

## Búsqueda armónica binaria para la selección de atributos

Máximo E. Pacheco-Martínez, Maya Carrillo-Ruíz

Benemérita Universidad Autónoma de Puebla,  
Facultad de Ciencias de la Computación,  
México

{maximo.pachecomartinez, maya.carrilloruiz}@viep.com.mx

**Resumen.** La selección de atributos es un proceso crucial en el aprendizaje automático, la cual, ayuda a eliminar información irrelevante, reduciendo el número de atributos y disminuyendo la complejidad del modelo. Debido a que es un problema NP-Completo, se propone usar el algoritmo de búsqueda armónica para llevar a cabo dicha reducción. El proceso descrito en este artículo, consiste en obtener distintos subconjuntos (arreglos binarios) del conjunto original de atributos, evaluar su aptitud y haciendo uso de la búsqueda armónica determinar el mejor de ellos. Este proceso es ejecutado sobre distintas bases de datos de índole médica, utilizando Redes Neuronales Artificiales como clasificador. Los resultados muestran que mediante la modificación al algoritmo de Búsqueda Armónica (Binaria), se logra reducir la dimensionalidad de atributos de las distintas bases médicas. Además, dichos resultados son comparados con otros algoritmos usados para el mismo propósito.

**Palabras clave:** Búsqueda armónica, búsqueda armónica binaria, redes neuronales artificiales, selección de atributos.

### Binary Harmonic Search for Feature Selection

**Abstract.** Feature selection is a crucial process in machine learning, which helps to eliminate irrelevant information by reducing the number of features and decreasing the model's complexity. Since it is an NP-Complete problem, we propose using the harmonic search algorithm to carry out this reduction. The process described in this article involves obtaining various subsets (binary arrays) from the original feature set, evaluating their fitness, and using harmonic search to determine the best one. This process is executed on various medical databases, using Artificial Neural Networks as a classifier. The results show that by modifying the Harmonic Search (Binary) algorithm, we can reduce the dimensionality of features in different medical databases. Additionally, these results are compared with other algorithms used for the same purpose.

**Keywords:** Harmonic search, binary harmonic search, artificial neural networks, feature selection.

## 1. Introducción

La selección de atributos permite reducir la dimensión del problema, obteniendo modelos menos complejos con un porcentaje aceptable en la clasificación. Similar al aprendizaje automático, se divide en dos tipos Supervisado y no Supervisado, únicamente se considera este último para el presente artículo. Los métodos Supervisados, a su vez se dividen en filter, intrinsic y wrapper. Los métodos filter están orientados a la estadística, con el uso del Coeficiente de Pearson y/o Chi cuadrada.

Los métodos de intrinsic a Árboles de Decisión y/o Regularización Lasso. Finalmente, los métodos wrapper a los Algoritmos Genéticos y/o a la Eliminación Recursiva de Atributos. El uso de la búsqueda armónica binaria cae dentro de los métodos wrapper, los cuales consisten en crear modelos a partir de subconjuntos de características.

En la sección 2, se hace un breve análisis de los trabajos relacionados con el tema. En la sección 3, se describe el algoritmo de búsqueda armónica. En la sección 4, se describe la codificación que tendrá la selección de atributos, así como el algoritmo de búsqueda armónica binaria. En la sección 5, se consolidan los conceptos anteriores dando la estructura del modelo que se propone. En la sección 6, se presentan los experimentos. Por último, en la sección 7 se encuentran las conclusiones.

## 2. Trabajo relacionado

Cuando se tienen problemas donde la función es multimodal, o problemas de combinatoria, donde explorar todas las opciones por fuerza bruta no es viable, se suele recurrir a las metaheurísticas como alternativa.

Algo similar ocurre en el caso de la selección de atributos, donde diversas metaheurísticas son utilizadas para este problema.

