarrow (\frac{utility(D_j)}{utility(D)})$
6: **end foreach**
7: $P \leftarrow initial(M, T, pop)$
8: **Return** P

According to [38], the fitness function for a specific encoding schema depends on two factors, i.e., value and order. The schema that only preserves the order is a permutation and mostly used in order problems [38], whereas binary and value encoding schemas preserve the value as well as order. For each item in the transaction, the quantity is the value of a purchased item. The uncertainty value can also be used as an encoding value. However, the uncertain value does not have any direct relation as it was multiple with each item to give the uncertain value of the transaction.

2) Reference point: In the designed MOEA-HEUPM, the Tchebycheff reference point [19] is utilized to find the noon-dominated solutions based on the utility and uncertainty factors (Algorithm 2, line 2). The Tchebycheff value is used to evaluate the goodness of the encoding schema [19], which is defined in equation 6. The purpose of Tchebycheff value is to generate an efficient set that helps to find the non-dominated high expected utility patterns.

$$min\, g^{te}(X|w_i, z^*) \leftarrow max_{j=1}^{2} \left\{ w_i^j \,.\, (|F_j(X) - z^*|) \right\} \quad (6)$$

An example is given below to show the progress for finding the reference point. Assume that the pop contains the number of sub-problem and set W to be a set of even weights such

that $(w_1, w_2, \ldots, w_n)$, in which w_i^1 + w_i^2 = 1. Suppose that we have an encoding individual such that {1, 0, 1, 0, 0} with the utility value 22 and uncertainty value is 0.13, respectively. Suppose that the maximum utility of the database is calculated as 107, and the maximum uncertainty is calculated as 0.715. Based on the equation 6, the maximum utility and uncertainty values of the individual will be used as the reference point z^* (Algorithm 2, lines 3 to 5). Assume that the w_i^1 is set as 0.34 and w_i^2 is set as 0.66, respectively, which are the even weights for uncertainty and utility. We then calculate the the utility value of the individual based on equation 6 as $0.34 \times |22 - 107|$ = 28.9, and the uncertainty value of the individual as $0.66 \times |0.13 - 0.715|$ = 0.38. The Tchebycheff value is then calculated as $max(28.9,\ 0.38)$ that is 28.9. Thus, the Tchebycheff values of all individuals in the population are then estimated.

3) Neighbor exploration: After that, the neighbors of the individuals are then calculated (Algorithm 2, lines 3 to 5). For each weight vector w_i $(i < pop)$, in which *pop* is the population size initialized at the beginning of the algorithm. The Euclidean distance of the each individual between weight vector w_i in the population (*pop*) is then calculated as: $\sqrt{(|u(ind) - w_i^1|^2 + |p(ind) - w_i^2|^2)}$. The neighbor set is used to perform two-way crossover for the mutated child produced in population evolution. Details of the calculation for the reference point and the neighbor exploration are shown in Algorithm 2.

Algorithm 2 MOEA-HEUPM: reference point calculation and neighbor exploration

INPUT: P, the size of neighbors n_s, crossover probability p_c and mutation probability p_m.
OUTPUT: Population with neighbors.

1: $P \leftarrow Encode(P)$ ▷ Binary or Value encoding
2: $z^* \leftarrow$ initialize reference point.
3: **for** $all\ p \in pop$ **do**
4: $N_i \leftarrow$from P get the ns individual using Euclidean distance between any individual in P the weight vector W_i.
5: **end for**
6: **Return** N_i

B. Population evolution

In the evolution progress, the MOEA/D procedure is then utilized here of the developed MOEA-HEUPM model (Algorithm 3, lines 1 to 6). For each individual in Pop_i , one individual $Pop_i^{'}$ is randomly chosen from N_j (i.e., neighbors of Pop_i). Two-way crossover operator is then used between Pop_i, $Pop_i^{'}$. The mutation operator is also performed in the same way. The *Tchebyshev* value of the offsprings is then calculated compared with the value of the neighbor N. If the *Tchebyshev* value of the individual shows better goodness, it will be used to replace the Pop_i with the offspring and update the reference point z^* as mentioned (Algorithm 3, lines 2 to 5). This progress is then iteratively performed until the termination max_{gen} is achieved. The *Tchebyshev* value [19], [39] provides an alternative way to find the non-dominated solutions; thus, the generic methodology in pattern mining with the priori defined thresholds (i.e., minimum utility and minimum uncertainty) can thus be avoided.

