

مجلة جامعة حمص

سلسلة العلوم الهندسية الميكانيكية والكهربائية
والمعلوماتية



مجلة علمية محكمة دورية

المجلد 48 . العدد 1

1447 هـ - 2026 م

الأستاذ الدكتور طارق حسام الدين رئيس جامعة حمص

المدير المسؤول عن المجلة

رئيس تحرير مجلة جامعة حمص للعلوم الإنسانية	أ. د. وليد حمادة
رئيس تحرير مجلة جامعة حمص للعلوم الطبية والهندسية والأساسية والتطبيقية	د.نعيمة عجيب

عضو هيئة التحرير	د.محمد فراس رمضان
عضو هيئة التحرير	د. مضر سعود
عضو هيئة التحرير	د. ممدوح عبارة
عضو هيئة التحرير	د. موفق تلاوي
عضو هيئة التحرير	د. طلال رزوق
عضو هيئة التحرير	د. أحمد الجاعور
عضو هيئة التحرير	د. الياس خلف
عضو هيئة التحرير	د. روعة الفقس
عضو هيئة التحرير	د. محمد الجاسم
عضو هيئة التحرير	د. خليل الحسن
عضو هيئة التحرير	د. هيثم حسن
عضو هيئة التحرير	د. أحمد حاج موسى

تهدف المجلة إلى نشر البحوث العلمية الأصيلة، ويمكن للراغبين في طلبها

الاتصال بالعنوان التالي:

رئيس تحرير مجلة جامعة حمص

سورية . حمص . جامعة حمص . الإدارة المركزية . ص . ب (77)

. هاتف / فاكس : ++ 963 31 2138071

. موقع الإنترنت : www.homs-univ.edu.sy

. البريد الإلكتروني : journal.homs-univ.edu.sy

ISSN: 1022-467X

شروط النشر في مجلة جامعة حمص

الأوراق المطلوبة:

- 2 نسخة ورقية من البحث بدون اسم الباحث / الكلية / الجامعة) + CD / word من البحث منسق حسب شروط المجلة.
 - طابع بحث علمي + طابع نقابة معلمين.
 - إذا كان الباحث طالب دراسات عليا:
يجب إرفاق قرار تسجيل الدكتوراه / ماجستير + كتاب من الدكتور المشرف بموافقة على النشر في المجلة.
 - إذا كان الباحث عضو هيئة تدريسية:
يجب إرفاق قرار المجلس المختص بإنجاز البحث أو قرار قسم بالموافقة على اعتماده حسب الحال.
 - إذا كان الباحث عضو هيئة تدريسية من خارج جامعة البعث :
يجب إحضار كتاب من عمادة كليته تثبت أنه عضو بالهيئة التدريسية و على رأس عمله حتى تاريخه.
 - إذا كان الباحث عضواً في الهيئة الفنية :
يجب إرفاق كتاب يحدد فيه مكان و زمان إجراء البحث ، وما يثبت صفته وأنه على رأس عمله.
 - يتم ترتيب البحث على النحو الآتي بالنسبة لكليات (العلوم الطبية والهندسية والأساسية والتطبيقية):
عنوان البحث .. ملخص عربي و إنكليزي (كلمات مفتاحية في نهاية الملخصين).
- 1- مقدمة
 - 2- هدف البحث
 - 3- مواد وطرق البحث
 - 4- النتائج ومناقشتها .
 - 5- الاستنتاجات والتوصيات .
 - 6- المراجع.

- يتم ترتيب البحث على النحو الآتي بالنسبة لكليات (الآداب - الاقتصاد - التربية - الحقوق - السياحة - التربية الموسيقية وجميع العلوم الإنسانية):
- عنوان البحث .. ملخص عربي و إنكليزي (كلمات مفتاحية في نهاية الملخصين).
- 1. مقدمة.
- 2. مشكلة البحث وأهميته والجديد فيه.
- 3. أهداف البحث و أسئلته.
- 4. فرضيات البحث و حدوده.
- 5. مصطلحات البحث و تعريفاته الإجرائية.
- 6. الإطار النظري و الدراسات السابقة.
- 7. منهج البحث و إجراءاته.
- 8. عرض البحث و المناقشة والتحليل
- 9. نتائج البحث.
- 10. مقترحات البحث إن وجدت.
- 11. قائمة المصادر والمراجع.
- 7- يجب اعتماد الإعدادات الآتية أثناء طباعة البحث على الكمبيوتر:
 - أ- قياس الورق 25×17.5 B5.
 - ب- هوامش الصفحة: أعلى 2.54- أسفل 2.54 - يمين 2.5- يسار 2.5 سم
 - ت- رأس الصفحة 1.6 / تذييل الصفحة 1.8
 - ث- نوع الخط وقياسه: العنوان . Monotype Koufi قياس 20
- كتابة النص Simplified Arabic قياس 13 عادي - العناوين الفرعية Simplified Arabic قياس 13 عريض.
- ج. يجب مراعاة أن يكون قياس الصور والجداول المدرجة في البحث لا يتعدى 12سم.
- 8- في حال عدم إجراء البحث وفقاً لما ورد أعلاه من إشارات فإن البحث سيهمل ولا يرد البحث إلى صاحبه.
- 9- تقديم أي بحث للنشر في المجلة يدل ضمناً على عدم نشره في أي مكان آخر، وفي حال قبول البحث للنشر في مجلة جامعة البعث يجب عدم نشره في أي مجلة أخرى.

10- الناشر غير مسؤول عن محتوى ما ينشر من مادة الموضوعات التي تنشر في المجلة
11- تكتب المراجع ضمن النص على الشكل التالي: [1] ثم رقم الصفحة ويفضل استخدام التهميش الإلكتروني المعمول به في نظام ورد WORD حيث يشير الرقم إلى رقم المرجع الوارد في قائمة المراجع.

تكتب جميع المراجع باللغة الانكليزية (الأحرف الرومانية) وفق التالي:

آ . إذا كان المرجع أجنبياً:

الكنية بالأحرف الكبيرة - الحرف الأول من الاسم تتبعه فاصلة - سنة النشر - وتتبعها معترضة (-) عنوان الكتاب ويوضع تحته خط وتتبعه نقطة - دار النشر وتتبعها فاصلة - الطبعة (ثانية . ثالثة) . بلد النشر وتتبعها فاصلة . عدد صفحات الكتاب وتتبعها نقطة .
وفيما يلي مثال على ذلك:

-MAVRODEANUS, R1986- **Flame Spectroscopy**. Willy, New York, 373p.

ب . إذا كان المرجع بحثاً منشوراً في مجلة باللغة الأجنبية:

— بعد الكنية والاسم وسنة النشر يضاف عنوان البحث وتتبعه فاصلة، اسم المجلد ويوضع تحته خط وتتبعه فاصلة — المجلد والعدد (كتابة مختزلة) وبعدها فاصلة — أرقام الصفحات الخاصة بالبحث ضمن المجلة.
مثال على ذلك:

BUSSE,E 1980 Organic Brain Diseases **Clinical Psychiatry News** , Vol. 4. 20 – 60

ج . إذا كان المرجع أو البحث منشوراً باللغة العربية فيجب تحويله إلى اللغة الإنكليزية و التقيد بالبنود (أ و ب) ويكتب في نهاية المراجع العربية: (المراجع In Arabic)

رسوم النشر في مجلة جامعة حمص

1. دفع رسم نشر (50000) ل.س أربعون ألف ليرة سورية عن كل بحث لكل باحث يريد نشره في مجلة جامعة البعث.
2. دفع رسم نشر (200000) ل.س مئة ألف ليرة سورية عن كل بحث للباحثين من الجامعة الخاصة والافتراضية .
3. دفع رسم نشر (200) مننًا دولار أمريكي فقط للباحثين من خارج القطر العربي السوري .
4. دفع مبلغ (15000) ل.س ستة آلاف ليرة سورية رسم موافقة على النشر من كافة الباحثين.

المحتوى

الصفحة	اسم الباحث	اسم البحث
36-11	د.م. أنس قرمو	دراسة أثر اتجاه الفجوات في الجدران المنشأة بالطباعة ثلاثية الأبعاد على أدائها الحراري
66-37	م. علي سليمان د. عبد الكريم السالم	تحليل صور الرنين المغناطيسي لمرض الزهايمر باستخدام تحسين التباين والشبكات العصبية
98-67	م. خالد قندقجي د. ماهر عباس د. أسماء شعاع	التنبؤ بحركة المرور ضمن المدن الذكية باستخدام خوارزميات التجميع والتعلم العميق
137-99	م. ميساء الضاهر د. ماهر عباس د. محسن عبود	مراجعة لموازنة حمل الخوادم في الشبكات المعرفة برمجياً

دراسة أثر اتجاه الفجوات في الجدران المنشأة بالطباعة ثلاثية الأبعاد

على أدائها الحراري

د.م. أنس قرمو

الملخص

في ظل الازدياد العالمي في الطلب على المباني المستدامة التي تتمتع بكفاءة طاقة عالية، تزايدت أهمية الدراسات المرتبطة بالأداء الحراري للجدران، فشهد قطاع البناء في السنوات الأخيرة تحولاً ملحوظاً، مع تبني تقنيات حديثة في طرق انشاء الأبنية من جهة وطرق وأدوات البحث من جهة أخرى، فمن ناحية انشاء الأبنية برزت الطباعة ثلاثية الأبعاد للجدران بوصفها إحدى أكثر التقنيات الواعدة في تطوير النظم الإنشائية التقليدية من حيث الكفاءة الزمنية، والدقة الهندسية، وتقليل الهدر في المواد. أما من ناحية التطور البحثي فقد برزت الطرق العددية والنمذجة في إجراء الأبحاث، كاستخدام برامج نمذجة حديثة لتلك الأغراض.