Como se mencionó en la primera sección, existen distintos tipos de métodos para la selección de atributos, una variante que engloba varios de ellos es el método GCSA (Genetic Crow Search Algorithm) [1] que utiliza una combinación de Árboles de Decisión (DT, Decision Tree), Máquina de Soporte Vectorial (SVM, Support Vector Machine), Naïve Bayes (NB) y Bosques Aleatorios (RF, Random Forest).

También se ha propuesto para este problema el Algoritmo Binario del Murciélago (BBT, Binary Bat Algorithm), incluso usado para el aprendizaje no supervisado [2]. Otra metaheurística popular es el Enjambre de Partículas (PSO, Particle Swarm Optimization) utilizado para determinar la frecuencia cardíaca fetal [3]. En cuanto a la Búsqueda Armónica, esta se ha aplicado para mejorar el comportamiento de los aparatos auditivos [4].

## 3. Búsqueda armónica

La búsqueda armónica es una metaheurística inspirada en la armonía musical [5], donde un conjunto de instrumentos emite notas dentro de un rango y tonalidad que al sonar al unísono forman una armonía. Similar a todas las metaheurísticas, se parte de una población inicial o en este caso llamada Memoria Armónica (HM, Harmony

**Tabla 1.** Representación de un subconjunto de atributos.

$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$
0	1	1	0	0	1

Memory), donde cada vector o individuo de la población representa las notas que toca un instrumento. El tamaño de la memoria armónica se denota como HMS (Harmony Memory Size) y la dimensión del problema como  $m$ :

$$HM = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{HMS1} & x_{HMS2} & \cdots & x_{HMSm} \end{bmatrix}. \quad (1)$$

Para producir una nueva armonía (un individuo  $\mathbf{x}^{new}$ ), se toma un elemento al azar de la memoria armónica, a este proceso se le denomina **consideración de la memoria armónica** (HMC, Harmony Memory Consideration). A grandes rasgos, es, generar un número aleatorio (random) y compararlo respecto a un umbral de consideración (HMCR Harmony Memory Consideration Rate), si es menor, entonces el elemento  $j$  del individuo  $i$  será dicho valor de la memoria armónica. Es decir:

$$\text{si random} < \text{HMCR}: x_{ij}^{new} = x_{ij}, \quad x_{ij} \in \{x_{1j}, x_{2j}, x_{3j}, \dots, x_{HMSj}\}. \quad (2)$$

Si no se cumple, se hace una **inicialización aleatoria** del elemento dentro de un rango específico dado por el límite inferior  $l$  y superior  $u$  del problema:

$$x_{ij}^{new} = l + \text{random} \cdot (u - l). \quad (3)$$

Una vez obtenido el elemento, se realiza una **corrección de tono** (PA, Pitch Adjusting), esto, nuevamente mediante un aleatorio y una comparativa respecto a un umbral de corrección de tono (PAR, Pitch Adjustment Rate). Para controlar el ajuste, se usa un ancho de banda (BW, Bandwidth). Es decir:

$$\text{si random} < \text{PAR}: x_{ij}^{new} = x_{ij} \pm \text{random} \cdot \text{BW}. \quad (4)$$

Una vez calculado todos los elementos de  $\mathbf{x}^{new}$ , el siguiente paso es **actualizar la memoria armónica**, para ello, es necesario evaluar la aptitud del nuevo individuo y verificar si es mejor que el peor existente, de ser así, el anterior es reemplazado por el nuevo. El Algoritmo 1 ilustra todos los pasos necesarios.

#### 4. Representación para la selección de atributos y búsqueda armónica binaria

En este trabajo se utiliza la búsqueda armónica para la selección de atributos, para ello, es necesario definir primero la codificación que tendrán los arreglos de la memoria armónica. Una vez dada esta codificación, la búsqueda armónica tal cual el Algoritmo 1 necesitará ser modificada.