Algorithm 3 MOEA-HEUPM: population evolution

INPUT: P, N_i and number of generations max_{gen}
OUTPUT: Non-dominated solutions

1: **while** max_{gen} **do**
2: **for** $all\ i \in pop$ **do**
3: $P_i^{'} \leftarrow Randomly\ select\ an\ individual from\ N_i$
4: $child \leftarrow CrossMutation(P_i^{'}, P_i)$
Compute $child$ objective function, if the Chebyshev value of the child is better than any individual ind in N_i, then replace ind with child and update reference point z^*
5: **end for**
6: **end while**
7: $Final_{solution} \leftarrow SelectNonDominatedItemsets(P)$
Apply fast non-dominated sorting strategy to get the non-dominated item sets from final population P.
8: **Return** $Final_{solution}$

C. Population selection

After the iteratively evolutionary progress of MOEA/D, the final population is generated and produced. The individuals in the population are then sorted [40] and selected as the non-dominated solutions for the final results (Algorithm 3, lines 7 to 8). Take population Pop containing K fronts $(1 < i < K)$ as an example to illustrate the steps. Firstly, all non-dominated solutions are found using Pop and assigned to F_1. After that, the assigned F_1 are removed as $Pop - F_1$. In this way, all fronts are extracted and selected for the final recommendation. The final evaluation is terminated when the algorithm reaches the maximal number of generations, as mentioned in Fig. 1. After that, the solutions are then produced. The steps of the algorithm are then described in Algorithm 3.

V. Experimental Evaluation

In this section, we have compared the proposed MOEA-HEUPM model with two baseline algorithms, i.e., U-Apriori algorithm [41] that requires the minimum uncertainty threshold and EFIM [42] that requires minimum utility threshold. The experiments were carried out on a Windows 10 PC with AMD Ryzen 5 PRO 3500U processor and 16 GB of RAM. For MOEA-HEUPM, the population size was set to 100, the maximal generation was set to 100, the size of neighbors was set to 10, the crossover probability was set to 1.0, and the mutation probability was set to 0.01.

For the experiments, we acquire six databases of varying characteristics. All these databases are available on SPMF library [43]. The characteristics of the datasets are shown in Table V that includes number of transactions *(#Transaction)* , number of distinct items *(#item)*, average length of the transactions in the dataset *(Ave-length)* and type of dataset (dense or sparse). The chess dataset has 3,196 number of

TABLE V: The used datasets in the experiments

Dataset	#Transactions	#items	Ave-Length	Type
Chess	3196	76	37	Dense
Mushroom	8124	120	23	Dense
Pumsb	49,045	21113	74	Dense
Kosarak	9899	1044	8	Sparse
Retail	8162	8345	10	Sparse
Accident	34018	468	33	Dense

transactions (game moves) and 76 items (position on the chessboard). The mushroom dataset has an average transaction length of 23, 120 distinct items and 8,124 transactions. The pumsb is the dense census data for population and housing. The average length of pumsb data is 74, having 49,046 transactions with 7,116 unique items. The kosarak dataset is a Hungarian news portal data that has information of click stream transactions. It contains 1,044 distinct items, 9,899 transactions and the average transaction length is 8. The retail dataset contains 8,162 customer transactions from a Belgian anonymous retail store. It has 8,345 unique market items and the average transaction length is 10. The accident dataset is a collection of Belgium road accidents (the anonymous incident happened on a public road). It has 34,018 transactions, 468 unique items, and the average transaction length is 33.

To generate the utility and uncertainty values for the datasets, we then used a similar strategy of the previous studied [7] for utility values. The individual probability of the items was randomly assigned to each transaction itemset in the range of 0.0 to 1 as used by the HUPM [21] by the normal distribution.

Two well-known measurements are used to assess the discovered patterns, respectively called hypervolume (HV) and Coverage (Cov) [19]. The HV is used to measure the distribution and convergence of the mined item-sets in the search space of the objective, i.e., *in our case, the values of uncertain and utility*. According to [19], [39], hypervolume can be defined as: Let $A = (x_1, x_2, \ldots, x_l) \subseteq X$ be a set of l decision vectors. The function $S(A)$ gives the volume enclosed by the union of the polytopes $p_1, p_2, \ldots, pl$ where each p_i is formed by the intersections of the following hyperplanes arising out of x_i , along with the axes: for each axis in the objective space, there exists a hyperplane perpendicular to the axis and passing through the point $(f_1(x_i), f_2(x_i), \ldots, f_k(x_i))$. In the two-dimensional case, each p_i represents a rectangle defined by the $points(0, 0)$ and $(f_1(x_i), f_2(x_i))$ [19].

