تحسين خوارزمية التجميع الـ DBSCAN بالاعتماد على الـEps والـ MinPts

إعداد: كيندا العيسى اشراف: د. زكريا زكريا

جامعة البعث، كلية العلوم، قسم الرياضيات

الملخص

نُقدم في هذا البحث تعديلاً لخوارزمية عنقدة البيانات الـDBSCAN، والتي تُعتبر من خوارزميات التجميع الأكثر استخداماً، ولكنها تُعاني من بعض المشاكل، حيث أن الخوارزمية لا تقوم بتعيين قيم الثوابتEps(نصف قطر الجوار) وMinPts(الكثافة) وإنما المستخدم يقوم بإدخاله عشوائياً ودون معرفة مسبقة بالبيانات، لذلك قد يُصبح عدد التكرارات لتنفيذ الخوارزمية كبير، وكذلك التحديد العشوائي لقيم الثوابت قد يؤدي لإعطاء نتائج مختلفة. قُمنا باقتراح خوارزمية معدلة قادرة على العمل بشكل آلي دون أي تدخل من المستخدم، حيث تعتمد الخوارزمية المعدلة على إيجاد قيمة مناسبة للثابت Eps، وللثابت MinPts آلياً بدلاً من إدخالها من قبل المستخدم.

كما نُقدم في هذا البحث مقارنة بين أداء خوارزمية الـDBSCAN الأصلية، وخوارزمية الـDBSCAN المعدلة، حيث استخدمنا عدة معايير لتقييم جودة عملية التجميع (V-measure, Rand Index, Silhouette)، فتمكنا من زيادة دقة التجميع في الخوارزمية المعدلة وتقليل نسبة الضجيج(noise) فيها، كما استخدمنا المعيارين Homogeneity وCompleteness لتقييم مدى تماسك وتكامل المجموعات التي تمت تجميعها.

الكلمات المفتاحية: التجميع، خوارزمية الـDBSCAN ، MinPts، Eps، Silhouette، Rand Index، V-measure.

Enhancing the DBSCAN clustering algorithm based on Epsilon and MinPts

Dr. Zakaria Zakaria Kinda Al-Issa

Albaath University, Faculty of Science, Mathematics Department.

ABSTRACT

In this paper, we introduce a modification of the DBSCAN data clustering algorithm, which is one of the most used clustering algorithms, but it has some problems, As the algorithm does not set the values of the Eps constants and the MinPts density, but the user enters it randomly and without prior knowledge of the data, so the number of iterations to implement the algorithm becomes large, As well as random determination of the values of constants may lead to different results. We proposed a modified algorithm that is able to work automatically without any user intervention. Where the modified algorithm relies on finding a suitable value for the constant Eps (the radius of the neighborhood) and the constant MinPts (density) automatically instead of entering it by the user.

We also present in this research a comparison between the performance of the original DBSCAN algorithm and the modified DBSCAN algorithm, Where we used several criteria to evaluate the quality of the clustering process (V-measure, Rand Index, Silhouette), So we were able to increase the assembly accuracy in the modified algorithm and reduce the noise ratio in it, We also used the criteria Homogeneity and Completeness to assess the coherence and complementarity of the groups that were assembled.

Keywords: Clustering, DBSCAN Algorithm, MinPts , Eps, Silhouette, Rand Index, V-measure.

١. مقدمة:

ظهر علم التنقيب بالبيانات (Data Mining) كتقنية للتحليل والبحث داخل كميات كبيرة من البيانات، وذلك باستخدام تقنيات وأدوات آلية للكشف عن العلاقات بين عناصر البيانات لم يتم كشفها سابقاً واستخراج معلومات مفيدة منها [1].

تُعتبر تجميع البيانات (Data Clustering) واحدة من أهم الأدوات المستخدمة في التنقيب بالبيانات، والتي يتم فيها تجميع البيانات المتشابهة في تجمعات غير معروفة العدد مسبقاً تدعى العناقيد. حيث أن التشابه بين كائنين ضمن تجمع معين أكبر من التشابه بين كائنين ضمن تجمعين مختلفين [1].

تُعتبر خوارزمية الـDBSCAN من أشهر الخوارزميات المستخدمة في تجميع البيانات إلا أنها تعاني من بعض المشاكل، فإن اختيار قيمة نصف قطر الجوار (Eps) له أهمية كبيرة، حيث ان الخوارزمية لا تقوم بتعيين قيمة نصف قطر الجوار (Eps) وانما المستخدم يقوم بإدخاله عشوائياً، وقد يؤدي اختيار نصف قطر كبير جداً إلى تجميع غير صحيح (دمج مجموعات مختلفة ومستقلة)، أو أنه لن يكون هناك سوى مجموعة واحدة تحتوي على جميع نقاط البيانات. وقد يؤدي اختيار نصف قطر صغير جداً إلى عدد كبير جدًا من المجموعات، ويمكن ان يكون لكل نقطة مجموعة خاصة بها، مما يؤدي إلى تقارب بطيء جداً للخوارزمية.

كما أن الخوارزمية لا تقوم بتعيين الكثافة (MinPts) وانما المستخدم يقوم بإدخاله بشكل عشوائي، لذلك قد يصبح عدد التكرارات لتنفيذ الخوارزمية كبير، وكذلك التحديد العشوائي لقيم الثوابت قد يؤدي لإعطاء نتائج مختلفة.

٢. دراسة مرجعية:

ظهرت العديد من خوارزميات تجميع البيانات كالـ K-Means والـ Fuzzy C-Means والـDBSCAN والـ Mean – Shift، ولكن مع هذا الكم الهائل من المعلومات كان لابد من ايجاد آليات لتحسين أداء وفاعلية هذه الخوارزميات، وجعلها تعمل بشكل أسرع ودقة عالية، حيث درست الكثير من الأبحاث لتطوير خوارزمية الـDBSCAN وحل مشاكلها.