##### 4.1 Codificación

Considere un problema donde se cuenta con 6 atributos, cada uno de estos será marcado con un 1 cuando sea un atributo a ocupar, o 0 en caso contrario. Es decir, se tiene una

codificación binaria. Continuando con el ejemplo, una posible selección de atributos dentro de 6 atributos posibles es la siguiente:

donde se observa que, del conjunto de atributos original, solo los atributos  $F_2$ ,  $F_3$  y  $F_6$  serán ocupados.

## 4.2 Búsqueda armónica binaria

El Algoritmo 1, está pensado para problemas con números reales, por ello, es necesario realizarle modificaciones para adaptarlo a la codificación binaria. La variación más sencilla [6], consiste en, remover el paso de ajuste de tono, para quitarse la introducción del aleatorio real y el ancho de banda, y cambiar la inicialización aleatoria a una binaria, pues no es necesario obtener un punto entre los límites superior e inferior y solo se necesita seleccionar aleatoriamente un 1 o un 0. La operación de selección de aleatorio binario se denota como  $random(0|1)$ . La búsqueda armónica con estos ajustes se muestra en el Algoritmo 2.

### Algoritmo 1. Búsqueda Armónica

Require: HMS, HMCR, PAR, BW, NI (no. iteraciones)

Ensure: Todos los aleatorios dentro de (0,1)

```

for i = 1 : i <= HMS do
    Genera el vector aleatorio  $x_i$  de la memoria.
    Evalúa la aptitud del individuo  $f(x_i)$ .
end for
for t = 1 : t <= NI do
    for i = 1 : i <= HMS do
        for j = 1 : j <= m do
            if random < HMCR then           ▷ Memory Consideration
                 $x_{ij}^{new} = x_{ij}$ ,  $x_{ij} \in \{x_{1j}, x_{2j}, x_{3j}, \dots, x_{HMSj}\}$  ▷  $i$  es aleatorio
            if random < PAR then           ▷ Pitch Adjustment
                 $x_{ij}^{new} = x_{ij} \pm random \cdot BW$ 
            end if
            if  $x_{ij}^{new} < l$  then           ▷ Evita estar fuera de rango
                 $x_{ij}^{new} = l$ 
            end if
            if  $x_{ij}^{new} > u$  then           ▷ Evita estar fuera de rango
                 $x_{ij}^{new} = u$ 
            end if
        else
             $x_{ij}^{new} = l + random \cdot (u - l)$  ▷ Inicialización Aleatoria
        end if
    end for
end for
if  $f(x^{new}) < f(x^{worst})$  then           ▷ Actualización de la Memoria
     $x^{worst} = x^{new}$ 
end if
end for

```

## 5. Estructura del modelo

En la sección anterior, se introdujo la codificación, así como la modificación a la búsqueda armónica para poder trabajar con arreglos binarios, con esto, es posible utilizar el Algoritmo 2 tal cual se ilustra, sin embargo, hay un componente pendiente por definir, y este es la función de aptitud.

Esta función depende del problema que se quiera resolver, en el presente trabajo, se hace uso de bases de datos de índole médica, las cuales tienen como objetivo clasificar ciertos padecimientos, es por ello, que la función de aptitud debe ser un clasificador. El utilizado en este artículo es una Red Neuronal, la cual genera un clasificador por cada subconjunto de atributos contenidos en la memoria armónica.

Cada uno de los clasificadores, es entrenado solo con los atributos seleccionados, dando como valor de aptitud el error calculado mediante la entropía cruzada. La red neuronal, tiene un número de neuronas en la capa de entrada igual al número de atributos seleccionados, mientras las neuronas en la capa de salida dependen del problema, es decir, del número de clases posibles de cada base.

El ejemplo de una estructura, se puede apreciar en la Figura 1, el problema en cuestión cuenta con dos clases y en cada clasificador se mantiene constante, mientras el número de neuronas de entrada varía entre uno y seis dependiendo de los atributos seleccionados en color negro. Finalmente, cada clasificador arroja cierto error, que se intenta minimizar, considerado como la aptitud (costo).