The Cov is a common measure to show the diversity of the derived solutions and item-set distinct behaviors [19]. The larger value of Cov indicates the higher diversity of the derived solutions and non-dominated solution set has distinct item-sets. The equation for convergence is shown in equation 7.

$$Cov = N_d/N, \tag{7}$$

where N is the total number of discovered patterns and N_d is the different patterns.

A. Encoding schemas analysis

The encoding schemas, i.e., value and binary encoding, are then compared in terms of several generations for the Cov and HV metrics. The HV and Cov are then evaluated to verify the efficiency of two binary and value encoding schemas. The results for the two variants of encoding schemas are then showed in Fig. 2 and 3, respectively. From Fig. 2, the value encoding is converged with the low number of generations, whereas the binary encoding performs better when more generations are achieved especially in the sparse dataset. Thus, we can conclude that the binary encoding is more suitable for the dataset that has a large number of itemsets. In contrast, when the number of several itemsets is lower, or dataset is a dense type, then value encoding schema is considered to be used. Fig. 3 represents the HV that depicts the same behavior as Cov.

TABLE VI: The compared algorithms in terms of Cov, HV, and number of patterns

Algorithm	Thresolds	Metrics	Datasets					
			Chess	Mushrooms	Korasak	Pumsb	Accidents	Retails
U-Apriori (minUncertanity)	0.2	*Cov*	0.32	0.24	0.63	0.35	0.39	0.45
		HV	2267.90	5800.12	430.06	3410.52	23996.42	375.19
		# patterns	18562.00	12926.00	26857.00	35035.00	6080.00	17054.00
	0.4	*Cov*	0.11	0.07	0.33	0.12	0.15	0.16
		HV	1920.48	4909.05	281.66	2754.43	20337.12	229.31
		# patterns	6076.00	3631.00	14057.00	12223.00	2322.00	6150.00
	0.6	*Cov*	0.03	0.02	0.14	0.03	0.04	0.05
		HV	1573.19	3732.24	230.45	2257.76	16675.81	187.61
		# patterns	1469.00	875.00	5800.00	3448.00	602.00	1912.00
	0.8	*Cov*	0.00	0.00	0.04	0.01	0.00	0.01
		HV	1165.46	1979.21	179.25	953.30	13006.52	10.97
		# patterns	210.00	163.00	1623.00	555.00	76.00	376.00
MOEA-HEUPM	Binary	*Cov*	1.00	0.80	0.50	0.60	0.60	1.00
		HV	215088.60	94314.83	138219.34	83481.53	83481.53	2117792.17
		# patterns	2900.00	1939.00	509.00	900.00	900.00	10766.71
	Value	*Cov*	1.00	0.65	0.50	1.00	1.00	1.00
		HV	138219.34	146600.20	146600.20	134567.41	134567.41	756963.07
		# patterns	486.00	2039.00	1929.00	500.00	500.00	9818.60
EFM (minUtil)	0.20	*Cov*	0.50	0.50	0.12	0.50	0.50	0.50
		HV	-	-	-	-	-	-
		# patterns	95381.00	4801.00	8.00	34270.00	3756.00	35288968.00