قدم [2] طريقة لتحسين خوارزمية الـDBSCAN وذلك لتقليل الوقت اللازم لعمل الخوارزمية والوصول الى النتائج بشكل أسرع، وذلك عن طريق ايجاد الجيران لكل النقاط بالتوازي باستخدام شجرة الغطاء (Cover Tree)، لكنه يقوم بإدخال نصف القطر والحد الادنى من النقاط بشكل يدوي. قدم [3] تقنية جديدة تسمى خوارزمية التجزئة الحساسة للمنطقة (locality–sensitive–hashing)، حيث يعتمد مبدأ LSH على تجزئة نقاط البيانات الى كثافات متعددة لتقليل التعقيد الحسابي لخوارزمية الـDBSCAN، ولكن هنا نحتاج إلى استخدام أسلوب التعلم التجريبي لاكتشاف المعلمات المثلى لكل مجموعة بيانات على حدى (نصف القطر، الحد الأدنى من النقاط). قدم [4] خوارزمية تسمى GriDBSCAN، والتي تعمل على تحسين أداء الـDBSCAN باستخدام تقسيم الشبكة والدمج، حيث يقوم بمعالجة البيانات في الشبكة الصغيرة أولاً، ثم دمج النتائج المعالجة في الشبكة الكبيرة. قدم [5] إطار عمل هجين لخوارزمية الـDBSCAN مع المنطق الضبابي، حيث أن النسخة المحسّنة هي تهجين خوارزمية الـDBSCAN مع قواعد if–then الضبابية، ولكن الخوارزمية المحسنة تأخذ معلمتين إدخال. قدم [6] خوارزمية قادرة على التعامل مع تباين الكثافة داخل مجموعة معينة (cluster)، وذلك عن طريق حساب متوسط الكثافة بالإضافة إلى تباين مجموعة معينة لأي من الكائنات الأساسية (Core Objects)، حيث تعطي هذه الخوارزمية نتائج محسنة وذلك بعد تجريب عدة قيم لمعلمات الادخال. قدم [7] خوارزمية جديدة وذلك باستخدام الدالة الغاوسية لحساب كثافة كل نقطة وايجاد النقاط الأساسية، ولكن بالإضافة الى معلمتي الادخال لخوارزمية الـDBSCAN فأن الخوارزمية المقترحة تتطلب ادخال معلمة إضافية لحساب الدالة الغاوسية. نم تطبيق الخوارزمية المقترحة لإيجاد أفضل مسار للوصول لموقع معين على الخريطة. قدم [8] طريقة لتحديد نصف القطر بشكل آلي في خوارزمية الـDBSCAN بدلاً من تحديدها من قبل المستخدم بشكل عشوائي، لكنه لم يتمكن من حل مشكلة التحديد العشوائي لكثافة المجموعات (الحد الأدنى من النقاط في كل مجموعة)، كذلك تم تطبيق خوارزمية الانتشار العكسي لحساب المسافة الإقليدية والذي يحدد التشابه وعدم التشابه بين النقاط. قدم [9] خوارزمية جديدة باستخدام خوارزمية KNN والخوارزمية

الجينية وبالاعتماد على مصفوفة الجوار الأقرب تقوم الخوارزمية بتحديد عدة قيم لنصف القطر (Eps) والحد الادنى من النقاط (MinPts) بدلاً من تحديدها من قبل المستخدم بشكل عشوائي، لكن الخوارزمية المقترحة تكلف الكثير من الوقت لإيجاد مصفوفة المسافة وإيجاد عدة قيم للمعلمات واختيار الامثل. قدم [١٠] إطار عمل هجين لخوارزمية الـDBSCAN مع خوارزمية الـK–median وخوارزمية الجار الأقرب، تقوم الخوارزمية بتحديد نصف القطر (Eps) والحد الادنى من النقاط (MinPts) بدلاً من تحديدها من قبل المستخدم، ولكن الخوارزمية لا تعطي قيمة واحدة للمعلمات، وانما يتم إيجاد قائمة من القيم وتجريبها جميعها واختيار الأمثل منها.

٣. مشكلة البحث:

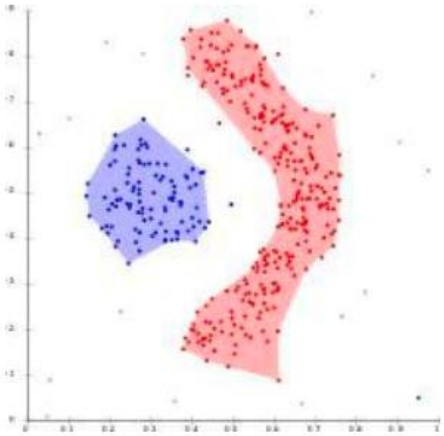
خوارزمية الـDBSCAN لا تقوم بتعيين قيمة نصف قطر الجوار (Eps) وانما المستخدم يقوم بإدخاله عشوائياً، كما أن الخوارزمية لا تقوم بتعيين الكثافة (MinPts) وانما المستخدم يقوم بإدخاله بشكل عشوائي، لذلك قد يصبح عدد نقاط الضجيج في الخوارزمية كبير وبذلك تصبح الخوارزمية اسوء في التجميع، وكذلك التحديد العشوائي لقيم الثوابت قد يؤدي لإعطاء نتائج مختلفة.

٤. الهدف من البحث:

نهدف من هذا البحث إلى تحسين خوارزمية الـDBSCAN وذلك لحل المشاكل التي تعاني منها هذه الخوارزمية. حيث تمكنا من إيجاد طريقة لجعل هذه الخوارزمية تعمل بشكل آلي ودون تدخل من المستخدم، من خلال وضع آلية لتعيين قيمة مناسبة للثابت Eps(نصف قطر الجوار)، وللثابت $MinPts$(الكثافة) ألياً بدلا من إدخالها من قبل المستخدم بشكل عشوائي، مما يساهم في تحسين أداء هذه الخوارزمية، وذلك بزيادة دقة التجميع وتقليل نسبة الضجيج(noise) فيها.

٥. خوارزمية الـ DBSCAN[11]:

تُعتبر خوارزمية الـDBSCAN من الخوارزميات الشائعة في تجميع البيانات، وتعتمد على المسافة والكثافة المتمثلة بعدد النقاط المحيطة بكل نقطة لتجميع النقاط المماثلة في مجموعات. تم تقديمها لأول مرة من قبل Martin Ester و Hans-Peter Kriegel و Jörg Sander و Xiaowei Xu في عام 1996 [12]. وهي اختصار لـ Density-Based Spatial Clustering of Applications with Noise. تأتي أهمية هذه الخوارزمية في قدرتها على تشكيل عناقيد عشوائية ولا يوجد شكل معين لشكل العناقيد، على عكس خوارزمية الـK-Means التي تشكل عناقيد كروية فقط، كما يوضح الشكل (1):



الشكل (1): يوضح تشكيل عناقيد عشوائية (كروية وغير كروية).

تعتمد خوارزمية الـDBSCAN على معيارين رئيسيين لتحديد المجموعات (Clusters):

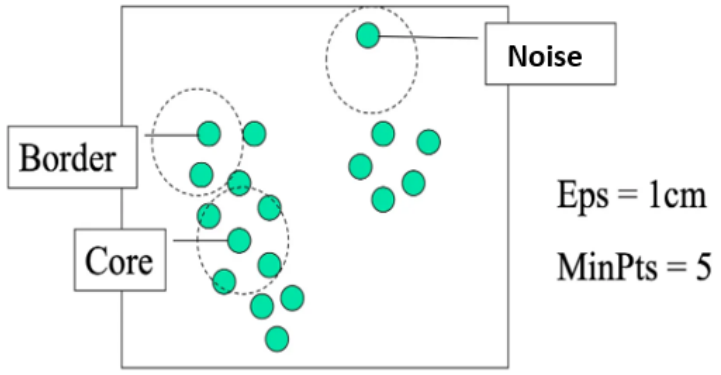
١. Epsilon: هو معيار يحدد مدى النقاط المتجاورة التي يجب تجميعها معًا، ويتم تحديد قيمتها يدوياً من قبل المستخدم.