### Algoritmo 2. Búsqueda Armónica Binaria

```

Require: HMS, HMCR, PAR, BW, NI (no. iteraciones)
Ensure: Todos los aleatorios dentro de (0,1)
for i = 1 : i <= HMS do
    Genera el vector aleatorio  $x_i$  de la memoria.
    Evalúa la aptitud del individuo  $f(x_i)$ .
end for
for t = 1 : t <= NI do
    for i = 1 : i <= HMS do
        for j = 1 : j <= m do
            if random < HMCR then          ▷ Memory Consideration
                 $x_{ij}^{new} = x_{ij}$ ,  $x_{ij} \in \{x_{1j}, x_{2j}, x_{3j}, \dots, x_{HMSj}\}$  ▷ i es aleatorio
            else
                 $x_{ij}^{new} = random(0|1)$           ▷ Inicialización Aleatoria
            end if
        end for
    end for
    if  $f(x^{new}) < f(x^{worst})$  then          ▷ Actualización de la Memoria
         $x^{worst} = x^{new}$ 
    end if
end for

```

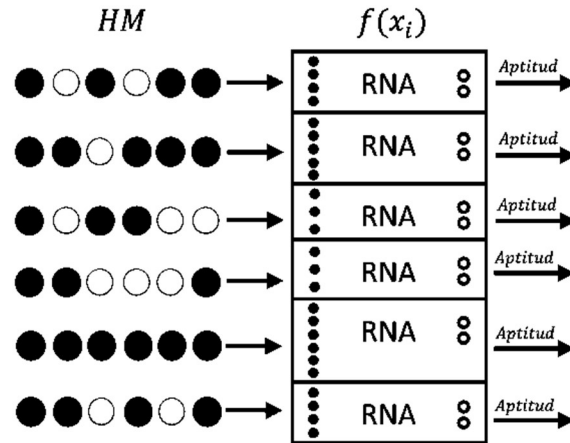


Fig. 1. Estructura del modelo para un problema de dos clases.

Tabla 2. Información de las bases de datos.

Base	Atributos	Instancias	Clases
Lung Cancer [7]	57	32	3
Hepatitis [8]	20	155	2
Dermatology [9]	35	366	6
Pima Indians Diabetes [10]	9	768	2

## 6. Experimentos

El lenguaje utilizado para la implementación, fue Julia, el cual, es un lenguaje compilado que permite tener velocidades de ejecución mayores a los lenguajes interpretados. El algoritmo de búsqueda binaria fue programado desde cero, mientras para la red neuronal, se usó la biblioteca FLUX, la cual nos permite crear redes neuronales de manera robusta y sencilla.

El algoritmo generado se denota como HS+ANN. El modelo se probó sobre cinco bases de datos distintas, donde el número de atributos, instancias (número de registros) y clases se da en la Tabla 2. Dentro de los atributos se incluye la clase a la que pertenece cada instancia. De cada base se tomó el 80% para el conjunto de entrenamiento y 20% para el conjunto de prueba.

Para el procesamiento de datos, todas las bases fueron normalizadas y los datos faltantes en alguna instancia fueron reemplazados por ceros, además, en todos los experimentos se incluye el vector de unos, es decir, el vector donde se ocupan todos los atributos excluyendo la clase. La Red Neuronal consiste de dos capas ocultas, con un número de neuronas por capa definido como:

$$\text{neuronas ocultas} = 2 \cdot N + 1, \tag{5}$$

**Tabla 3.** Resultados de los 100 experimentos por base de datos.

Variable	Media	Min	Med	Max
exentre	1	1	1	1
exprueb	0.5	0.22	0.55	0.66
noatrib	34	27	34	56

(a) Lung Cancer

Variable	Media	Min	Med	Max
exentre	1	1	1	1
exprueb	0.78	0.67	0.77	0.87
noatrib	15	10	15	19

(b) Hepatitis

Variable	Media	Min	Med	Max
exentre	1	1	1	1
exprueb	0.963	0.92	0.96	0.986
noatrib	28	22	29	33