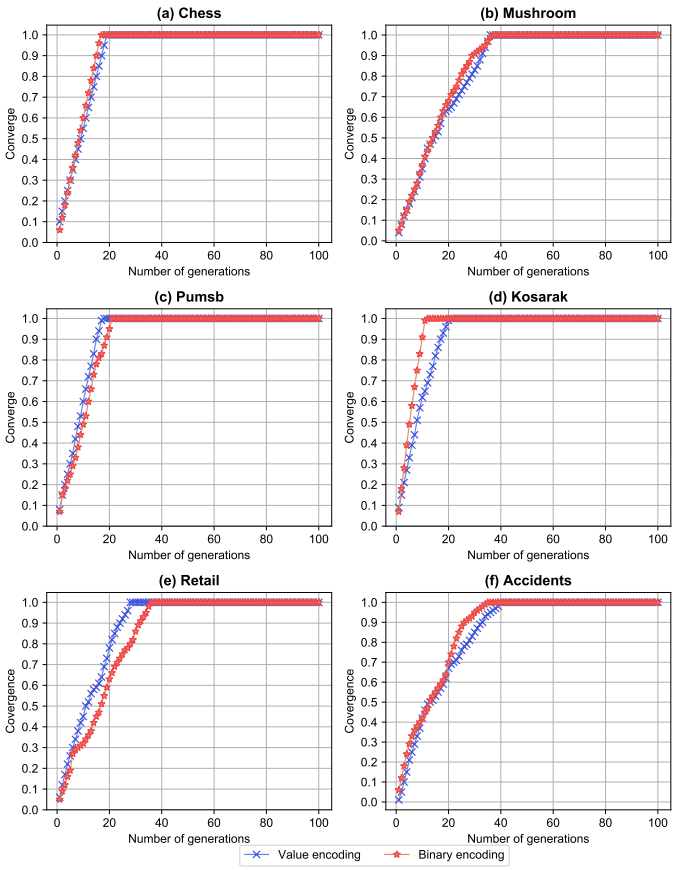

Fig. 2: The Cov of two encoding schemas

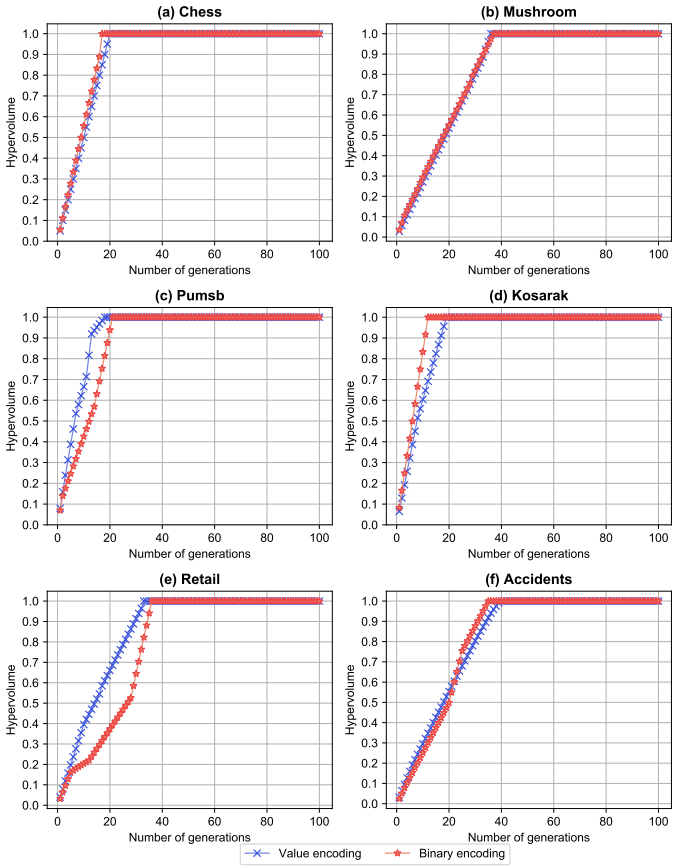

Fig. 3: The HV of two encoding schemas

B. Pattern analysis

Since there is no existing algorithm working on the evolutionary progress by considering both utility and uncertainty factors, the standard baseline algorithms called U-Apriori [41] and EFIM [42] are then compared with the designed model in terms of HV, Cov, and the number of generated patterns. However, those two algorithms require to have the priori knowledge to set the threshold value whether for utility or uncertainty, we then respectively set them from 20% to 80% with 20% increment each time for the U-Apriori, and set 20% for the EFIM since the resting thresholds (i.e., 40%, 60% and 80%) generate empty patterns (HUIs) in the mining progress. The number of discovered patterns is also shown in Table VI.

The patterns produced by MOEA-HEUPM is at least 5 to 10 times smaller than the number of patterns generated by the U-Apriori and EFIM. For example, the number of the pattern produced by the proposed algorithm on six datasets without any threshold value is 486 and maximum is 2,900, respectively, as mention in Table VI. However, U-Apriori generated a minimum of 12,926 and maximum of 35,035 and EFIM generated a minimum of 67 and a maximum of 190,573. Due to this reason, both U-Apriori and EFIM require more computational time, whereas the proposed model is quickly converged and provides fewer patterns for decision-making.

As mention in Table VI, when we use the low value for the uncertainty, the magnitude of the U-Apriori is relatively high for both measures, i.e., Cov and HV in the dense and sparse datasets. When the threshold is set higher for the uncertainty, the magnitude of the U-Apriori is dropped in datasets, and it generated a huge amount of patterns, as it can be seen in Table VI. For this reason, it is not an efficient algorithm for mining high uncertain patterns, whereas EFIM is not able to get the solution when the utility threshold is set higher, i.e., more than 20%. The algorithm performs better in the kosarak dataset. However, the EFIM algorithm only generated 67 candidates, as seen in Table VI. When the EFIM algorithm is running on other datasets, it extracted many patterns. The reason is that it only considers the utility factor but the uncertainty. Thus, both of those two standard algorithms cannot handle the uncertainty and utility factors together to obtain valuable and meaningful patterns.