في هذا البحث تم إجراء دراسة حرارية لخمسة نماذج جدران يحتوي اثنان منها على فجوات هوائية شاقولية شكل فجوات أحدها بيضوي وشكل فجوات الآخر مربع، والنموذجتان الآخران يحتويان على فجوات هوائية افقية شكل فجوات أحدها بيضوي وشكل فجوات الآخر مربع، والنموذج الأخير مصمت لا يحتوي على فجوات كنموذج مرجعي، وتمت الدراسة وفق الشروط التصميمية الحرارية لمدينة حمص. عند تصميم النماذج تم الأخذ بعين الاعتبار نسبة فراغات ثابتة لجميع النماذج المفرغة بلغت 20% . تم إجراء الدراسة بطريقة النمذجة العددية على مساحة 1 m^2 باستخدام برنامج نمذجة حاسوبي هو ANSYS وقد تم تصميم النماذج المدروسة باستخدام برنامج SOLID WORKS ثم تصديرها لبرنامج النمذجة. بعد الانتهاء من النمذجة العددية، تم إجراء مقارنة بين كل نموذجين لهما نفس شكل الفجوات ويختلفان فقط باتجاه الفجوات لمعرفة تأثير هذا الاختلاف. بينت النتائج أن الاتجاه الأفقي للفجوات هو الأفضل، فقد حقق النموذج ذو الفجوات البيضوية الأفقية وفاقياً بلغ 12.66% عن النموذج ذو الفجوات البيضوية الشاقولية، وحقق النموذج ذو الفجوات المربعة الأفقية وفاقياً قدره 13.03% عن النموذج ذو الفجوات المربعة الشاقولية، وبذلك كان الاتجاه الأفقي للفجوات هو الاتجاه الأفضل والذي يحقق أداءً حرارياً أفضل في كل من الشكلين المدروسين لمقاطع الفجوات.

الكلمات المفتاحية: الجدران المطبوعة ثلاثية الأبعاد- فجوات الجدران الافقية والشاقولية - النمذجة الحرارية.

Studying the effect of gap orientation in 3D printed walls on their thermal performance

Dr. Anas Qarmo

Abstract

With the increasing global demand for sustainable, energy-efficient buildings, the importance of studies related to the thermal performance of walls has grown. The construction sector has witnessed a remarkable transformation in recent years, adopting modern technologies in both building construction methods and research tools. In terms of construction, 3D printing of walls has emerged as one of the most promising technologies for improving traditional construction systems in terms of time efficiency, engineering precision, and material waste reduction. On the research front, numerical methods and modeling have become prominent, utilizing modern modeling software for these purposes.

This research conducted a thermal study of five wall models. Two of these models contained vertical air gaps (one oval and one square), while the other two contained horizontal air gaps (one oval and one square). The final model was solid and served as a reference model. The study was conducted according to the thermal design requirements of Homs city. When designing the models, a fixed void ratio of 20% was considered for all hollow models. The study was conducted using numerical modeling on a 1 m² area with the ANSYS computer modeling software. The studied models were designed using SOLID WORKS software and then exported to the modeling software. After the numerical modeling was completed, a comparison was made between each pair of models with the same void shape but differing only in the void orientation to determine the effect of this difference. The results showed that the horizontal void orientation was the best. The model with horizontal oval voids achieved an energy saving of 12.66% compared to the model with vertical oval voids, and the model with horizontal square voids achieved an energy saving of 13.03% compared to the model with vertical square voids. Thus, the horizontal void orientation

was the best orientation, achieving better thermal performance in both studied void cross-section shapes.

Key Words: 3D printed walls - horizontal and vertical wall gaps - thermal modeling.

1. مقدمة:

شهدت العقود الأخيرة تطوراً متسارعاً في تقنيات البناء، مدفوعة بالحاجة المتنامية لإيجاد حلول إنشائية تجمع بين الكفاءة الاقتصادية، والجودة الهندسية العالية، والمتطلبات البيئية المتعلقة بالاستدامة وترشيد استهلاك الطاقة. وفي هذا السياق برزت تقنية الطباعة ثلاثية الأبعاد في البناء (D Printing Construction³) بوصفها إحدى أكثر التقنيات ثورية، وقدرتها على إعادة صياغة تقانات قطاع البناء، إذ تمكّنت هذه التقنية من تجاوز القيود التقليدية لعمليات البناء، سواءً من حيث سرعة الإنجاز أو التحكم بالشكل الهندسي أو حتى إمكانية إدخال فراغات وهياكل داخلية معقدة لا يمكن الحصول عليها بالطرق التقليدية، إلا بكلفة عالية وتعقيد كبير [1]. تتميز الطباعة ثلاثية الأبعاد للمباني بقدرتها على بثق المواد وفق مسارات محددة خاضعة للتحكم الرقمي الدقيق، مما يتيح بناء جدران وأجزاء إنشائية ذات مقاطع داخلية مبتكرة، تتضمن فجوات وهياكل خلوية محسوبة هندسياً. وقد فتح ذلك المجال أمام الباحثين لتقييم تأثير الشكل الهندسي للفجوات واتجاهها وحجومها وتوزيعها على الخصائص الميكانيكية والحرارية للعناصر المبنية بهذه التقنية. يمكن للتغيرات الصغيرة في اتجاه الفراغات أو شكلها أن تؤثر في مقاومة انتقال الحرارة، والموصالية الحرارية المكافئة، والتدفق الحراري عبر الجدار، وبالتالي في أداء الأبنية من حيث استهلاك الطاقة.

تزايدت أهمية الدراسات المرتبطة بالأداء الحراري للجدران في ظل الارتفاع العالمي في الطلب على مبانٍ مستدامة تتمتع بكفاءة طاقة عالية. إذ تشير تقارير الوكالة الدولية للطاقة (IEA) إلى أن المباني تمثل نحو 30-40% من استهلاك الطاقة العالمي، وأن تحسين الغلاف الحراري للمبنى هو أحد أكثر الطرق فعالية لخفض استهلاك الطاقة وتقليل الانبعاثات الحرارية. ومن هنا تأتي الحاجة إلى تقييم الأثر الحراري للفراغات الداخلية في الجدران المبنية بالطباعة ثلاثية الأبعاد، خصوصاً أن هذه الجدران ليست مصممة عادة، بل تحتوي على مسارات داخلية يتم تشكيلها نتيجة عملية الطباعة طبقة فوق طبقة، ويمكن إعادة تشكيلها هندسياً [2].

ومن بين أهم المتغيرات الهندسية التي ثبت أن لها تأثيراً مباشراً في أداء الجدران الحراري اتجاه الفجوات الداخلية. فبينما تعتمد الجدران التقليدية غالباً على مقاطع مصممة أو فجوات شاقولية محدودة، توفر الطباعة ثلاثية الأبعاد إمكانية بناء جدران ذات فجوات شاقولية أو أفقية ضمن نفس الحجم الكلي للفراغات. وهذا يخلق تباينات كبيرة في مسارات انتقال الحرارة، وفي مقاومة الجدار للتوصيل الحراري، وفي ظاهرة الحمل الحراري داخل الفجوات.

وتبيّن الدراسات السابقة أن الفجوات الأفقية يمكن أن تسهم في تحسين العزل الحراري بسبب إعاقة انتقال الحرارة بالحمل الحراري وتقليل انتقال الحرارة بالتوصيل، بينما يمكن أن تؤدي الفجوات الشاقولية إلى تغيرات في الحمل الحراري الطبيعي داخل الفراغ، مما يزيد أو يقلل التوصيل الحراري تبعاً لشكل الفجوة وأبعادها وتدرج درجات الحرارة في الجدار. وعلى الرغم من وجود أبحاث عديدة درست تأثير شكل الفجوات الهندسية على الأداء الحراري، فإن الأبحاث التي تركز على اتجاه الفجوات مع ثبات حجمها ومساحتها لا تزال محدودة إلى حد كبير، وخاصة في سياق الطباعة ثلاثية الأبعاد التي تتميز بترتيبات هندسية مختلفة جذرياً عن الجدران التقليدية.

وبما أن الطباعة ثلاثية الأبعاد تعتمد أساساً على بثق المادة على شكل طبقات، فإن اتجاه الفجوات سواء كان موافقاً لاتجاه البثق أو عمودياً عليه يمكن أن يؤثر أيضاً في الترابط الطبقي وخصائص سطح الفجوة، مما ينعكس بشكل غير مباشر على قدرة الجدار على تمرير أو منع انتقال الحرارة. كما أن شكل الفجوة واتجاهها قد يغير ديناميكية الهواء المحبوس داخل الجدار، وبالتالي يؤثر في الحمل الحراري الداخلي، الذي يمثل جزءاً أساسياً من مقاومة انتقال الحرارة في الجدران ذات الفراغات واسعة المساحة. [3]

2. أهمية البحث وأهدافه:

يهدف هذا البحث إلى تقديم مقارنة شاملة بين نموذجين من الجدران المبنية بتقنية الطباعة ثلاثية الأبعاد تعتمد على دراسة عددية، أحدهما يحتوي على فجوات شاقولية، والآخر يحتوي على فجوات أفقية، مع الحفاظ على ثبات بعض الثوابت الأساسية في كلا النموذجين.

وترتكز الدراسة على استخدام النمذجة العددية لمحاكاة انتقال الحرارة، عبر تحليل التوزيع الحراري والتدفق الحراري ضمن الجدارين باستخدام أحد برامج النمذجة الحرارية المعتمدة هو برنامج ANSYS ، وذلك بهدف تقييم أثر اتجاه الفجوات على الخصائص الحرارية والطاقية. وتسعى هذه الدراسة للإجابة على تساؤلات جوهرية تتعلق بأداء الجدران في الأبنية المستقبلية التي تعتمد تقنية الطباعة ثلاثية الأبعاد، ومنها:

- كيف يؤثر اتجاه الفجوات في آلية انتقال الحرارة داخل الجدران المطبوعة؟
 - هل يؤدي الاتجاه الأفقي للفجوات إلى زيادة أو نقصان الحمل الحراري داخل الفراغ مقارنة بالاتجاه الشاقولي؟
 - هل تتغير قيمة التدفق الحراري الكلي عند ثبات الحجم الهندسي للفراغات؟
 - أي الاتجاهين يحقق أفضل مقاومة حرارية وبالتالي أفضل كفاءة طاقية للجدار؟
- كما يُتوقع أن تساعد نتائج هذا البحث في وضع أسس تصميم حراري أمثل للجدران المطبوعة بالطباعة ثلاثية الأبعاد، خاصة في المناطق ذات الظروف المناخية المختلفة، حيث يمكن توجيه الفجوات بطريقة تحقق أفضل أداء حراري بما يتناسب مع البيئة المحيطة. إن أهمية هذا البحث لا تكمن فقط في كونه يدرس جانباً لم يحظَ ببحث كافٍ في الأدبيات العلمية الحالية، بل في كونه يساهم في توجيه تصميم الجدران المستقبلية المنشأة بالطباعة ثلاثية الأبعاد نحو مستويات أعلى من الكفاءة الطاقية والاستدامة، وهي إحدى الركائز الأساسية التي تعتمد عليها الهندسة الحديثة في مواجهة التحديات البيئية العالمية.