٢. MinPts: هو معيار يحدد الحد الأدنى من النقاط التي من الممكن أن توضع داخل العنقود، ويتم تحديد قيمتها يدوياً من قبل المستخدم.

قبل البدء بشرح هذه الخوارزمية يجب التعرف على بعض الرموز والمفاهيم الأساسية[9]:

- $Eps(\varepsilon)$: نصف قطر الجوار.
- $MinPts$: الحد الأدنى من النقاط التي من الممكن أن توضع داخل العنقود.
- D: مجموعة البيانات.
- الجوار Eps للنقطة P: مجموعة النقاط التي تبعد عن P مسافة أصغر أو تساوي Eps يرمز لها بـ $N_\varepsilon(p)$، حيث $N_\varepsilon(p) = \{q \in D \mid dist(p,q) \leq \varepsilon\}$.
- الكثافة (Density): هو عدد النقاط التي توجد في جوار النقطة حيث $Density = |N_\varepsilon(p)|$.
- النقطة الأساسية (Core Point): هي كل نقطة تحقق $|N_\varepsilon(p)| \geq MinPts$.
- النقطة الحدودية (Border Point): هي كل نقطة تحقق $|N_\varepsilon(p)| < MinPts$، لكن تقع داخل جوار لنقطة أساسية.
- نقطة الضجيج (Noise Point): كل نقطة لا تكون نقطة حدودية أو أساسية هي نقطة ضجيج.

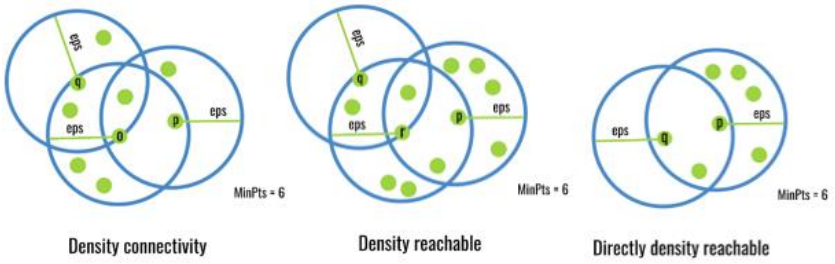
يوضح الشكل (2) مفهوم النقطة الأساسية والنقطة الحدودية ونقطة الضجيج.



الشكل (2): يوضح مفهوم النقطة الأساسية والنقطة الحدودية ونقطة الضجيج.

- قابلية الوصول المباشر للكثافة (Direct density reachability): نقول أن النقطة q يمكن الوصول إليها مباشرة من النقطة p إذا كانت q تقع داخل جوار p $(q\epsilon N_\varepsilon(p))$، و p هي نقطة أساسية $(|N_\varepsilon(p)| \geq MinPts)$.
- قابلية الوصول للكثافة (Density reachability): نقول إن النقطة q يمكن الوصول إليها من النقطة p إذا كانت هناك سلسلة من النقاط $p_1, p_2, \dots, p_n$ حيث $p_1 = p$ و $p_n = q$ ، وكانت p_{i+1} قابلة للوصول مباشرة من p_i حيث $1 \leq i \leq n$.
- الاتصال بالكثافة (Density connected): نقول أن النقطة q عبارة عن كثافة متصلة بالنقطة p إذا كانت هناك نقطة o بحيث يكون كل من p و q قابلين للوصول إلى الكثافة من o.

يوضح الشكل (3) مفهوم قابلية الوصول والاتصال بالكثافة.



الشكل (3): يوضح مفهوم قابلية الوصول والاتصال بالكثافة.

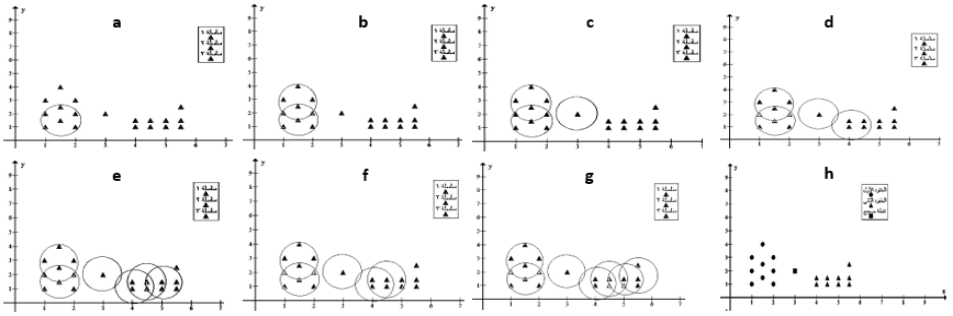
- العنقود C: نقول عن المجموعة الغير فارغة C انه عنقود إذا حقق الشرطين:

١. أياً كانت النقطتين p و q من D، إذا كانت p تنتمي إلى C وكانت النقطة q يمكن الوصول إليها من النقطة p، فان q تنتمي إلى C.

٢. أياً كانت النقطتين p و q من C، فان النقطة q عبارة عن كثافة متصلة بالنقطة p.

يوضح الشكل (4) طريقة عمل خوارزمية الـDBSCAN، حيث تعتمد الخوارزمية على تحديد كل نقطة إذا تمت زيارتها او لم تتم زيارتها. تبدأ الخوارزمية مع أي نقطة بيانات ولتكن p

مثلاً والتي لم تتم زيارتها سابقاً، وبعدها يُفحص جوار النقطة P ويُحسب عدد النقاط الموجودة في الجوار ليكن m مثلاً، إذا كان $m \geq MinPts$ (أي نقطة أساسية)، عندها يُشكل العنقود C_1 ويوضع فيه النقطة P وكل النقاط الموجودة في جوار النقطة P، بالإضافة لذلك توضع النقاط الموجودة في جوار النقطة P بمجموعة لتكن N_1، من أجل كل نقطة بيانات موجودة في المجموعة N_1، لتكن q مثلاً يتم فحص جوارها إذا كانت نقطة أساسية يضاف جوارها للمجموعة N_1، وتوضع النقطة q داخل العنقود C_1، وتحذف q من المجموعة N_1. وإلا لا تضاف نقاط الجوار إلى المجموعة N_1. ولا يوضع عليها إشارة أنها نقطة ضجيج بشكل مؤقت لأنها من الممكن أن تنتمي لعنقود آخر فيما بعد. وهكذا يتم فحص جميع نقاط البيانات الموجودة في المجموعة N_1 حتى تصبح المجموعة N_1 فارغة. تكرر العملية حتى يتم زيارة كل نقاط البيانات.