Furthermore, the developed MOEA-HEUPM gets the highest Cov value with a maximum of 1 and a minimum of 0.50, as it can be seen in Table VI. In the case of HV, the proposed model has achieved the maximum value of 138,219 and minimum of 83,481. However, the number of generated patterns is much less than the other two approaches, and those derived patterns are the non-dominated solutions for making the efficient decisions. In the experiments, the uncertainty and utility values are usually large in dense databases (i.e., chess, mushroom, kosarak, retails and accidents) that results in the large value of HV. The itemset size of the dense dataset is small that causes no overlapping item sets. It is thus resulting in higher Cov. The higher Cov is also due to meta-itemset and transaction-itemset population initialization method as it helps to get higher diversity of the solutions after each generation. For sparse dataset (i.e., Pumsb), the itemset contains overlapping itemset due to sparse nature, as mentioned in Table VI. For this reason, two-bit or more meta-itemset should be used to get higher convergence values. Thus the MOEA-HEUPM was able to extract the high expected utility patterns based on the magnitude of Cov and HV. To be concluded, for the sparse dataset, two-bit or more meta-itemset should be considered to obtain better solutions.

C. Scalability analysis

In this section, we analyze the scalability of the proposed algorithm compare to the other algorithms in the synthetic dataset $T10I4N4KDXK$ (X is the size of the dataset in terms of the number of the transaction). The synthetic dataset is generated by IBM Quest Synthetic Data generator [44]. The results under different threshold values are shown in in Table VII. From the results, it can be seen that the U-Apriori and EFM both generated a huge number of patterns and have less performance in terms of Cov and HV compared to the designed MOEA-HEUPM. When the uncertain threshold is set higher, the number of patterns is then decreased, as well as the HV. It shows the same trend in the results of EFM. In this experiments, it is not able to calculate the hypervolume as a large number of patterns discovered in EFIM. Furthermore, the utility and uncertainty are both required, which is difficult for the calculation due to the large number transaction and a huge number of extracted patterns in EFIM. The threshold in EFIM is set very high because it generates a large number

TABLE VII: Scalability analysis

Algorithm	Thresolds	Metrics	T40I10DXK				
			X=20	*X=40*	*X=60*	*X=80*	*X=100*
U-Apriori (minUncertainty)	*0.2*	Cov	1	0.56	0.39	0.39	0.38
		HV	41818.9	41820.0	41819.5	41819.8	41820.0
		# patterns	164589	380353	624123	624760	625398
	0.4	Cov	1	0.56	0.39	0.39	0.39
		HV	203.9	203.9	203.9	203.9	203.9
		# patterns	30756	71415	117669	118077	118485
	0.6	Cov	1	0.56	0.39	0.38	0.38
		HV	163.0	163.04	163.04	163.06	163.03
		# patterns	8197	18808	30840	31080	31318
	0.8	Cov	1	0.51	0.34	0.28	0.24
		HV	120.69	120.04	121.50	121.33	121.89
		# patterns	116	239	365	446	523
MOEA-HEUPM	Binary	*Cov*	0.97	0.97	0.96	0.97	0.98
		HV	2013667.9	4110374.0	4624663.4	8113592.7	7249258.8
		# patterns	4182	3570	2754	3774	6222
	Value	*Cov*	0.97	0.96	0.97	0.97	0.97
		HV	1515809.4	4109259.5	4622814.3	6111761.8	9586723.6
		# patterns	4998	3264	4386	4284	4284
EFM (minUtility)	0.8	*Cov*	1	0.74	0.69	1	0.69
		HV	-	-	-	-	-
		# patterns	71705	250558	733474	239405	518260

of patterns. The U-Apriori and EFM achieved significantly higher convergence rates and numerous patterns when the synthetic dataset number of transaction is $20K$. However, when the number of the transactions increases, the Cov is then decreased. However, the proposed MOEA-HEUPM still obtains good performance as the increasing of database size. The empirical evidence thus shows that the proposed model is useful to extract a set of high utility uncertain itemsets without a threshold value.