3. الدراسات المرجعية:

في ظل تقدم وتطور أدوات البحث العلمي قدمت الأدبيات العلمية العديد من الأبحاث التي تناولت الأداء الحراري للجدران الاسمنتية المطبوعة ثلاثية الأبعاد، وتتنوع طرق ووسائل تلك الدراسات، فقام البعض بالاعتماد على الطرق التجريبية لإجراء أبحاثه، في حين اعتمد البعض الآخر على الطرق العددية في إجراء أبحاثه، فقاموا بعمل دراساتهم باستخدام برامج

نمذجة مختصة. وعلى الرغم من وجود عدد من تلك الدراسات إلا أنه يبقى هنالك افتقار لأبحاث تغطي الجانب الحراري بشكل كافي. بالنظر إلى الدراسات المرجعية نجد:

درس مارايس وزملاءها في العام 2021 تحسين الأداء الحراري للجدران الاسمنتية المطبوعة ثلاثية الأبعاد والمطبوعة باستخدام اسمنت رغوي خفيف الوزن، حيث عمل الباحثون على مقارنة حالات تحتوي عدد معين من التجاويف مع حالات تحتوي على عدد تجاويف أكبر مع المحافظة على نسبة الفراغات ضمن النماذج، كما عملوا على دراسة تأثير حجم التجاويف على الأداء الحراري للجدران.

أظهرت الدراسة أن تقليل عرض التجاويف يؤدي إلى تحسين العزل الحراري. وتميز مقطع جداري من الخرسانة الرغوية خفيفة الوزن بأربعة تجاويف بعرض 20 mm بأداء حراري أفضل من جدار صلب من الخرسانة الرغوية خفيفة الوزن بنفس أبعاد المقطع الخارجي. كما أن زيادة عدد التجاويف ذات العرض الأصغر يُحسّن الأداء الحراري، وقد تبين أن نسبة الفراغات إلى المساحة الكلية التي تتجاوز 50% تُحقق أداءً حراريًا أقل. [4]

درس الزهراني ورفاقه في عام 2022 تحسين الأداء الحراري للجدران المطبوعة بطابعة ثلاثية الأبعاد وذلك عن طريق التحكم بمادة حشو تلك الجدران، حيث اقترح الباحث عدة مواد حشو مقترحة، وتم اجراء الدراسة الحرارية لجميع النماذج المقترحة، وتمت الدراسة بشكل عددي باستخدام برنامج نمذجة حاسوبي، وكان الهدف المنشود إيجاد التصميم ذي أداء حراري مُحسّن لتقليل استهلاك الطاقة في المباني دون تكبد تكاليف إضافية للمواد. وبناءً على ذلك، تم تطوير نموذج حسابي لمحاكاة السلوك الحراري لمختلف النماذج المقترحة التي يمكن استخدامها في جدران الخرسانة المطبوعة ثلاثية الأبعاد. وقد تم الحفاظ على تركيبة الخرسانة، ونسبة الخرسانة إلى الفراغات ثابتة للتركيز على تأثير هيكل الحشو (التغيرات الهندسية). تمت مقارنة الأداء الحراري وإمكانية توفير الطاقة لجدران الخرسانة المطبوعة ثلاثية الأبعاد مع مواد البناء التقليدية،

بما في ذلك الطوب الطيني والخرساني. تُظهر النتائج أن تغيير بنية الحشو لجدران العزل الحراري ثلاثي الأبعاد يؤثر على موصليتها الحرارية، وبالتالي على الأداء الحراري للمبنى. وقد وُجد أن الموصلية الحرارية لبنى الحشو المدروسة تتراوح بين $0.17 - 0.122 \text{ W/m.k}$ ، وفي حال نجاح هذه البنى في استبدال مواد البناء التقليدية، فإن الحد الأدنى للتوفير السنوي في تكلفة الطاقة سيبلغ حوالي دولار واحد لكل متر مربع . لذا، يُعد اختيار بنية الحشو المناسبة أمراً بالغ الأهمية لخفض استهلاك الطاقة في المبنى. [5]

في العام 2023 درس رافولا وفريقه تحسين الأداء الحراري لجدران الخرسانة المطبوعة ثلاثية الأبعاد (3DPC) باستخدام مواد عزل مختلفة وسماكات متفاوتة لطبقات العزل. وتمت هذه الدراسة باستخدام النمذجة العددية، حيث طُوّرت نماذج حرارية باستخدام برنامج THERM 7.8، وجرى التحقق من صحتها بالرجوع إلى الدراسات المنشورة وحالات الاختبار الحراري وفقاً لمعيار ISO 10211 . تم تطوير 42 نموذجاً عددياً، مع تغيير عرض الجدار 100 mm و 200 mm ، ونسب العزل مختلفة، ومواد العزل (الصوف المعدني، والصوف الصخري، والصوف الزجاجي)، وأظهرت النتائج أن زيادة نسبة العزل تؤدي إلى انخفاض قيمة معامل انتقال الحرارة. تم تطوير معادلات تصميم جديدة لرصد تأثير نسب العزل، وقد تتبأت هذه المعادلات بدقة بقيم معامل انتقال الحرارة. تُسهم هذه الدراسة في فهم الأداء الحراري لجدران (3DPC) مع خيارات عزل مختلفة، ويمكن للمعادلات المقترحة أن تساعد في تصميم مبانٍ موفرة للطاقة باستخدام (3DPC). [6]

4. طرائق البحث:

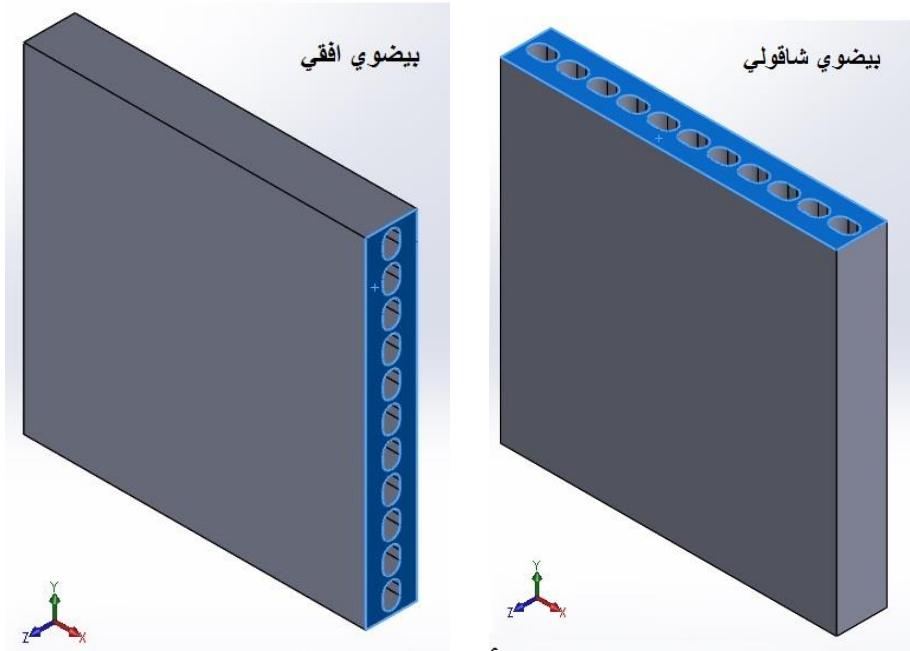
تمت هذه الدراسة بالاعتماد على الطرق العددية، وذلك باستخدام برنامج النمذجة والمحاكاة برنامج انسيس ANSYS وسيرفراته لنمذجة التصاميم المدروسة حرارياً، ومن أجل تصميم النماذج تم الاعتماد على برنامج سوليد وركس SOLID WORKS .

تم إجراء هذا البحث على عدة مراحل وهي:

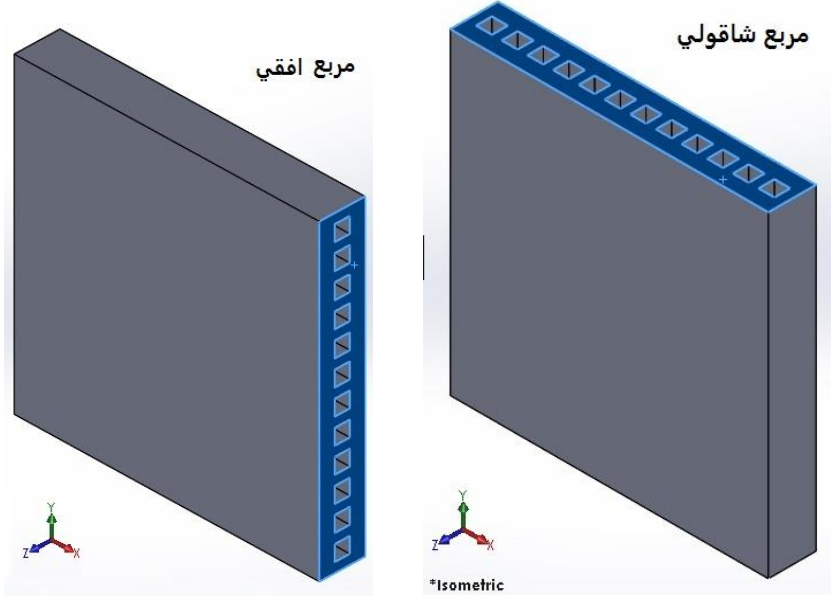
- 1- اقتراح وتصميم النماذج الهندسة المدروسة.
- 2- تصدير النماذج المدروسة إلى برنامج المحاكاة وقد تم استخدام برنامج النمذجة والمحاكاة انسيس ANSYS.
- 3- اجراء تقطيع عملية mesh (تقطيع) للنموذج المدروس.
- 4- تصدير الأشكال المقطعة التي جرت عليها عملية ميش (mesh) من ANSYS Workbench إلى ANSYS Fluent.
- 5- البدء بوضع الاعدادات عملية التشغيل.
- 6- تحديد الشروط الحدية.
- 7- البدء بإجراء الحل، والانتظار حتى الوصول إلى نهاية الحل (تقارب مقبول).
- 8- قراءة وتحليل النتائج.
- 9- ترتيب نتائج عملية النمذجة وصياغة نتائج نهائية، مبنية على نتائج النمذجة.