(c) Dermatology

Variable	Media	Min	Med	Max
exentre	0.788	0.76	0.788	0.827
exprueb	0.713	0.649	0.72	0.766
noatrib	7.7	6	8	8

(d) Pima Indians Diabetes

donde N es el número de características del modelo a ser probado. Con función de activación relu en las capas ocultas y softmax en la capa de salida. Finalmente, la función de costo a minimizar es la entropía cruzada. El número de épocas se fijó en 150 y la tasa de aprendizaje en 0.01, ambos parámetros se mantienen constantes en todas las pruebas y bases de datos.

Con respecto a los parámetros de la búsqueda armónica, estos son los siguientes: HMS = 20, HMCR = .95, NI = 100 y una tolerancia de iteraciones sin mejora del 10%. Similar a los parámetros anteriores, se mantienen constantes en todas las pruebas y bases de datos. En cada experimento se determina el mejor clasificador de acuerdo con la entropía cruzada y el conjunto de atributos que lo generó se guarda como el mejor, así como la exactitud de entrenamiento y pruebas. Dicho experimento se repite un total de 100 veces para obtener los 100 mejores conjuntos de atributos. Para elegir el mejor de estos 100, se toma aquel que pondera la mejor exactitud tanto en el entrenamiento como en las pruebas usando la regla:

$$\text{exactitud} = \text{exactitud de entrenamiento} \cdot \text{exactitud de pruebas.} \tag{6}$$

Los resultados estadísticos de las 100 ejecuciones se muestran en la Tabla 3, donde se da: la media, mediana (med), mínimo (min) y máximo (max) para la exactitud del entrenamiento (exentre), exactitud de las pruebas (exprueb) y el número de atributos (noatrib). La información del mejor obtenido dentro de estos 100 se encuentra en Tabla 4, donde, los atributos originales corresponden al número total de atributos menos el atributo de la clase a la que pertenece cada instancia.

Atributos seleccionados, es el número de atributos utilizados por el mejor experimento y por ende el resultado final. Atributos reducidos es solo la cantidad en la que se redujo la dimensión del problema (originales - seleccionados). Los atributos elegidos para cada uno de los mejores, se encuentran en la Tabla 5.

Tanto la base Lung Cancer, Hepatitis y Dermatology, en los 100 experimentos, la exactitud en el entrenamiento es del 100%, esto se puede afirmar pues no es posible tener valores en la exactitud por arriba de 1, así, una media 1 efectivamente representa

**Table 4.** Información del mejor experimento para cada base de datos empleando exactitud

Variable	Lung Cancer	Hepatitis	Dermatology	Pima I. Diabetes
exentre	1	1	1	0.788
exprueb	0.66	0.87	0.986	0.766
atrib. originales (sin clase)	56	19	34	8
atrib. seleccionados	34	16	28	7
atrib. reducidos	21	3	6	1

**Table 5.** Características elegidas para el mejor experimento por base de datos.

Base	Características
<b>Lung Cancer</b>	<i>P1, P2, P3, P5, P6, P7, P8, P13, P14, P16, P17, P18, P19, P20, P21, P23, P24, P27, P28, P29, P30, P34, P38, P39, P40, P42, P43, P44, P46, P48, P50, P51, P54, P56</i>
<b>Hepatitis</b>	Age, Sex, Steroid, Antivirals, Fatigue, Malaise, Liverfirm, Spleenpalpable, Spiders, Ascites, Varices, Alkphosphate, Sgot, Albumin, Protime, Histology
<b>Dermatology</b>	erythema, definite_borders, itching, koebner_phenomenon, polygonal_papules, follicular_papules, oral_mucosal_involvement, knee_and_elbow_involvement, scalp_involvement, family_history, eosinophils_in_the_infiltrate, pnl_infiltrate, fibrosis_of_the_papillary_dermis, exocytosis, acanthosis, hyperkeratosis, parakeratosis, clubbing_of_the_rete_ridges, elongation_of_the_rete_ridges, thinning_of_the_suprapapillary_epidermis, focal_hypergranulosis, disappearance_of_the_granular_layer, vacuolisation_and_damage_of_basal_layer, spongiosis, sawtooth_appearance_of_retes, follicular_horn_plug, perifollicular_parakeratosis, band-like_infiltrate
<b>Pima Indians Diabetes</b>	Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, Age