VI. Conclusion and Future work

Utility-pattern mining can discover more valuable information rather than traditional and generic association-rule mining, which attracts more attention in recent decades. As the rapid growth of technologies, data uncertainty is also considered as an important factor in the pattern-mining field. Most existing and generic methods have, however, to set a priori threshold for mining the required information, which is a non-trivial task and unreasonable in many domains and applications. In this paper, we first consider both the utility and uncertainty factors together and develop an evolutionary model called MOEA-HEUPM to find the non-dominated high expected-utility patterns (HEUPs) based on MOEA/D. Two binary and value encoding schemas are then used in the developed MOEA-HEUPM to show its effectiveness. Based on the developed MOEA-HEUPM, it is efficient to produce fewer but valuable non-dominated HEUPs for decision-making without the priori threshold value in the uncertain environment. Experiments are then conducted to show the efficiency and effectiveness of the designed model compared to the generic and standard U-Apriori and EFIM model in terms of convergence, hypervolume and number of the discovered patterns.

Since the proposed multi-objective model is first work to use both objectives, i.e., utility and uncertainty in an uncertain database, there are many new opportunities that can be pursued in the future works. For example, the model can address the tuple uncertainty as to the multi-objective problem where each tuple has an associated probability distribution. More factors can also be considered for the multi-objective problems to derive fewer but valuable non-dominated patterns. It is also an interesting topic to extend the proposed MOEA-based model to the dynamic data mining, stream data mining and top-k pattern mining fields.

References

[1] P. Fournier-Viger, J. C. W. Lin, B. Vo, T. C. Truong, J. Zhang, and H. B. Le, "A survey of itemset mining," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 7, no. 4, p. e1207, 2017.

[2] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the International Conference on Very Large Data Bases*, 1994, pp. 487–499.

[3] P. Fournier-Viger, C. Wu, S. Zida, and V. S. Tseng, "FHM: faster high-utility itemset mining using estimated utility co-occurrence pruning," in *Proceedings of the International Symposium on Methodologies for Intelligent Systems*, vol. 8502, 2014, pp. 83–92.

[4] W. Gan, J. C. W. Lin, P. Fournier-Viger, H. Chao, V. S. Tseng, and P. S. Yu, "A survey of utility-oriented pattern mining," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2019.

[5] T. Bernecker, H. Kriegel, M. Renz, F. Verhein, and A. Züfle, "Probabilistic frequent itemset mining in uncertain databases," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 119–128.

[6] Y. Liu, W. keng Liao, and A. Choudhary, "A fast high utility itemsets mining algorithm," in *Proceedings of the International Workshop on Utility-based Data Mining*, 2005, pp. 90–99.

[7] M. Liu and J. Qu, "Mining high utility itemsets without candidate generation," in *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2012, pp. 55–64.

[8] H. Ryang and U. Yun, "Top-k high utility pattern mining with effective threshold raising strategies," *Knowledge-Based Systems*, vol. 76, pp. 109–126, 2015.

[9] S. Kannimuthu and K. Premalatha, "Discovery of high utility itemsets using genetic algorithm with ranked mutation," *Applied Artificial Intelligence*, vol. 28, no. 4, pp. 337–359, 2014.

[10] J. C. W. Lin, L. Yang, P. Fournier-Viger, J. M. Wu, T. Hong, S. L. Wang, and J. Zhan, "Mining high-utility itemsets based on particle swarm optimization," *Engineering Applications of Artificial Intelligence*, vol. 55, pp. 320–330, 2016.

[11] J. C. W. Lin, L. Yang, P. Fournier-Viger, T. Hong, and M. Voznák, "A binary PSO approach to mine high-utility itemsets," *Soft Computing*, vol. 21, no. 17, pp. 5103–5121, 2017.

[12] J. M. T. Wu, J. Zhan, and J. C. W. Lin, "An ACO-based approach to mine high-utility itemsets," *Knowledge-Based Systems*, vol. 116, pp. 102–113, 2017.

[13] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, "Survey of multiobjective evolutionary algorithms for data mining: Part II," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 20–35, 2014.

[14] F. Zou, D. Chen, D. S. Huang, R. Lu, and X. Wang, "Inverse modelling-based multi-objective evolutionary algorithm with decomposition for community detection in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 513, pp. 662–674, 2019.

[15] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Adaptive replacement strategies for MOEA/D," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 474–486, 2016.

[16] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[17] R. Shi and K. Y. Lee, "Multi-objective optimization of electric vehicle fast charging stations with SPEA-II," *IFAC-PapersOnLine*, vol. 48, no. 30, pp. 535–540, 2015.