1.4. تصميم النماذج المدروسة:

تم تصميم أربعة نماذج على برنامج سوليد وركس SOLID WORKS ، النموذج الأول فجواته بيضوية شاقولية، والنموذج الثاني فجواته بيضوية افقية، والنموذج الثالث فجواته مربعة شاقولية، والنموذج الرابع فجواته مربعة افقية مبيبة النماذج المذكورة في الأشكال 1 و 2 و 3 و 4 على الترتيب.



الشكل (1) النموذج الأول ذو فجوات بيضوية شاقولية الشكل (2) النموذج الثاني ذو فجوات بيضوية أفقية



الشكل (3) النموذج الثالث ذو فجوات مربعة شاقولية الشكل (4) النموذج الرابع ذو فجوات مربعة افقية

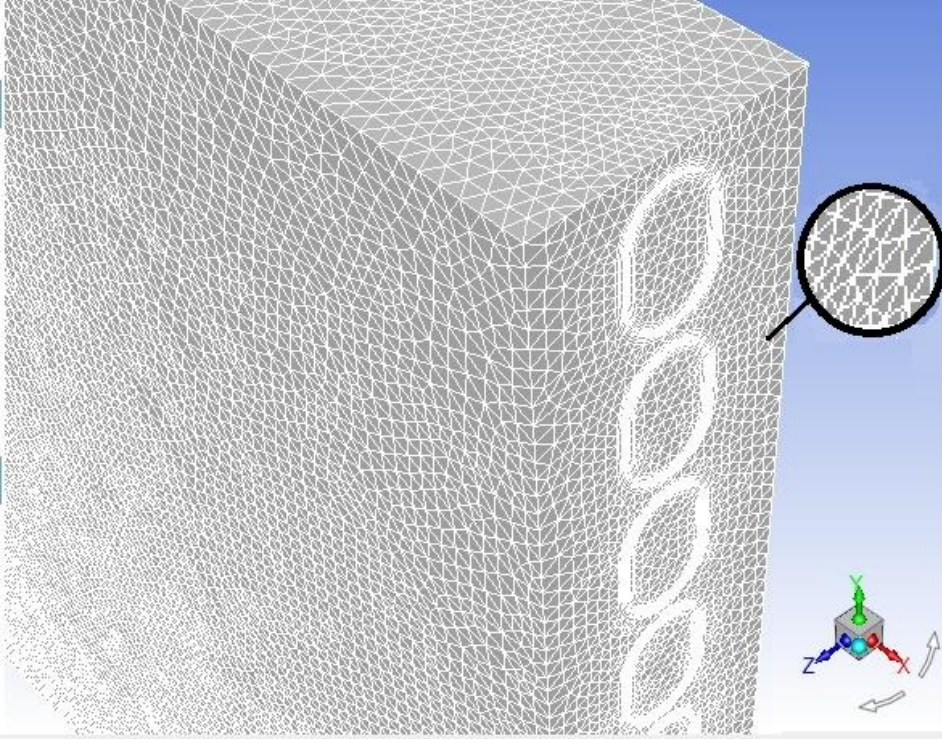
الأبعاد الخارجية لجميع النماذج المقترحة متساوية، وهي $1 \times 1 \times 0.15$ m ، وكانت نسبة الفراغ الهوائي لجميع النماذج 20% من مجمل حجم النموذج، الفجوات تتخلل النماذج من الطرفين وواقعة في الثلث الثاني من السماكة.

2.4. تصدير النماذج المدروسة إلى برنامج المحاكاة:

هذه المرحلة تتبع مرحلة تصميم النماذج، حيث بعد الانتهاء من تصميم النماذج يتم حفظ التصميم، ويتم تصديره إلى برنامج النمذجة الذي سيتم استخدامه، حيث تم تصدير كل نموذج مدروس إلى برنامج SOLID WORKS ليقوم البرنامج بالتعرف على النموذج وإجراء ملائمة له وتعريفه على بيئة البرنامج من أجل إتمام مراحل النمذجة اللاحقة.

3.4. إجراء عملية mesh (تقطيع) للنموذج المدروس:

يتم اجراء عملية تقطيع (mesh) للنموذج المدروس وذلك باستخدام التعليمة mesh في برنامج SOLID WORKS ، حيث تم تقطيع النموذج إلى خلايا رباعية الشكل من نوع (Tetra hydra)، وتستمر هذه العملية فترة زمنية بعد إعطاء الأمر. ويبين الشكل (5) تقطيع أحد النماذج المدروسة والتي يظهر فيها شكل الخلايا.

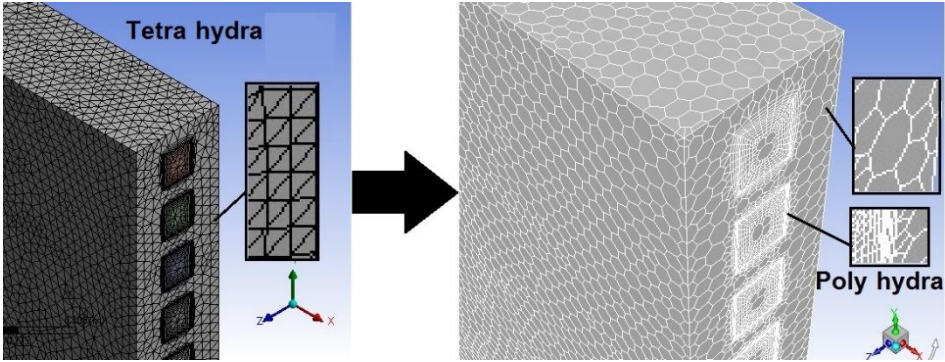


الشكل (5) يبين تقطيع أحد النماذج المدروسة، ويظهر فيها شكل الخلايا

4.4. تصدير الأشكال المقطعة التي جرت عليها عملية ميش (mesh) من ANSYS

Workbench إلى ANSYS Fluent:

بعد أن تمت عملية تقطيع النموذج في المرحلة السابقة يتم حفظ النموذج ثم تصديره إلى سيرفر آخر من سيرفرات برنامج انسيس وهو ANSYS Fluent ، تم بعد ذلك تحويل شكل خلايا النموذج من (Tetra hydra) إلى (Poly hydra) كما هو ظاهر في الشكل(6).



الشكل (6) تحويل شكل خلايا النموذج من (Tetra hydra) إلى (Poly hydra)، وتنعيم

السطح جوار السطح الملامس للمائع

5.4. إعدادات النمذجة وتحديد الشروط الحدية:

في هذه المرحلة تم اجراء تحسين لسطوح النموذج وذلك بإجراء تنعيم للخلايا بجوار الجدار الملامس للمائع (inflation) كما هو موضح في الشكل (6) السابق، وتم بعد ذلك وضع الخصائص الأساسية للحل، فكان معامل التوصيل الحراري للجزء الصلب من النماذج 1.4 W/m.K ، والسعة الحرارية 900 J/kg.K والكثافة 2200 kg/m^3 [7]، تم بعد ذلك وضع الشروط الحدية والتي تم تحديدها بفرض أن الدراسة تمت في فصل الشتاء وفق مناخ مدينة حمص السورية [7]، حيث تم تحديد درجة حرارة خارجية 3°C ودرجة حرارة داخلية 22°C وتحديد معامل التمدد الحجمي 3.5×10^{-3} ، وتم ضبط البرنامج على أساس المعطيات المذكورة.

تم بعد ذلك تحديد المعادلات الرياضية الداخلة بالحل.

وتم بعد ذلك تحديد المعادلات المستخدمة في النمذجة وهي: [8]

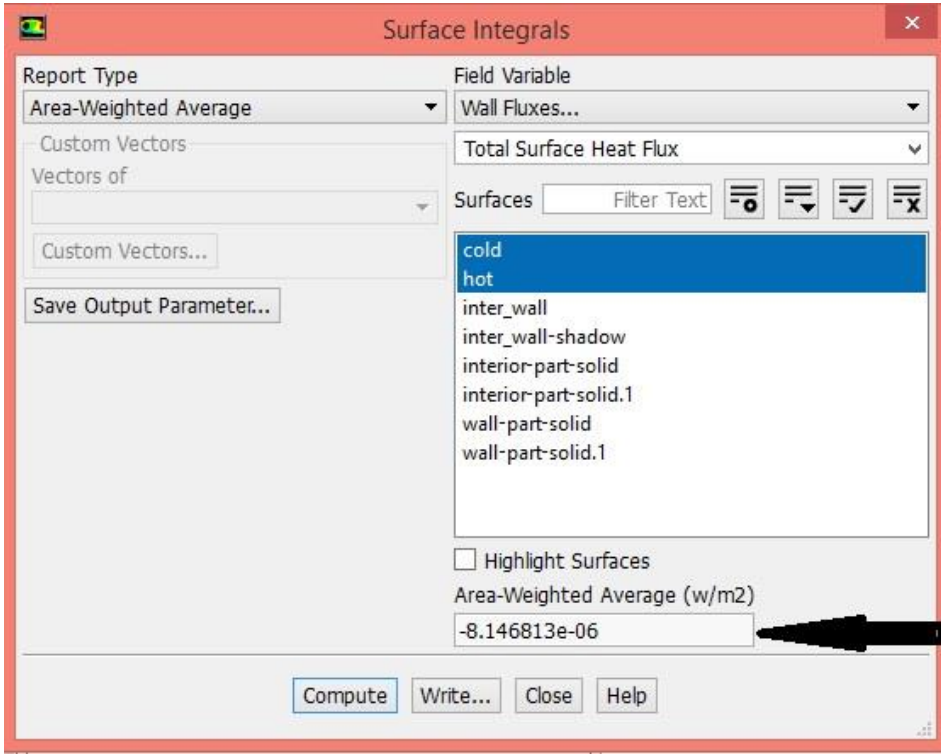
- 1- مبدأ حفظ الكتلة (معادلة الاستمرار).
 - 2- مبدأ حفظ كمية الحركة (معادلة كمية الحركة).
على المحور X وعلى المحور Y على المحور Z
 - 3- مبدأ حفظ الطاقة (معادلة الطاقة).
- بالإضافة لبعض المعادلات التي يحددها البرنامج تلقائياً.