الشكل (4): طريقة عمل خوارزمية الـDBSCAN.

دخل الخوارزمية:

١. مجموعة البيانات D.
٢. نصف قطر الجوار Eps.
٣. الكثافة $MinPts$.

خرج الخوارزمية: عدد من العناقيد.

خطوات خوارزمية الـDBSCAN[13]:

يوضح الشكل (5) مخطط الانسياب الخاص بخوارزمية الـ[14] DBSCAN، حيث تتألف الخوارزمية من الخطوات الاتية:

١. ادخال قيمة لـ Eps و $MinPts$ بشكل عشوائي من قبل المستخدم.

٢. توضع علامة $unvisited$ لكل نقطة بيانات في مجموعة البيانات D. تبدأ الخوارزمية من أي نقطة عشوائياً ولتكن P مثلاً، يوضع عليها علامة $visited$.

٣. إيجاد $N_\varepsilon(p)$:

a) إذا كان $|N_\varepsilon(p)| \geq MinPts$ ، فإننا نعتبر P نقطة أساسية (Core Point)، عندها يُشكل عنقود C_1 ويوضع فيه النقطة P وكل النقاط الموجودة في جوار النقطة P، وتوضع النقاط الموجودة في جوار النقطة P بمجموعة لتكن N_1، ومن ثم يتم الانتقال للخطوة 4.

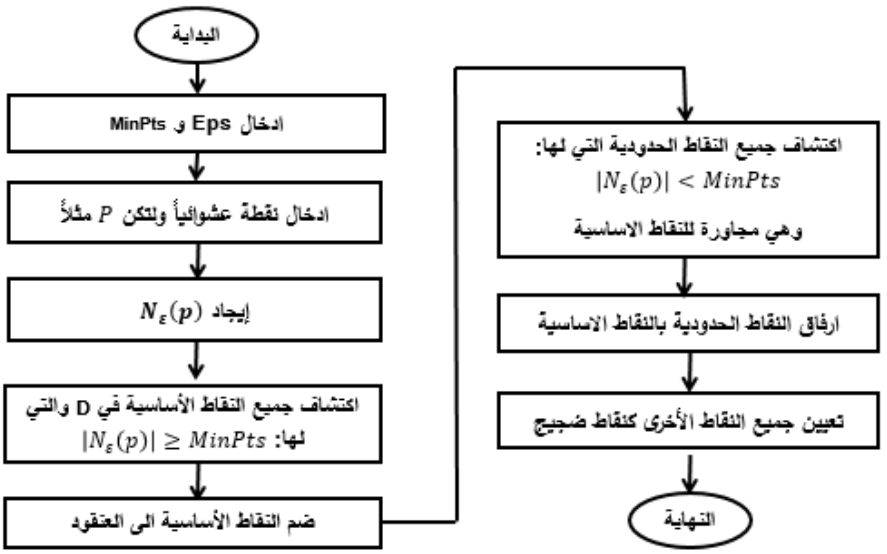
b) إذا كان $|N_\varepsilon(p)| < MinPts$ ، وأحد جيرانها هو نقطة أساسية، فإننا نعتبر P نقطة حدودية (Border Point)، ومن ثم يتم الانتقال للخطوة 6.

c) إذا لم تكن P نقطة حدودية أو أساسية، فإننا نعتبرها نقطة ضجيج، ومن ثم يتم الانتقال للخطوة 6.

٤. من أجل كل نقطة بيانات موجودة في المجموعة N_1، لتكن q مثلاً يتم فحص جوارها إذا كانت نقطة أساسية يضاف جوارها للمجموعة N_1، وتوضع النقطة q داخل العنقود C_1، وتحذف q من المجموعة N_1. وإلا لا تضاف نقاط الجوار إلى المجموعة N_1.

٥. تكرار الخطوة 4 حتى تصبح المجموعة N_1 فارغة.

٦. تكرار الخطوات من 1 إلى 6 حتى يتم زيارة جميع نقاط البيانات.



الشكل (5): مخطط انسياب خوارزمية الـDBSCAN.

مزايا خوارزمية الـDBSCAN[5]:

١. يمكنها اكتشاف مجموعات بأشكال وأحجام مختلفة.
٢. قوية بالتعامل مع القيم المتطرفة (الشاذة).
٣. لا تطلب تحديد عدد العناقيد مسبقاً.

المشاكل التي تعاني منها خوارزمية الـDBSCAN[5]:

١. إدخال قيم الثوابت Eps والكثافة $MinPts$ من قبل المستخدم بشكل عشوائي ودون معرفة مسبقة بالبيانات.
٢. تستطيع إيجاد عناقيد ذات احجام واشكال مختلفة، لكنها لا تستطيع إيجاد عناقيد ذات كثافات مختلفة.

٣. في حال وجود مجموعة كبيرة من البيانات، فإنه سيتطلب الكثير من الذاكرة لتخزين معلومات الكائن الأساسي.

٦. الخوارزمية المعدلة:

تتعامل هذه الخوارزمية مع m من الكائنات (النقاط) $N = \{x_1. x_2. \dots \dots . x_m\}$، وكل من هذه الكائنات يملك n خاصة (واصفة)، يمثل العدد n بعد الفضاء، حيث أن:

$$x_i = (x_{i1}. x_{i2}. \dots \dots . x_{in}) \quad : \quad 1 \leq i \leq m$$

حيث تقوم بتجميع هذه النقاط في عدة عناقيد. تعتمد هذه الخوارزمية على تحديد نصف القطر Eps الذي يجب أن تتواجد فيه الكائنات (النقاط)، وتحديد كثافة النقاط $MinPts$. إن عملية تحديد الثابت Eps تعتبر الخطوة الأساسية في بعض خوارزميات التجميع كخوارزمية الـMountain [15]، وخوارزمية الـ Entropy–Based Fuzzy Clustering [16]، ويتم تحديد هذا الثابت بطرق مختلفة بما يتناسب مع الخوارزمية المستخدمة، وقد قمنا بإيجاد صيغة جديدة لحساب الثابت Eps تتناسب مع خوارزميتنا المعدلة، حيث تمكنا من تحديد قيمة مناسبة لهذا الثابت، ومتوافقة مع الخوارزمية المعدلة، فلحساب هذا الثابت نقوم في البداية بإيجاد مصفوفة الابعاد D، وهي مصفوفة مربعة من البعد $(m \times m)$ وعناصرها قيم الأبعاد بين كل كائنين مختلفين، وذلك بالاعتماد على قانون البعد الإقليدي المعطى بالعلاقة (1):

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk)}{}^2} \qquad \dots (1)$$

حيث أن:

$$D_{ij} = \mathrm{d}(x_i . x_j) \quad : \quad 1 \le i \le m, 1 \le j \le m$$

بعد ذلك يتم حساب نصف القطر Eps وفق الخطوات التالية:

i. إيجاد أصغر بعد لكل نقطة من نقاط مجموعة البيانات:

$$min_i = min(D_{ij}) \ : (1 \le i \le m).(1 \le j \le m)$$

ii. إيجاد المتوسط الحسابي للأبعاد الأصغرية لنقاط مجموعة البيانات:

$$mean\ dist = mean(min_i) \ ; (1 \le i \le m)$$

iii. حساب قيمة الثابت Eps من العلاقة(2):

$$\mathrm{Eps} = \sqrt{\frac{mean\ dist}{log_2(N)}} \ \ldots \ (2)$$

حبث:N هو عدد الكائنات (النقاط) الموجودة في الفضاء.