[18] U. Ahmed, M. Aleem, Y. N. Khalid, M. A. Islam, and M. A. Iqbal, "RALB-HC: A resource-aware load balancer for heterogeneous cluster," *Concurrency and Computation: Practice and Experience*, p. e5606, 2019.

[19] L. Zhang, G. Fu, F. Cheng, J. Qiu, and Y. Su, "A multi-objective evolutionary approach for mining frequent and high utility itemsets," *Applied Soft Computing*, vol. 62, pp. 974–986, 2018.

[20] Y. Djenouri, P. Fournier-Viger, A. Belhadi, and J. C. Lin, "Metaheuristics for frequent and high-utility itemset mining," in *Studies in Big Data*. Springer, 2019, pp. 261–278.

[21] J. C. W. Lin, W. Gan, P. Fournier-Viger, T. Hong, and V. S. Tseng, "Efficient algorithms for mining high-utility itemsets in uncertain databases," *Knowledge-Based Systems*, vol. 96, pp. 171–187, 2016.

[22] G. Lee and U. Yun, "A new efficient approach for mining uncertain frequent patterns using minimum data structure without false positives," *Future Generation Computer Systems*, vol. 68, pp. 89–110, 2017.

[23] G. Lee, U. Yun, and H. Ryang, "An uncertainty-based approach: Frequent itemset mining from uncertain data with different item importance," *Knowledge-Based Systems*, vol. 90, pp. 239–256, 2015.

[24] Y. Liu, "Mining time-interval univariate uncertain sequential patterns," *Data & Knowledge Engineering*, vol. 100, pp. 54–77, 2015.

[25] A. M. Palacios, A. Martínez, L. Sánchez, and I. Couso, "Sequential pattern mining applied to aeroengine condition monitoring with uncertain health data," *Engineering Applications of Artificial Intelligence*, vol. 44, pp. 10–24, 2015.

[26] J. C. W. Lin, W. Gan, P. Fournier-Viger, T. Hong, and V. S. Tseng, "Weighted frequent itemset mining over uncertain databases," *Applied Intelligence*, vol. 44, no. 1, pp. 232–250, 2016.

[27] A. U. Ahmed, C. F. Ahmed, M. Samiullah, N. Adnan, and C. K. S. Leung, "Mining interesting patterns from uncertain databases," *Information Sciences*, vol. 354, pp. 60–85, 2016.

[28] N. Bui, B. Vo, V. Huynh, C. W. Lin, and L. T. T. Nguyen, "Mining closed high utility itemsets in uncertain databases," in *Proceedings of the Symposium on Information and Communication Technology*, 2016, pp. 7–14.

[29] U. Yun, H. Nam, G. Lee, and E. Yoon, "Efficient approach for incremental high utility pattern mining with indexed list structure," *Future Generation Computer Systems*, vol. 95, pp. 221–239, 2019.

[30] U. Yun, G. Lee, and E. Yoon, "Efficient high utility pattern mining for establishing manufacturing plans with sliding window control," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 9, pp. 7239–7249, 2017.

[31] L. Zhang, P. Luo, E. Chen, and M. Wang, "Revisiting bound estimation of pattern measures: A generic framework," *Information Sciences*, vol. 339, pp. 254–273, 2016.

[32] T. Mai, L. T. T. Nguyen, B. Vo, U. Yun, and T. Hong, "Efficient algorithm for mining non-redundant high-utility association rules," *Sensors*, vol. 20, no. 4, p. 1078, 2020.

[33] Y. Baek, U. Yun, E. Yoon, and P. Fournier-Viger, "Uncertainty based pattern mining for maximizing profit of manufacturing plants with list structure," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2019.

[34] G. Lee, U. Yun, and K. Lee, "Analysis of tree-based uncertain frequent pattern mining techniques without pattern losses," *The Journal of Supercomputing*, vol. 72, no. 11, pp. 4296–4318, 2016.

[35] J. C. W. Lin, W. Gan, P. Fournier-Viger, T. Hong, and V. S. Tseng, "Efficiently mining uncertain high-utility itemsets," *Soft Computing*, vol. 21, no. 11, pp. 2801–2820, 2017.

[36] W. Gan, J. C. W. Lin, P. Fournier-Viger, H. C. Chao, and V. S. Tseng, "Mining high-utility itemsets with both positive and negative unit profits from uncertain databases," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, ser. Lecture Notes in Computer Science, vol. 10234, 2017, pp. 434–446.