6.4. البدء بالحل للوصول لتقارب مقبول:

يتم البدء بالحل بالانتقال الى تبويب الحل وتحديد عدد الحلول المطلوب من البرنامج القيام بها، فإن كانت الحلول كافية للوصول لتقارب مقبول يتم انتهاء الحل، وإن لم تكن كافية فيتم تحديد عدد حلول جديد لإتمام الحل. استمر الحل فترة زمنية طويلة في جميع النماذج المدروسة، وتم الحصول على تقاربات مقبولة من مرتبة 10^{-13} وهي تقاربات جيدة.

5. النتائج والمناقشة:

بعد إتمام البرنامج لعملية النمذجة، واعطائه رسالة تفيدنا بانتهاء الحل العددي والحصول على تقارب جيد ومقبول، تم البدء بعملية قراءة النتائج من البرنامج، حيث تم الاعتماد في تقييم النماذج والمفاضلة بينها من خلال كمية الحرارة المتدفقة عبر كل نموذج، حيث تتم المقارنة بين كل نموذجين متشابهين من حيث شكل الفجوة ومختلفين من حيث اتجاهها، فيتم مقارنة النموذج ذو الفجوات البيضوية الشاقولية مع النموذج ذو الفجوات البيضوية الافقية، والنموذج ذو الفجوات المربعة الشاقولية مع النموذج ذو الفجوات المربعة الافقية. بالإضافة لكميات الحرارة المتدفقة عبر كل نموذج يقوم البرنامج بإظهار توازن الطاقة في كل نموذج من النماذج المدروسة، والتي تعتبر مؤشراً هاماً من مؤشرات صحة الحل. ويبين الشكل (7) توازن الطاقة في أحد النماذج المدروسة بعد الانتهاء من الحل.



الشكل (7) توازن الطاقة في أحد النماذج المدروسة بعد الانتهاء من الحل

نلاحظ من الشكل السابق أن كمية الحرارة المتدفقة عبر السطح الساخن متقاربة جداً مع الحرارة المتدفقة عبر السطح البارد، وأن الفرق بينهما هو عدد صغير جداً من مرتبة 10^{-6} ، وهذا يعطي مؤشر إلى أن عملية النمذجة سليمة، حيث يتم التأكد بالعادة من جودة عملية النمذجة من خلال الفرق في كمية الحرارة بين الطرف الساخن والطرف البارد، فكلما كان العدد صغير كلما كانت النمذجة أكثر صحة. تم أيضاً حساب معدل التدفق الحراري النظري عبر النموذج المصمت (الحالة المرجعية) باستخدام نفس المعاملات الفيزيائية، حيث يقتصر انتقال الحرارة فيه على آلية التوصيل فقط. وقد أسفر الحساب النظري عن قيمة قدرها 233.33 W/m^2 ، وعند مقارنة هذه القيمة النظرية مع النتيجة العددية التي وفرها البرنامج، والتي

بلغت 234.22 ، يُلاحظ تقارب كبير بينهما. حيث يُقدّر الفرق بين القيمتين بنسبة خطأ نسبي لا تتجاوز 0.38 % ، تُعدّ هذه النسبة الضئيلة من الخطأ مؤشراً على دقة النمذجة العددية وموثوقية إعدادات النموذج الشبكي، مما يُعزز من مصداقية النتائج العددية المُتحصل عليها ويُبرر الاعتماد عليها في التحليلات اللاحقة.

تم أيضاً حساب معامل انتقال الحرارة الكلي لكل نموذج بالاعتماد على كمية الحرارة المقروءة من البرنامج لكل نموذج، وفقاً لمعادلة:

$$Q = U \cdot A \cdot \Delta T \Rightarrow U = Q / A \cdot \Delta T$$

حيث:

Q: كمية الحرارة المتدفقة عبر النموذج يقدر بـ [W]

U: معامل انتقال الحرارة الكلي يقدر بـ [W/m².K]

A: مساحة سطح العينة التي تعبر الحرارة بين طرفيها تقدر بـ [m²]

ΔT: فرق درجات الحرارة بين طرفي الجدار (T_n-T_c) يقدر بـ [K]

تم ترتيب نتائج كمية الحرارة المتدفقة عبر كل نموذج ومعامل انتقال الحرارة الكلي الذي تم حسابه في باستخدام المعادلة السابقة لكل نموذج في الجدول (1) وفق الآتي:

الجدول (1) يبين كميات الحرارة المتدفقة عبر النماذج المدروسة ومعامل انتقال الحرارة الكلي

لها

معامل انتقال الحرارة الكلي [W/m ² .K]	كمية الحرارة المتدفقة عبر النموذج [W/m ²]	النموذج		الرقم النموذج
		اتجاه الفجوة	شكل الفجوة	
5.341	133.54	شاقولية	بيضوي	1

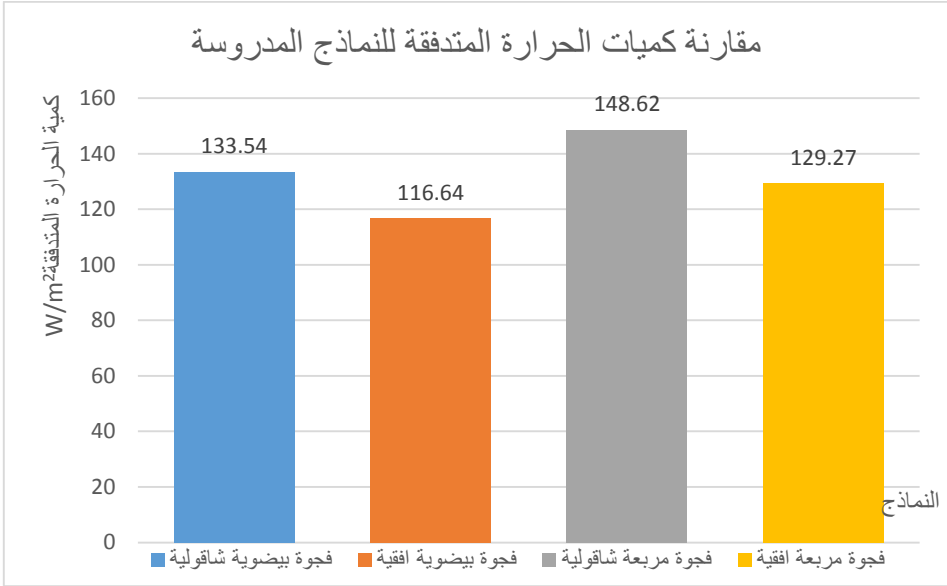
4.665	116.64	افقية		2
5.9448	148.62	شاقولية	مربعة	3
5.17	129.27	افقية		4

يبين لنا الجدول (1) كمية الحرارة المتدفقة عبر النماذج المدروسة ومعامل انتقال الحرارة الكلي لكل نموذج، حيث يتضح لنا أن كمية الحرارة المتدفقة عبر النموذج الأول (ذو الفجوات البيضوية الشاقولية) بلغت 133.54 W/m^2 وكان معامل انتقال الحرارة الكلي للنموذج المذكور $5.341 \text{ W/m}^2.\text{K}$ ، ونجد بأن كلاً من كمية الحرارة المتدفقة ومعامل انتقال الحرارة الكلي للنموذج الثاني (ذو فجوات البيضوية الافقية) كانت 116.64 W/m^2 و $4.665 \text{ W/m}^2.\text{K}$ على الترتيب، وبلغت كمية الحرارة العابرة ومعامل انتقال الحرارة الكلي للنموذج الثالث (ذو الفجوات المربعة الشاقولية) 148.62 W/m^2 و $5.9448 \text{ W/m}^2.\text{K}$ على التوالي، بينما بلغت كمية الحرارة العابرة ومعامل انتقال الحرارة الكلي للنموذج الرابع (ذو الفجوات المربعة الافقية) 129.27 W/m^2 و $5.17 \text{ W/m}^2.\text{K}$ على التوالي.

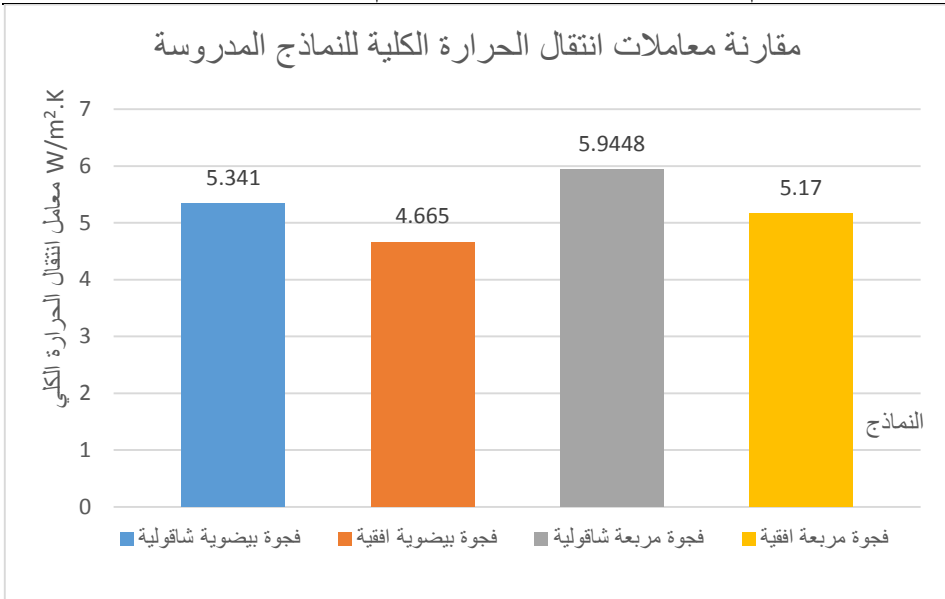
بمقارنة النموذج ذو الفجوات الشاقولية مع النموذج ذو الفجوات الافقية للشكل البيضوي نجد بأن النموذج الافقي تفوق حرارياً على النموذج الشاقولي، حيث كان التدفق الحراري للنموذج الافقي أقل منه في النموذج الشاقولي بالرغم من تساوي نسب الفجوات الهوائية والتشابه الهندسي التام بين النموذجين مع وجود فارق واحد هو اتجاه الفجوات، وبإجراء نفس المقارنة بين النموذجين الثالث والرابع وهما الشاقولي والافقي لشكل الفجوات المربعة نجد أيضاً أن النموذج الافقي أفضل حرارياً من النموذج الشاقولي، حيث تدفقت من النموذج الافقي كمية حرارة أقل.