بعد حساب الثابت Eps نقوم بحساب الكثافة $MinPts$ وفق الخطوات التالية:

i. إيجاد كثافة كل نقطة من نقاط مجموعة البيانات[2]:

$$P_{i,r} = \sum_{j=1}^{m} \chi(d_{i,j} \le r) : \quad 1 \le i \le m$$

حيث:

$$\chi(x) = \begin{cases} 1 : & x\ is\ true \\ 0 : & else \end{cases}$$

ومنه نجد أن كثافة النقطة هو العدد الكلي من الجيران لهذه النقطة.

ii. إيجاد المتوسط الحسابي لكثافات نقاط مجموعة البيانات:

$$mean\ densities = mean(P_i) \ ; (1 \le i \le m)$$

iii. حساب قيمة الثابت $MinPts$ من العلاقة(3):

$$MinPts = \sqrt{\frac{mean\ densities}{N}} \quad \dots \ (3)$$

حبث: N هو عدد الكائنات (النقاط) الموجودة في الفضاء.

بعد حساب قيم الثوابت Eps و $MinPts$ ، تصبح الخوارزمية قادرة على العمل بشكل آلي دون أي تدخل من المستخدم، وبهذا بعد تشغيل الخوارزمية نحصل على عدة تجمعات (عناقيد) في كل منها مجموعة من النقاط المتجاورة.

خطوات الخوارزمية المعدلة:

الدخل: مجموعة من الكائنات تحوي m عنصر $N = \{x_1.x_2.x_3.\dots\dots.x_m\}$ وكل من هذه الكائنات يملك n خاصة (يمثل العدد n بعد الفضاء) والتي يمكن تمثيلها في الفضاء على شكل نقاط حيث:

$$x_i = (x_{i1}.x_{i2}.\dots\dots.x_{in}) \quad : \quad 1 \le i \le m$$

الخرج: عدد k من العناقيد في كل عنقود عدد من النقاط (الكائنات).

الخطوات:

١. إيجاد قيمة الثابت Eps الذي يمثل نصف قطر الجوار من العلاقة (2).

٢. إيجاد قيمة الثابت $MinPts$ الذي يمثل كثافة النقاط من العلاقة (3).

٣. توضع علامة $unvisited$ لكل نقطة بيانات في مجموعة البيانات D. تبدأ الخوارزمية من أي نقطة عشوائياً ولتكن P مثلاً، يوضع عليها علامة $visited$.

٤. إيجاد $N_\varepsilon(p)$:

a) إذا كان $|N_\varepsilon(p)| \geq MinPts$، فإننا نعتبر P نقطة أساسية (Core Point)، عندها يُشكل عنقود C_1 ويوضع فيه النقطة P وكل النقاط الموجودة في جوار النقطة P، وتوضع النقاط الموجودة في جوار النقطة P بمجموعة لتكن N_1، ومن ثم يتم الانتقال للخطوة 5.

b) إذا كان $|N_\varepsilon(p)| < MinPts$، وأحد جيرانها هو نقطة أساسية، فإننا نعتبر P نقطة حدودية (Border Point)، ومن ثم يتم الانتقال للخطوة 7.

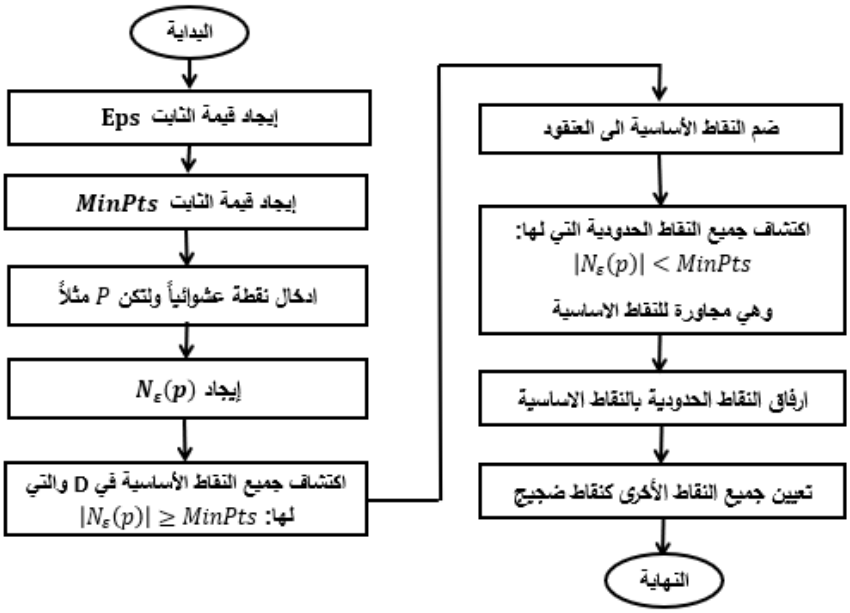
c) إذا لم تكن P نقطة حدودية أو أساسية، فإننا نعتبرها نقطة ضجيج، ومن ثم يتم الانتقال للخطوة 7.

٥. من أجل كل نقطة بيانات موجودة في المجموعة N_1، لتكن q مثلاً يتم فحص جوارها إذا كانت نقطة أساسية يضاف جوارها للمجموعة N_1، وتوضع النقطة q داخل العنقود C_1، وتحذف q من المجموعة N_1. وإلا لا تضاف نقاط الجوار إلى المجموعة N_1.

٦. تكرار الخطوة 5 حتى تصبح المجموعة N_1 فارغة.

٧. تكرار الخطوات من 1 إلى 7 حتى يتم زيارة جميع نقاط البيانات.

يوضح الشكل (6) مخطط الانسياب الخاص بخوارزمية الـDBSCAN المعدلة.



الشكل (6): مخطط انسياب خوارزمية الـDBSCAN المعدلة.