[37] W. Gan, J. C. W. Lin, P. Fournier-Viger, H. Chao, and P. S. Yu, "A survey of parallel sequential pattern mining," *ACM Transactions on Knowledge Discovery from Data*, vol. 13, no. 3, pp. 25–34, 2019.

[38] A. Kumar, "Encoding schemes in genetic algorithm," *The International Journal of Advanced Research in IT and Engineering*, vol. 2, no. 3, pp. 1–7, 2013.

[39] F. G. Mohammadi, M. H. Amini, and H. R. Arabnia, "Evolutionary computation, optimization, and learning algorithms for data science," in *Advances in Intelligent Systems and Computing*, 2020, pp. 37–65.

[40] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to nondominated sorting for evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 201–213, 2015.

[41] M. Hooshsadata, S. Bayat, P. Naeimi, M. S. Mirian, and O. R. Zaiane, "Uapriori: an algorithm for finding sequential patterns in probabilistic data," *Uncertainty Modeling in Knowledge Engineering and Decision Making*, pp. 907–912, 2012.

[42] S. Zida, P. Fournier-Viger, J. C. W. Lin, C. Wu, and V. S. Tseng, "EFIM: A highly efficient algorithm for high-utility itemset mining," in *Proceedings of the Mexican International Conference on Artificial Intelligence*, vol. 9413, 2015, pp. 530–546.

[43] P. Fournier-Viger, J. C. W. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z. Deng, and H. T. Lam, "The SPMF open-source data mining library version 2," in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, vol. 9853, 2016, pp. 36–40.

[44] R. Agrawal, M. Mehta, J. C. Shafer, R. Srikant, A. Arning, and T. Bollinger, "The quest data mining system," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 244–249.

Usman Ahmed Usman Ahmed is a PhD candidate at the Western Norway University of Applied Sciences (HVL). He has rich experience in building and scaling high-performance systems based on data mining, natural language processing, and machine learning. His research interests are sequential data mining, heterogeneous computing, natural language processing, recommendation systems, and machine learning.

Jerry Chun-Wei Lin received his Ph.D. from the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan in 2010. He is a full Professor at Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway. He has published more than 300 research articles in refereed journals (IEEE TKDE, IEEE TCYB, IEEE SysJ, IEEE SensJ, ACM TKDD, ACM TDS, ACM TMIS) and international conferences (IEEE ICDE, IEEE ICDM, DASSFA, PKDD, PAKDD). His research interests include data mining, soft computing, artificial intelligence and machine learning, and privacy preserving and security technologies. He is the Editor-in- Chief of the International Journal of Data Science and Pattern Recognition, Associate/Guest Editor of, IEEE Access, JIT, PlosOne, International Journal of Interactive Multimedia and Artificial Intelligence, IEEE TFS, IEEE TII, Applied Sciences, and Sensors. He is the Fellow of IET (FIET), senior member for both IEEE and ACM.

Gautam Srivastava was awarded his B.Sc. degree from Briar Cliff University in the U.S.A. in the year 2004, followed by his M.Sc. and Ph.D. degrees from the University of Victoria in Victoria, British Columbia, Canada in the years 2006 and 2012, respectively. Dr. G, as he is popularly known, is active in research in the field of Cryptography, Data Mining, Security and Privacy, and Blockchain Technology. In his 5 years as a research academic, he has published a total of 45 papers in high-impact conferences in many countries and in high-status journals (SCI, SCIE). He is an IEEE Senior Member.

Rizwan Yasin research interest are data mining, machine learning and natural language processing. He was awarded Gold Medal Master of Science in Computer Science from Capital University of Science and Technology, Islamabad, Pakistan. He is associated with industry at various levels for the last 15 years.

Youcef Djenouri obtained the PhD in Computer Engineering from the University of Science and Technology USTHB Algiers, Algeria, in 2014. He was granted a post-doctoral fellowship from the Unist university on South Korea, and he worked on BPM project supported by Unist university in 2016. In 2017, he was postdoctoral research at Southern Denmark University, where he has working on urban traffic data analysis. He was granted a post-doctoral fellowship from the European Research Consortium on Informatics and Mathematics (ERCIM), and worked at the Norwegian University of Science and Technology (NTNU), in Trondheim, Norway. He is working as the researcher in SINTEF Digital, Oslo, Norway. He is working on topics related to artificial intelligence and data mining, with focus on association rules mining, frequent itemsets mining, parallel computing, swarm and evolutionary algorithms and pruning association rules.