من خلال تلك المقارنتين يمكن الملاحظة أن الاتجاه الافقي للفجوات يعطي أداءً حرارياً أفضل من الاتجاه الشاقولي للفجوات عند تشابه الشكل الهندسي والشروط الحدية المطبقة. يبين

الشكل (8) مخططاً شريطياً فيه مقارنة كميات الحرارة المتدفقة عبر النماذج، ويبين الشكل (9) مخططاً شريطياً فيه مقارنة بين المعاملات الكلية لانتقال الحرارة.



الشكل (8) يبين مخططاً شريطياً فيه مقارنة كميات الحرارة المتدفقة عبر النماذج



الشكل (9) يبين مخططاً شريطياً فيه مقارنة معاملات انتقال الحرارة الكلية للنماذج المدروسة

نجد من الشكل (9) أن معامل انتقال الحرارة الكلي للنموذج ذو للفجوات الافقية أقل منه في الفجوات الشاقولية سواءً في مقارنة النماذج ذات شكل فجوات بيضوية أو مربعة، حيث كانت قيمة معامل انتقال الحرارة الكلي للنموذج الافقي البيضوي أقل منه في النموذج الشاقولي البيضوي بنسبة % 12.66 ، وعند اجراء نفس المقارنة في النماذج ذات شكل فجوة مربعة نجد بأن النموذج ذو الفجوات الافقية المربعة يملك معامل انتقال حرارة كلي أقل منه في النموذج ذو الفجوات الشاقولية المربعة بنسبة % 13.03 .

بالمحصلة ومن خلال النتائج نجد أن النماذج ذات الفجوات الافقية تتفوق على النماذج ذات الفجوات الشاقولية من الناحية الحرارية، فكمية الحرارة المتدفقة عبر النماذج ذات الفجوات الافقية أصغر من كمية الحرارة المتدفقة عبر النماذج ذات الفجوات الشاقولية.

من أجل تفسير تفوق النماذج ذات الفجوات الأفقية على تلك التي تملك فجوات شاقولية، نقوم بإجراء تحليل يظهر لنا بعض التفسيرات للنتائج.

انتقال الحرارة بين طرفي الجدار في الحالة المدروسة يتم بشكلين: الأول هو انتقال الحرارة بالتوصيل الحراري في الجزء الصلب من الجدار، والثاني هو انتقال الحرارة بالحمل عبر الهواء داخل فجوات الجدار.

النموذج ذو الفجوات الشاقولية يشابه هندسياً النموذج ذو الفجوات الأفقية تماماً، من حيث نسبة الفراغ وعدد الفجوات وشكلها ومساحة سطحها وباقي الأبعاد الهندسية، مع وجود فارق واحد فقط هو اتجاه الفجوات داخل النماذج، فالأول فجواته شاقولية أما الثاني فجواته أفقية.

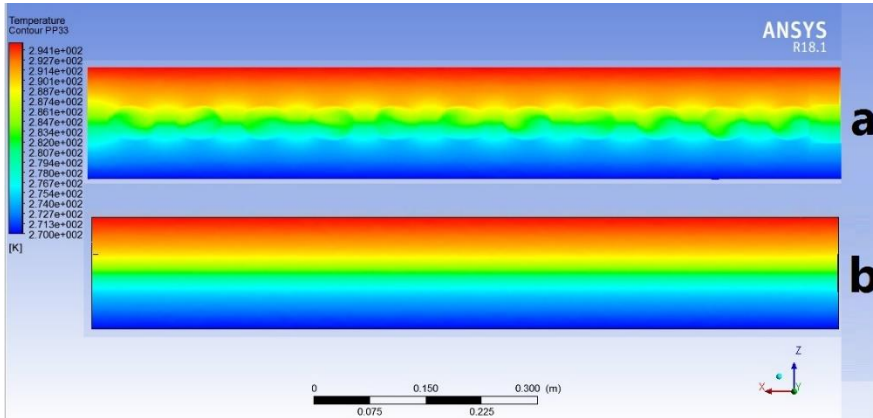
أن تشابه النموذجين هندسياً وتساوي الأبعاد ولا سيما مقاطع الجدار المختلفة والأعصاب التي تنتقل عبرها الحرارة يجعل من عملية انتقال الحرارة بالتوصيل بين طرفي الجدار عملية ذات آلية متشابهة لكلا النموذجين، أي أن عملية انتقال الحرارة بالتوصيل ليس لها تأثير على اختلاف كمية الحرارة المنتقلة في كلا النموذجين، وبالتالي فإن سبب اختلاف كمية الحرارة المنتقلة في النموذج الأول عن تلك الكمية في النموذج الثاني تعود لعملية انتقال الحرارة بالحمل الذي يحدث داخل الفجوات الهوائية والذي يعتبر السبب الرئيسي في الاختلاف.

إن الوضع الشاقولي للفجوات في النموذج ذو الفجوات الشاقولية واختلاف درجات الحرارة بين طرفي الجدار يزيد من جريان الهواء بالحمل الحر داخل الحبرات.

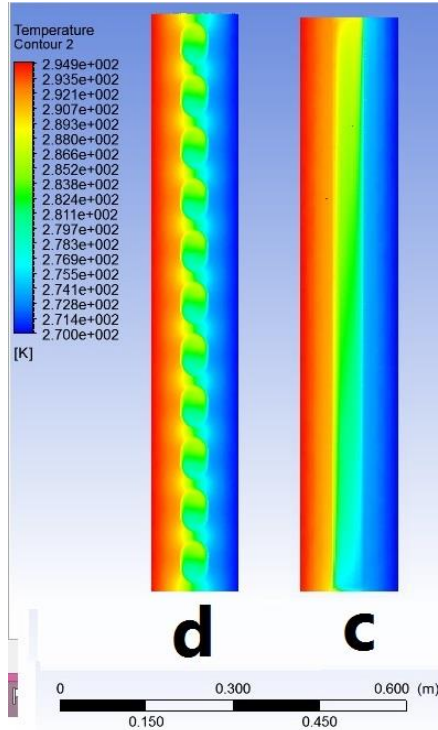
إن الفجوات الشاقولية تتسبب في حدوث حركة حرارية بشكل عمودي من أعلى الجدار إلى أسفله وبالعكس، مما يخلق نوعاً من التيارات الهوائية المتدفقة عامودياً عبر الجدار، وهذه الحركة الهوائية تزيد من معدل انتقال الحرارة بالحمل الحر، حيث يساهم انتقال الحرارة عبر الهواء في الجدار بتقليل المقاومة الحرارية. ومن المعلوم أن الفجوات الشاقولية قد تساهم في زيادة تيارات الحمل الحراري داخل الجدران، لأن الهواء الحار يميل للارتفاع والهواء البارد يهبط مما يساهم في زيادة كفاءة نقل الحرارة.

إن توضع الفجوات بشكل افقي يؤدي الى تدفق الهواء بشكل افقي، مما يقلل من حركة الهواء الصاعدة والهابطة مقارنةً بنموذج الفجوات الشاقولية، وفي هذه الحالة تزداد مقاومة الحمل الحرارية لأن حركة الهواء محدودة ولا تستفيد من القوانين الطبيعية لتدفق الحرارة الرأسية، بسبب محدودية ارتفاع الفجوات. عادةً ما تعمل الفجوات الافقية على تقليل التدفق الحراري عبر الجدار لأنها تشكل تيارات غير فعالة من حيث انتقال الحرارة، كما أن التيارات الافقية تعمل على توزيع الحرارة بشكل افقي موزع تمنع تسريع انتقال الحرارة عبر الجدار في الاتجاه العمودي، الذي يحدث عادةً في الجدران المسخنة.

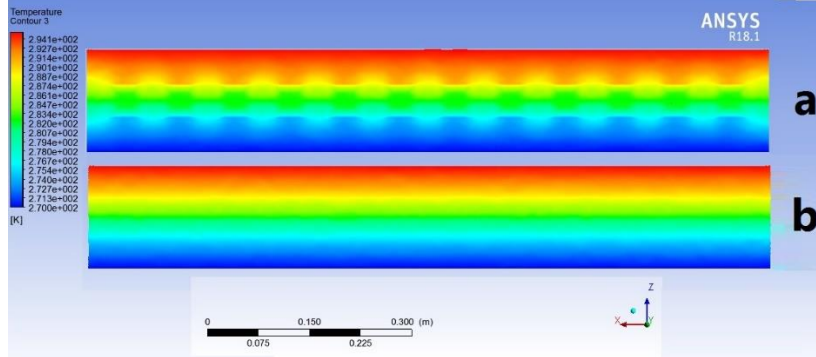
من الطبيعي أن يتدفق بين طرفي النموذج ذو الفجوات الشاقولية كمية حرارة أكبر مقارنةً بالنموذج ذو الفجوات الافقية، بسبب الاختلاف في طريقة تكوين تيارات الهواء (حمل حراري شاقولي مقابل حمل حراري افقي)، حيث النموذج الأول يسمح بانتقال كمية حرارة أكبر بفضل التيارات الشاقولية التي تسهم في تسريع التبادل الحراري بين الأطراف. تبين الأشكال (10) و(11) مخططات توزع درجات الحرارة في الجدران ذات الفجوات البيضوية.