٧. النتائج والمناقشة:

1.7 مؤشرات التقييم:

استخدمنا يعض المقاييس لتقييم أداء الخوارزميتين الاصلية والمعدلة، وهي العدد المقدر لعدد المجموعات (number of clusters) والعدد المقدر لنقاط الضجيج (number of noise points) وعدة معايير لتقييم جودة عملية التجميع (V−measure, Rand Index, Silhouette) واستخدمنا المعيارين (Completeness, Homogeneity) لتقييم مدى تجانس وتكامل المجموعات التي تمت تجميعها.

مؤشر Silhouette (SIL)[17] :

يستخدم هذا المعيار لتقييم نتائج التجميع، حيث يعبر عن مدى تشابه كائن مع مجموعته الخاصة بالمقارنة مع المجموعات الأخرى. مؤشر الـSilhouette يعطى بالعلاقة (4):

$$SIL = \frac{1}{n}\sum_{x\in X}\frac{b(x)-a(x)}{max(a(x).b(x))} \quad ...(4)$$

حيث أن:

$$a(x) = \frac{1}{n_k - 1}\sum_{x_c\in C_k} d(x.x_c)$$

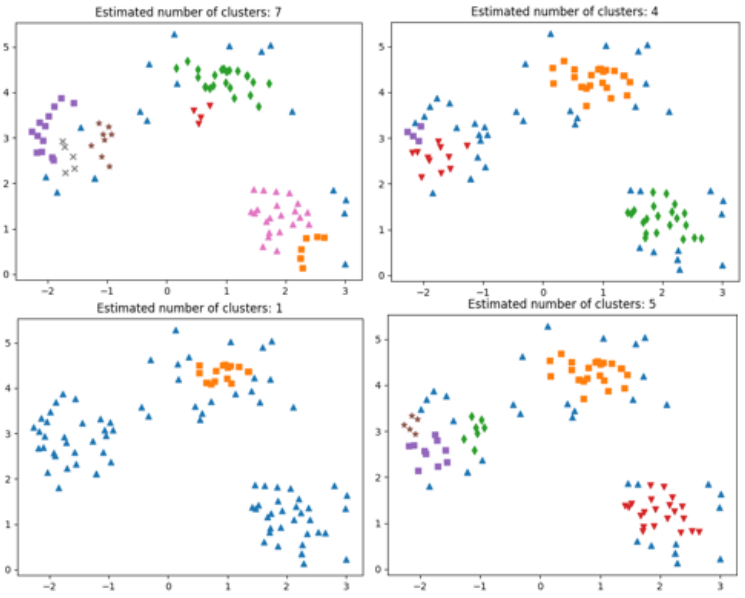
و

$$b(x) = \min\left(\frac{1}{n_l}\sum_{x_l\in C_l} d(x.x_l)\right)$$

$a(x)$ هو عبارة عن متوسط المسافات الإقليدية بين كل نقطة (كائن) ونقاط العنقود الأخرى.
$b(x)$ هو عبارة عن أصغر متوسط للمسافات الإقليدية بين كل نقطة (كائن) ونقاط العناقيد الأخرى.

2.7 مناقشة تنفيذ الخوارزمية المعدلة مع خوارزمية الـDBSCAN:

لقد قمنا بكتابة الكود البرمجي لخوارزمية الـDBSCAN الاصلية والمعدلة، وذلك باستخدام لغة البرمجة Python. قمنا بتطبيق الخوارزمية الاصلية والخوارزمية المعدلة على أربع مجموعات بيانات (blobs, moon, iris, digits) مستوردة من مكتبة الـ sklearn (من أشهر مكتبات البايثون المختصة بعلم البيانات وتعليم الآلة)، حيث تم تجربة الخوارزميتين عدة مرات على نفس مجموعة البيانات (blobs)، لنرى نتائج التنفيذ للخوارزميتين في كل مرة. يوضح الشكل (7) نتيجة تطبيق خوارزمية الـDBSCAN الاصلية للمجموعة السابقة عدة مرات على الحاسب، ويوضح الشكل (8) نتيجة تطبيق الخوارزمية المعدلة للمجموعة السابقة عدة مرات ايضاً على الحاسب.

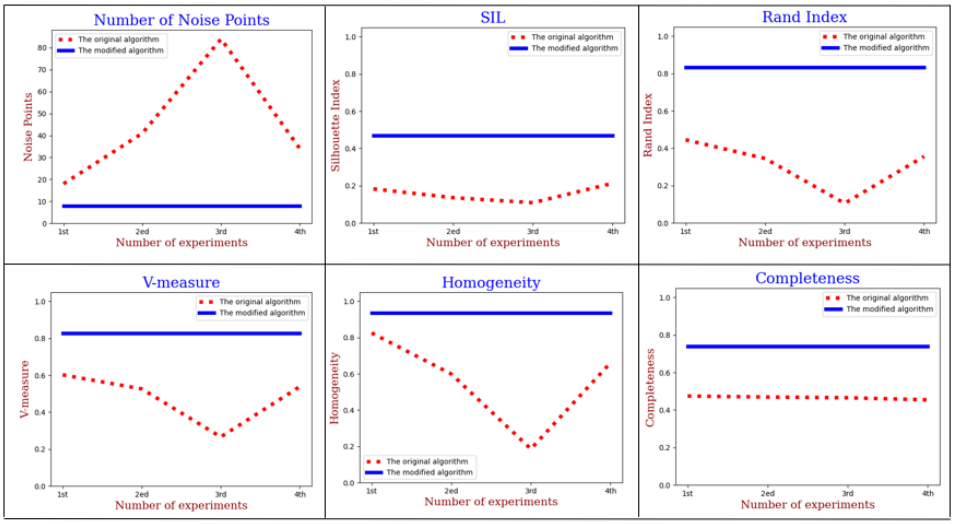


الشكل (7): يوضح نتيجة تطبيق الخوارزمية الاصلية على الحاسب.

الشكل (8): يوضح نتيجة تطبيق الخوارزمية المعدلة على الحاسب.

نلاحظ انه وبسبب الاختيار العشوائي لنصف القطر Eps والكثافة $MinPts$ من قبل المستخدم، فأن الخوارزمية الاصلية تقوم بتجميع البيانات بشكل مختلف في كل مرة، وينتج

عن هذه الاختيارات العشوائية نتائج مختلفة لعدد المجموعات وعدد نقاط الضجيج وللمعايير (V-measure, Rand Index, Silhouette, Completeness, Homogeneity) في كل مرة نطبق فيها الخوارزمية الاصلية، اما الخوارزمية المعدلة فتعطينا نتائج ثابتة لتجميع البيانات ونتائج أفضل في كل مرة ننفذ فيها الخوارزمية. يوضح الشكل (9) المخطط البياني لعدد نقاط الضجيج، وللمعايير (V-measure, Rand Index, Silhouette, Completeness, Homogeneity) وذلك بعد تطبيق الخوارزميتين عدة مرات على مجموعة البيانات (blobs).