el 100% en todos los experimentos. El número medio de atributos por base es de 34 para la primera, 15 para la segunda y 28 para la tercera, los cuales son un número de atributos similares al mejor experimento de cada base dado en la Tabla 4, donde el único con un valor diferente es el de Hepatitis.

Sin embargo, esto no ocurre para la exactitud en las pruebas para Lung Cancer y Hepatitis, en esta ocasión, la media es de 0.5 cuando el mejor tiene una exactitud de 0.66 y media 0.78 con valor del mejor 0.87 respectivamente, para el caso de Dermatology sí se cumple, donde 0.963 es cercano a 0.986. En todos los casos, el mejor experimento está por arriba de la media, esto refuerza lo bien conocido, que para este tipo de heurísticas el número de experimentos es importante, puesto que, cada ejecución depende de los aleatorios que genere la máquina, dando resultados distintos.

La base **Pima Indians Diabetes**, se mantiene coherente con sus medias y los resultados del mejor, sin embargo, es la clase que menos reducción de atributos tiene,



**Table 6.** Comparativa directa con [1] empleando exactitud.

Base	Caract.	Instancias	Clases	Atributos Seleccionados	exprueb
Lung Cancer	56/55	32/31	3/2	34/26	66/81.30
Hepatitis	19/18	155/153	2/2	16/9	87/80.01
Dermatology	34/32	366/364	6/6	28/23	98.6/97.54
Pima Indians Diabetes	8/8	768/765	2/2	7/5	76.6/89.40

**Table 7.** Comparativa directa con otros algoritmos incluidos en [1] empleando exactitud.

Base	Bagging with C4.5 method	Boosting with C4.5 method	ACO bagging	ACO boosting	GCSA with DCNN	HS with ANN
Lung Cancer	84.20	85.45	77.34	77.34	<b>88.23</b>	66.00
Hepatitis	74.50	74.28	79.30	77.80	80.09	<b>87.00</b>
Dermatology	96.20	96.50	97.55	97.55	97.12	<b>98.60</b>
Pima I. Diabetes	78.34	75.10	<b>89.45</b>	87.45	89.34	76.60

solamente disminuyendo en 1, si bien, Hepatitis también reduce poco comparado con su número total de variables, en su media representa un valor de 15, mientras la media para Pima Indians Diabetes es cercana a 8, que es prácticamente sin cambios, siendo solo el mejor el que logra la reducción de 1. En cualquiera de los casos, la búsqueda armónica permite la reducción de atributos en todas las bases presentadas.

En la Tabla 6, se muestra una comparativa con el trabajo previo directo dado en [1], en el cual, se utilizan las mismas bases. Cabe resaltar, que, a pesar de ser las mismas, existen algunas diferencias, muy posiblemente dadas por la época entre el presente artículo y el artículo de [1], además, en [1] no se especifica el “%” de datos ocupados para el entrenamiento y pruebas, por lo que una comparativa estricta no es posible, sin embargo, sirve como referencia para ubicar el presente trabajo con los realizados previamente.

La información se presenta separada por una “/”, donde, al lado izquierdo se ubica la información de este artículo, y a la derecha, la información en [1]. En todos los casos, el algoritmo presentado en [1] (GCSA+DCNN), resulta en una reducción mayor de atributos, sin embargo, en la exactitud, únicamente rebasa al algoritmo presentado en este artículo en dos experimentos, por lo tanto, una reducción mayor de atributos no garantiza una mayor exactitud. Finalmente, un comparativo de exactitud obtenida con diferentes algoritmos, también ofrecida por [1], se muestra en la tabla 7, donde se puede ubicar al HS+ANN junto a otros cinco algoritmos.