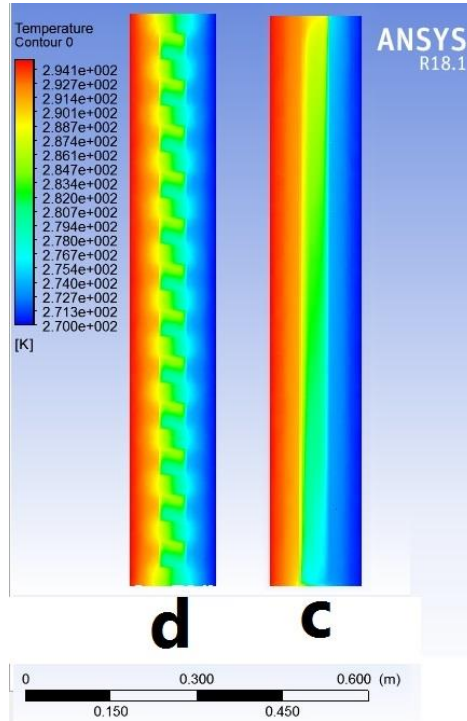
الشكل (10) يبين توزع درجات الحرارة في مقطع افقي في الجدران اللذان يحتويان على فجوات بيضوية حيث a هو للفجوات البيضوية الشاقولية و b للفجوات البيضوية الافقية



الشكل (11) يبين توزيع درجات الحرارة في مقطع جانبي في الجدران اللذان يحتويان على فجوات بيضوية حيث **c** هو للفجوات البيضوية الشاقولية و **d** للفجوات البيضوية الأفقية وتبين الأشكال (12) و(13) مخططات توزيع درجات الحرارة في الجدران ذات الفجوات المربعة.



الشكل (12) يبين توزيع درجات الحرارة في مقطع افقي في الجداران اللذان يحتويان على فجوات مربعة حيث a هو للفجوات المربعة الشاقولية و b للفجوات المربعة الافقية



الشكل (13) يبين توزيع درجات الحرارة في مقطع جانبي في الجداران اللذان يحتويان على فجوات مربعة حيث c هو للفجوات المربعة الشاقولية و d للفجوات المربعة الافقية

6. الاستنتاجات والتوصيات:

- بالمحصلة ومن خلال نتائج الدراسة الحالية، يمكن استخلاص مجموعة من الاستنتاجات على النحو الآتي:
1. حقق نموذج الجدار المطبوع بتقنية الطباعة ثلاثية الأبعاد ذو الفجوات الأفقية تفوقاً على نظيرها ذو الفجوات الشاقولية من حيث الأداء الحراري للجدران (مقاومة حرارية أكبر).
 2. تفوق النموذج ذو الفجوات الأفقية على النموذج ذو الفجوات الشاقولية عندما كان شكل الفجوة بيضوياً بنسبة % 12.66 بينما كان هذا التفوق بنسبة % 13.03 عندما كان شكل الفجوة مربعاً، حيث تم هذا التقييم بالاعتماد على كمية الحرارة الأقل المتدفقة.
 3. لعبت تيارات الحمل الحر دوراً أساسياً في عملية انتقال الحرارة في النماذج المدروسة، وكان لها الدور الأكبر في عملية تقييم الأداء الحراري، في ظل تشابه دور انتقال الحرارة بالتوصيل في العينات ذات شكل فجوات متشابهة، وذلك لتشابه وتساوي المقاطع الهندسية الناقلة للحرارة بين طرفي الجدار.
 4. حقق النموذج ذو الفجوات الأفقية وشكل فجوات بيضوي وفراً طاقياً على النموذج ذو الفجوات الشاقولية وشكل فجوات بيضوي بنسبة % 12.66، وكان هذا الوفراً بنسبة % 13.03 في النموذج ذو الفجوات الأفقية وشكل فجوات مربعة مقارنةً بالنموذج ذو الفجوات الشاقولية وشكل فجوات مربعة.

التوصيات:

من خلال ما سبق يمكن أن نوصي بما يلي:

- 1- استخدام نموذج الفجوات الأفقية بدلاً من الشاقولية عند تطبيق تقنية طباعة الجدران ثلاثية الأبعاد،
- 2- ضرورة إجراء باقي الاختبارات على النماذج المدروسة من أجل التأكد من الخواص الميكانيكية للجدران وتحديد مدى متانتها ميكانيكياً.
- 3- إجراء دراسات تهدف لتحسين خواص حبر الطباعة لتحسين الخواص الحرارية والميكانيكية.

7. المراجع

1. Buswell R A. et al. 2018- **3D printing using concrete extrusion: A roadmap for research.** *Cement and Concrete Research, ScienceDirect (2018), 0008-8846/ © 2018 The Authors. Published by Elsevier Ltd.*
2. Wolfs R J M. et al 2018- **Early age mechanical behaviour of 3D printed concrete: Numerical modelling and experimental testing.** *Cement and Concrete Research 106 (2018), ScienceDirect, Page 103–116.*
3. Perrot A. et al, **Structural built-up of cement-based materials used for 3D printing extrusion techniques.** *Materials and Structures (2016) 49 Page 1213–1220.*
4. Marais H. et al, (2021). **Computational assessment of thermal performance of 3D printed concrete wall structures with cavities.** *Science Direct, Journal of Building Engineering 41 (2021) 102431, Page 12.*
5. Alzahrani A. et al, (2022). **Computational Optimization of 3D-Printed Concrete Walls for Improved Building Thermal Performance,** *Science Direct, Buildings 2022, 12(12), 2267;*
6. Ravula R. et al, (2023). (2023). **On the thermal-energy performance of 3D printed concrete wall panels designed with varying insulation ratios.** *Journal of Building Engineering, Volume 77 (15)*

7. **Syrian Arab Code for the design and implementation of installations in reinforced concrete.**
8. **Qarmo A. 2021- The study of improving thermal performance of bricks manufactured from the recycling of rubble. PhD thesis, Pages 147-148.**

تحليل صور الرنين المغناطيسي لمرض الزهايمر باستخدام تحسين التباين والشبكات العصبية

م. علي سليمان*، د. عبد الكريم السالم**

*طالب ماجستير، قسم هندسة الإلكترونيات والاتصالات، كلية الهندسة الميكانيكية والكهربائية، جامعة حمص **أستاذ في قسم هندسة الإلكترونيات والاتصالات، كلية الهندسة الميكانيكية والكهربائية، جامعة حمص

الملخص

يُعدّ مرض الزهايمر Alzheimer's Disease – AD اضطرابًا تنكسيًا عصبيًا تدريجيًا يشكّل تحديًا كبيرًا لأنظمة الرعاية الصحية على مستوى العالم. ولا تزال عملية التصنيف المبكر والدقيق لمراحل المرض باستخدام التصوير بالرنين المغناطيسي MRI مهمة صعبة، وذلك بسبب الفروق البنوية الطفيفة بين مراحل المرض، وانخفاض تباين الصور.

يقترح هذه البحث إطارًا محسنًا قائمًا على صور الرنين المغناطيسي لتصنيف مرض الزهايمر متعدد المراحل، حيث يدمج بين تحسين التباين وتقنيات التعلّم العميق واستراتيجيات التصنيف الهجين. ولتحسين جودة الصور وإبراز البنى التشريحية ذات الأهمية التشخيصية، يتم تطبيق معادلة تسوية الهيستوغرام التكيفي ذات التوزيع الغاوسي

Gaussian-based Contrast-Limited Adaptive Histogram Equalization – G-CLAHE

خلال مرحلة المعالجة المسبقة. بعد ذلك، تُستخدم الشبكة العصبية الالتفافية Convolutional Neural Network – CNN لاستخلاص تمثيلات عميقة للميزات من الصور المحسنة. وفي مرحلة التصنيف، تم استخدام كلٍّ من نموذج CNN مستقل، وإطار تصنيف هجين يجمع بين CNN وآلة متجه الدعم CNN-SVM.

تم تقييم المنهجية المقترحة باستخدام مجموعة بيانات عامة لصور الرنين المغناطيسي، حيث يتم تصنيف الحالات إلى أربع فئات: طبيعي إدراكيًا، ضعف إدراكي بسيط مبكر، ضعف إدراكي بسيط متأخر، ومرض الزهايمر. أظهرت النتائج التجريبية أن تحسين التباين يسهم بشكل ملحوظ في زيادة

تحليل صور الرنين المغناطيسي لمرض الزهايمر باستخدام تحسين التباين والشبكات العصبية قابلة تمييز الميزات. وقد حقق نموذج CNN المستقل دقة تصنيف كلية بلغت %96، في حين أظهر النموذج الهجين CNN-SVM تحسناً إضافياً في الأداء، محققاً دقة %97 مع قيم استرجاع أعلى ومثانة أفضل، لا سيما في تصنيف المراحل المرضية ضعيفة التمثيل. الكلمات المفتاحية: مرض الزهايمر، الشبكة العصبية الالتقافية، تحسين التباين، التعلّم العميق، التصوير بالرنين المغناطيسي، آلة متجه الدعم.

MRI Image Analysis for Alzheimer's Disease Using Contrast Enhancement and Neural Networks

Abstract:

Alzheimer's disease - AD is a progressive neurodegenerative disorder that poses a significant challenge to healthcare systems worldwide. Early and accurate classification of AD stages using magnetic resonance imaging-MRI remains difficult due to subtle structural differences between disease stages, low image contrast.

This paper presents an enhanced MRI-based framework for multi-stage Alzheimer's disease classification that integrates contrast optimization with deep learning and hybrid classification strategies. To improve image quality and highlight diagnostically relevant anatomical structures, a Gaussian-based Contrast-Limited Adaptive Histogram Equalization -G-CLAHE technique is applied during preprocessing. A convolutional neural network - CNN is then employed to extract deep feature representations from the enhanced images. For classification, both a standalone CNN model and a hybrid CNN-Support Vector Machine CNN-SVM framework are investigated.

The proposed approach is evaluated on a publicly available MRI dataset and classifies subjects into four categories: cognitively normal, early mild cognitive impairment, late mild cognitive impairment, and Alzheimer's disease. Experimental results demonstrate that contrast enhancement significantly improves feature discriminability. The standalone CNN achieves an overall classification accuracy of 96%, while the hybrid CNN-SVM model further improves performance, reaching 97% accuracy with

higher recall and robustness, particularly for underrepresented disease stages.