الشكل (9): يوضح نتائج المعايير المختلفة بعد تطبيق الخوارزميتين عدة مرات.

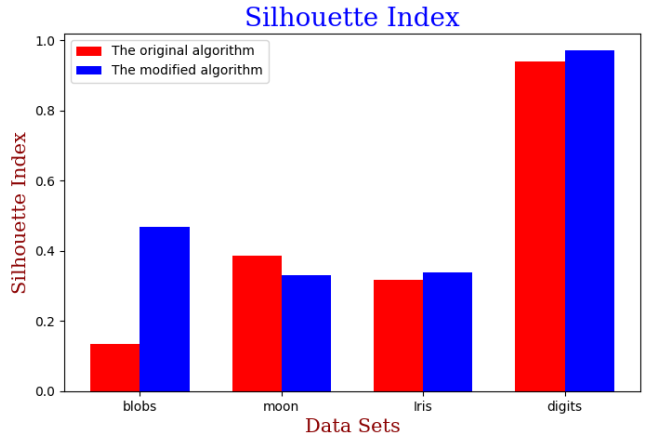
نقوم بتطبيق الخوارزمية الاصلية والخوارزمية المعدلة على مجموعات مختلفة من النقاط، حيث تم اختيار هذه المجموعات بحيث تظهر لدينا نتائج مختلفة، والنتائج تظهر في الجدول (1).

مقاييس التقييم / مجموعة البيانات		blobs	*moon*	*Iris*	digits
Silhouette	الخوارزمية الاصلية	0.136	0.386	0.319	0.941
	الخوارزمية المعدلة	0.469	0.331	0.338	0.971
Rand Index	الخوارزمية الاصلية	0.345	0.204	0.484	0.01
	الخوارزمية المعدلة	0.834	1.00	0.495	0.01
V–measure	الخوارزمية الاصلية	0.526	0.428	0.566	0.086
	الخوارزمية المعدلة	0.826	1.00	0.604	0.086
Homogeneity	الخوارزمية الاصلية	0.599	0.944	0.543	0.053
	الخوارزمية المعدلة	0.934	1.00	0.568	0.054
Completeness	الخوارزمية الاصلية	0.469	0.277	0.592	0.232
	الخوارزمية المعدلة	0.739	1.00	0.646	0.216
Noise Points	الخوارزمية الاصلية	41	14	12	138
	الخوارزمية المعدلة	8	0	9	71
Number of clusters	الخوارزمية الاصلية	4	11	3	2
	الخوارزمية المعدلة	4	2	3	3

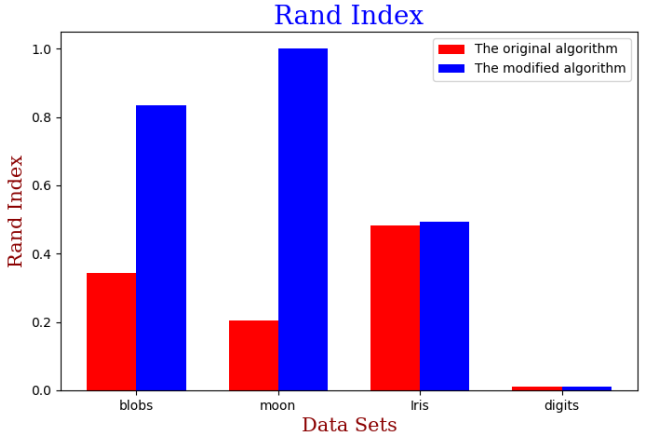
الجدول (1): نتيجة تطبيق الخوارزمية الاصلية والمعدلة مع بيانات مختلفة.

يوضح الشكل (10) المخطط البياني لمعيار الـSilhouette في الخوارزمية الاصلية والخوارزمية المعدلة، والشكل (11) يوضح المخطط البياني لمعيار الـRand Index ، والشكل (12) يوضح المخطط البياني لمعيار الـV–measure ، كما يوضح الشكل (13)

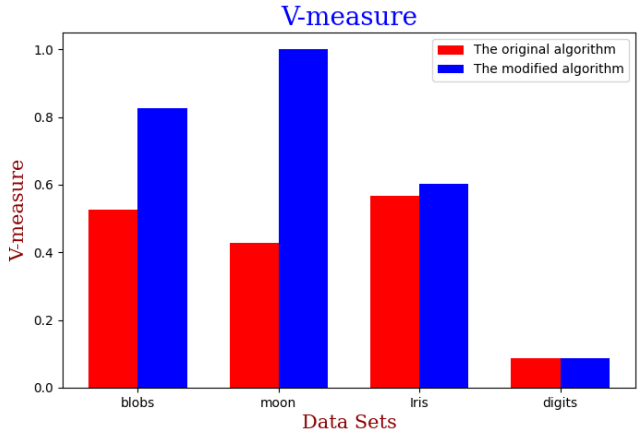
المخطط البياني لمعيار الـHomogeneity ، والشكل (14) يوضح المخطط البياني لمعيار الـCompleteness ، والشكل (15) يوضح المخطط البياني لعدد نقاط الضجيج في الخوارزمية الاصلية والخوارزمية المعدلة.



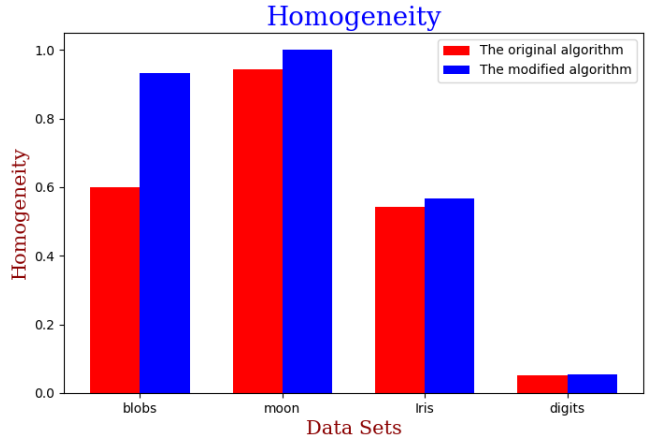
الشكل (10): المخطط البياني لمعيار الـSilhouette في الخوارزمية الاصلية والخوارزمية المعدلة.



الشكل (11): المخطط البياني لمعيار الـRand Index في الخوارزمية الاصلية والخوارزمية المعدلة.