## 7. Conclusiones

Las metaheurísticas son de utilidad para la selección de atributos, dentro de esta gama de algoritmos, la búsqueda armónica puede ser utilizada mediante la variación

binaria. Los resultados fueron presentados en la sección 6, donde se aprecia la reducción obtenida, esto, siempre y cuando se conjunte con un buen algoritmo de clasificación, como lo son las Redes Neuronales, con las que, trabajando en conjunto, se logran buenos resultados en la exactitud del entrenamiento, variando un poco en el de pruebas.

Este último problema está ligado a los datos de cada base, por lo que un análisis mayor a cada una o variación del clasificador será necesario para obtener mejores resultados. Como trabajo futuro, se podría realizar un análisis a profundidad sobre la presencia de ciertos atributos dentro de los 100 experimentos realizados, para tener un panorama mejor de los atributos de mayor relevancia y los cuales deben de estar presentes en cualquier modelo.

Además, se puede trabajar en cada base por separado para lograr ajustar de mejor manera los parámetros de la red neuronal y evitar un sobreajuste, recordando que en todas las bases se mantuvo la misma estructura y parámetros. Cabe destacar, que como cualquier heurística requirió de un número alto de iteraciones para obtener los resultados mostrados. Pese a que las pruebas contenían el vector de unos, los modelos generados resultan con atributos reducidos logrando el objetivo principal de reducir atributos empleando búsqueda armónica.

## Referencias

1. Nagarajan, S. M., Muthukumaran, V., Murugesan, R., Joseph, R. B., Meram, M., Prathik, A.: Innovative feature selection and classification model for heart disease prediction. *Journal of Reliable Intelligent Environments*, vol. 8, no. 4, pp. 333–343 (2021) doi: 10.1007/s40860-021-00152-3
2. Rani, A. S. S., Rajalaxmi, R. R.: Unsupervised feature selection using binary bat algorithm. In: 2nd International Conference on Electronics and Communication Systems, pp. 451–456 (2015) doi: 10.1109/ecs.2015.7124945
3. Inbarani, H. H., Banu, P. K. N., Azar, A. T.: Feature selection using swarm-based relative reduct technique for fetal heart rate. *Neural Computing and Applications*, vol. 25, no. 3–4, pp. 793–806 (2014) doi: 10.1007/s00521-014-1552-x
4. Geem, Z. W.: *Music-inspired harmony search algorithm: theory and applications*. vol. 191, Springer (2009)
5. Geem, Z. W., Kim, J. H., Loganathan, G.: A new heuristic optimization algorithm: Harmony search. *Simulation*, vol. 76, no. 2, pp. 60–68 (2001) doi: 10.1177/003754970107600201
6. Kong, X., Gao, L., Ouyang, H., Li, S.: A simplified binary harmony search algorithm for large scale 0-1 knapsack problems. *Expert systems with applications*, vol. 42, no. 12, pp. 5337–5355 (2015) doi: 10.1016/j.eswa.2015.02.015
7. Hong, Z. Q., Yang, J. Y.: Lung Cancer. UCI Machine Learning Repository (1992) [archive.ics.uci.edu/ml/datasets/lung+cancer](http://archive.ics.uci.edu/ml/datasets/lung+cancer)
8. Hepatitis. UCI Machine Learning Repository (1988) [archive.ics.uci.edu/ml/datasets/hepatitis](http://archive.ics.uci.edu/ml/datasets/hepatitis)
9. Dermatology. Homepage (2022) [datahub.io/machine-learning/dermatology](https://datahub.io/machine-learning/dermatology)
10. Pima Indians diabetes database. UCI Machine Learning (2016) [www.kaggle.com/datasets/uciml/pima-indians-diabetes-database?resource=download](https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database?resource=download)