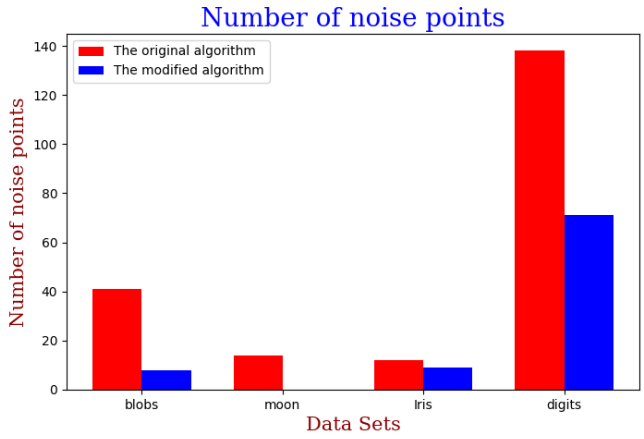


الشكل (12): المخطط البياني لمعيار الـV–measure في الخوارزمية الاصلية والخوارزمية المعدلة.



الشكل (13): المخطط البياني لمعيار الـHomogeneity في الخوارزمية الاصلية والخوارزمية المعدلة.

الشكل (14): المخطط البياني لمعيار الـCompleteness في الخوارزمية الاصلية والخوارزمية المعدلة.



الشكل (15): المخطط البياني لعدد نقاط الضجيج في الخوارزمية الاصلية والخوارزمية المعدلة.

مما سبق نلاحظ أن خوارزميتنا المعدلة والتي تحدد نصف القطر (Eps) والكثافة ($MinPts$) بشكل آلي، تزيد من دقة التجميع وتقلل نسبة الضجيج(noise)، حيث أننا استخدمنا عدة معايير لتقييم جودة عملية التجميع في خوارزميتنا المعدلة، كما استخدمنا المعيارين Homogeneity و Completeness لتقييم مدى تجانس وتكامل المجموعات التي تمت تجميعها.

٨. أهم النتائج التي تم الوصول إليها:

قدمنا في هذا البحث تعديلاً لخوارزمية الـDBSCAN، حيث قمنا باقتراح خوارزمية معدلة قادرة على العمل بشكل آلي دون أي تدخل من المستخدم، حيث تعتمد الخوارزمية المعدلة على إيجاد قيمة مناسبة للثابت Eps(نصف قطر الجوار)، وللثابت $MinPts$ (الكثافة) ألياً بدلا من إدخالها من قبل المستخدم.

وضحنا أيضا فعالية الخوارزمية المعدلة من خلال تطبيقها على مجموعات مختلفة من البيانات، حيث استخدمنا عدة معايير لتقييم جودة عملية التجميع، وتمكنا من تقليل عدد نقاط الضجيج وبذلك تصبح الخوارزمية أفضل في التجميع، وتمكنا ايضا من زيادة دقة التجميع في خوارزمية الـDBSCAN المعدلة.

٩. التوصيات:

بالإمكان العمل مستقبلاً لتعميم خوارزميتنا المعدلة على قواعد بيانات ضخمة جداً ($Big\ Data$) والاستفادة من هذا التعديل.

دمج الخوارزمية المعدلة مع تقنيات التعلم غير الخاضع للإشراف لتحسين نتائج التجميع في البيانات المعقدة والمتعددة الأبعاد.

المراجع

1. Han. J, & Kamber. M, 2006- Data Mining: Concepts and Techniques (2nd ed.). New Delhi: Morgan Kaufmann Publishers.
2. Li. S, 2020- An Improved DBSCAN Algorithm Based on the Neighbor Similarity and Fast Nearest Neighbor Query. Department of IT Development and Management, Huaqiao University, Xiamen, 361021 , China. February 2020, IEEE Access PP(99):1-1.
3. Wu. Y, Guo. J, & Zhang. X, 2007- A linear dbscan algorithm based on lsh. In 2007 International Conference on Machine Learning and Cybernetics, 2007, pp. 2608-2614.
4. Mahran. S, & Mahar. K, 2008- Using grid for accelerating density-based clustering. in 2008 8th IEEE International Conference on Computer and Information.Technology, 2008, pp. 35-40.
5. Raut. A, 2017- A Hybrid Framework using Fuzzy if-then rules for DBSCAN Algorithm. Department of Computer Science & Engineering, Amravati. India. pp. 933-942.
6. Ram. A, Jalal.S, Jalal. A, & Kumar. M, 2010- A density based algorithm for discovering density varied clusters in large spatial databases. Int. J. of Computer Applications 3.6 (2010): 1-4.
7. Luo. T, Zheng. X, Xu. G, Fu. K, & Ren. W, 2017- An Improved DBSCAN Algorithm to Detect Stops in Individual Trajectories. Int. J. Geo-Inf. 2017, 6, 63.
8. Lal. S, & Singh. V, 2022- Techniques to Enhance the Performance of DBSCAN Clustering Algorithm in Data Mining. International Journal for Research in Applied Science & Engineering Technology. Volume 10 Issue V May 2022.

9. Bin. Mu, Meng. D, & Shijin. Y, 2019- DBSCAN-KNN-GA: a multi Density-Level Parameter-Free clustering algorithm. IOP Conf. Series: Materials Science and Engineering 715 (2020) 012023.
10. Li. Y, Yang. Z, Jiao. SH, & Li. Y, 2022- Partition KMNN-DBSCAN Algorithm and Its Application in Extraction of Rail Damage Data. Hindawi Mathematical Problems in Engineering Volume 2022, Article ID 4699573.
11. Tran. T, Drab. K, & Daszykowski. M, 2013- Revised DBSCAN Algorithm to Cluster Data with Dense Adjacent Clusters. Chemometrics and Intelligent Laboratory Systems 120 (2013) 92–96.
12. Ester. M, Kriegel. P, Sander. J, & Xu. X, 1996- A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (pp. 226-231).
13. Shinde. H, & Sankhe. A, 2017- Comparison of Enhanced DBSCAN Algorithms: A Review. International Journal of Engineering Research & Technology. Volume 5, Issue 01.
14. Jain. D, Singh. M, & Sharma. A, 2017- Design and Implementation of DBSCAN Algorithm for Generating Clusters in Dense Region of Complex Datasets. International Journal Of Research In Electronics And Computer Engineering. VOL. 5 ISSUE 4.
15. Wang. J, & Liu. D, 2011- A Modified Mountain Clustering Algorithm based on Hill Valley Function. Hunan University of Science and Technology, Xiangtan, China, JOURNAL OF NETWORKS, VOL. 6, NO. 6, JUNE 2011.
16. BORA. D, & Gupta. A, 2014- A Comparative Study Between Fuzzy Clustering Algorithm and Hard Clustering Algorithm. International Journal of Computer Trends and Technology, India.
17. Starczewski. A, & Krzy˙zak. A, 2016- A Modification of the Silhouette Index for the Improvement of Cluster Validity Assessment. Springer International Publishing Switzerland.