Keywords: Alzheimer's disease, convolutional neural networks, contrast enhancement, deep learning, magnetic resonance imaging, support vector machine

1. المقدمة:

يُعدّ مرض الزهايمر Alzheimer's Disease – AD أكثر الاضطرابات التنكسية العصبية انتشارًا، والسبب الرئيسي للخرف على مستوى العالم. ويتميز بتدهور معرفي تدريجي، وضعف في الذاكرة، وتراجع في الوظائف الإدراكية، مما يؤدي في المراحل المتقدمة إلى إعاقة شديدة وفقدان الاستقلالية [1]. ومع الازدياد المستمر في أعداد كبار السن، يتصاعد العبء العالمي لمرض الزهايمر بوتيرة متسارعة، الأمر الذي يفرض حاجة ملحة إلى حلول تشخيصية مبكرة وموثوقة. ويُعدّ التصوير بالرنين المغناطيسي Magnetic Resonance Imaging – MRI أحد أكثر تقنيات التصوير العصبي استخدامًا في تقييم مرض الزهايمر، وذلك لما يتميز به من كونه غير تدخلية وقدرته على توفير معلومات بنيوية عالية الدقة عن تشريح الدماغ. ويلعب التصوير البنيوي بالرنين المغناطيسي دورًا محوريًا في الكشف عن أنماط ضمور الدماغ، ولا سيما في مناطق مثل الحُصين Hippocampus والفصوص الصدغية الإنسية، والتي ترتبط ارتباطًا وثيقًا بتطور مرض الزهايمر. ومع ذلك، لا يزال التصنيف الدقيق لمراحل المرض باستخدام صور MRI مثل تحديًا كبيرًا، خاصة في المراحل المبكرة والمتوسطة، حيث تكون التغيرات التشريحية طفيفة وغالبًا ما تتداخل بين الفئات المرضية المختلفة.

في السنوات الأخيرة، تم توظيف تقنيات التعلّم الآلي والتعلّم العميق على نطاق واسع في تشخيص مرض الزهايمر بشكل آلي. وتعتمد الأساليب التقليدية في التعلّم الآلي بشكل كبير على الميزات المصممة يدويًا، والتي غالبًا ما تعاني من ضعف القدرة على التعميم عند تطبيقها على بيانات

MRI غير المتجانسة. في المقابل، أظهرت الشبكات العصبية الالتفافية Convolutional Neural Networks – CNNs قدرة عالية على تعلّم تمثيلات هرمية للميزات مباشرة من الصور الطبية، وحققت نتائج واعدة في مهام تصنيف مرض الزهايمر. وعلى الرغم من هذه التطورات، لا تزال نماذج CNN حساسة لجودة الصور، الضجيج، وانخفاض التباين، وعدم توازن الفئات، وهي خصائص شائعة في مجموعات بيانات MRI الواقعية.

ويُعدّ جودة تباين الصور أحد العوامل الحاسمة، إلا أنه لم يحظَ بالاهتمام الكافي نسبيًا في تصنيف مرض الزهايمر المعتمد على صور الرنين المغناطيسي [2]. إذ غالبًا ما تعاني صور الرنين المغناطيسي من انخفاض التباين وعدم تجانس الشدة، مما قد يؤدي إلى طمس التفاصيل التشريحية الدقيقة ويؤثر سلبيًا على عملية استخراج الميزات العميقة. وعلى الرغم من تركيز العديد من الدراسات على تحسين بنية الشبكات العصبية، فإن القليل منها تناول الدمج المنهجي لتقنيات تحسين التباين المصممة خصيصًا لصور الدماغ قبل تطبيق نماذج التعلّم العميق. إن تعزيز البنى التشريحية ذات الأهمية التشخيصية قبل مرحلة استخراج الميزات يمكن أن يسهم بشكل كبير في تحسين متانة التصنيف، لا سيما في الكشف المبكر عن المرض.

إضافة إلى ذلك، وعلى الرغم من تفوق الشبكات العصبية الالتفافية في تعلّم الميزات، فإن طبقات التصنيف النهائية فيها لا توفر دائمًا حدود قرار دقيقة، خاصة عند التعامل مع بيانات محدودة أو غير متوازنة. وتُعدّ استراتيجيات التعلّم الهجين التي تجمع بين استخراج الميزات باستخدام CNN والمصنّفات التقليدية، مثل آلة متجه الدعم Support Vector Machine، بديلًا واعدًا. إذ تتميز خوارزميات SVM بقدرتها العالية على التعميم وفعاليتها في فضاءات الميزات عالية الأبعاد، مما يجعلها مناسبة بشكل خاص لمهام التصنيف التي تعاني من نقص العينات في بعض الفئات [3]، [4].

انطلاقًا من هذه التحديات، تقترح هذه الدراسة إطارًا محسنًا قائمًا على صور الرنين المغناطيسي لتصنيف مرض الزهايمر متعدد المراحل، يجمع بين تحسين التباين وتقنيات التعلّم العميق واستراتيجيات التصنيف الهجين. ويعتمد الإطار المقترح على استخدام تقنية المعادلة التكيفية

المحدودة على النقيض للهستوغرام المعتمدة على التوزيع الغاوسي Gaussian-based Contrast-Limited Adaptive Histogram Equalization – G-CLAHE

لتحسين جودة صور MRI وإبراز البنى التشريحية ذات الأهمية التشخيصية. بعد ذلك، يتم استخدام شبكة عصبية التلافيفية لاستخلاص الميزات العميقة من الصور المحسنة، يليها تصنيف باستخدام كل من نموذج CNN مستقل وإطار هجين يجمع بين CNN و SVM [5].

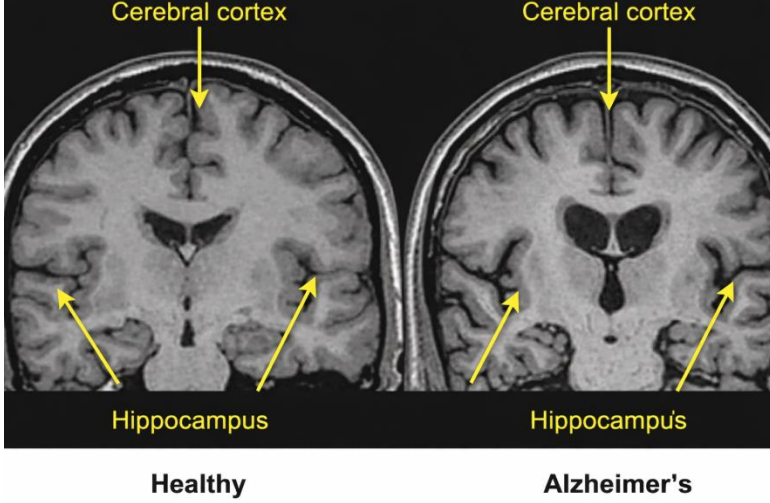
- المساهمات Contributions:

يمكن تلخيص المساهمات الرئيسية لهذا البحث على النحو الآتي:

1. اقتراح خطوات معالجة للصور لتحسين التباين قائم على التوزيع الغاوسي ومُصمَّم خصيصًا لصور الرنين المغناطيسي للدماغ، بهدف تحسين جودة الصور وقابلية تمييز الميزات في تصنيف مرض الزهايمر.
2. تطوير إطار تصنيف هجين يجمع بين CNN و SVM لدمج تعلم الميزات العميقة مع التصنيف القائم على الهامش، مما يعزز المتانة والحساسية، خاصةً في المراحل المرضية ضعيفة التمثيل.
3. إجراء تجارب موسَّعة باستخدام مجموعة بيانات عامة لصور MRI لتقييم فعالية المنهجية المقترحة بالاعتماد على عدة مقاييس أداء، بما في ذلك الدقة، والاسترجاع، ومقياس F1، وتحليل ROC-AUC.
4. تعزيز قابلية تفسير النموذج باستخدام تقنية Grad-CAM++، من خلال توفير تفسيرات بصرية تُبرز المناطق الدماغية ذات الصلة السريرية المرتبطة بمرض الزهايمر.

2.مراجعة الأدبيات والأعمال ذات الصلة

شهدت تقنيات التصوير والمعالجة تقدمًا ملحوظًا في تشخيص مرض الزهايمر Alzheimer's Disease-AD، مدفوعًا بالتطورات المتسارعة في مجال الذكاء الاصطناعي. وعلى وجه الخصوص، أظهرت الأساليب المعتمدة على التصوير والمقترنة بتقنيات التعلم الآلي والتعلم العميق إمكانات كبيرة في الكشف الآلي عن مرض الزهايمر وتحديد مراحله. ومن بين مختلف تقنيات التصوير، لا يزال التصوير بالرنين المغناطيسي MRI أحد أكثر الأدوات استخدامًا، نظرًا لطبيعته غير التداخلية، ودقته المكانية العالية، وقدرته على التقاط التغيرات البنوية في الدماغ المرتبطة بالتكس العصبي، مثل ضمور القشرة الدماغية cerebral cortex وانكماش الحصين hippocampus، كما هو موضح في الشكل 1.



الشكل (1) صورة بالرنين المغناطيسي لدماع سليم ودماع مصاب بمرض الزهايمر ويوضح الضمور البنيوي المرتبط بتطور المرض.

اعتمدت الأساليب المبكرة في تشخيص مرض الزهايمر باستخدام التعلّم الآلي بشكل أساسي على الميزات المصممة يدويًا والمستخرجة من صور الرنين المغناطيسي، مثل سماكة القشرة الدماغية، وحجم الحُصين، وواصفات النسيج. وعلى الرغم من تحقيق هذه الطرق لنجاح متوسط، إلا أن أدائها كان غالبًا محدودًا بسبب تعقيد تصميم الميزات وضعف قدرتها على التعميم عبر مجموعات البيانات المختلفة. ولمعالجة هذه القيود، تم الاعتماد بشكل متزايد على تقنيات التعلّم العميق، ولا سيما الشبكات العصبية الالتفافية Convolutional Neural Networks، التي تتيح تعلّم تمثيلات هرمية للميزات مباشرة من بيانات التصوير العصبي.

أظهرت العديد من الدراسات أن نماذج التعلّم العميق، وبخاصة الشبكات العصبية الالتفافية، فعالة للغاية في استخراج ميزات تمييزية من بيانات الرنين المغناطيسي MRI والتصوير المقطعي بالإصدار البوزيتروني PET. على سبيل المثال، اقترح Raza et al إطارًا عميقًا متعدد الوسائط يدمج بيانات PET وMRI مع المعلومات السريرية باستخدام بنية CNN-3D، محققًا دقة تصنيف بلغت 98.1%. وقد أكدت هذه الدراسة أهمية دمج البيانات متعددة الوسائط والجينية في تحسين الأداء التنبؤي [6]. وبالمثل، قدّم Nazir et al 2025 مراجعة منهجية ركزت على الاتجاهات المستقبلية في التنبؤ بمخاطر مرض الزهايمر باستخدام التعلّم العميق، حيث أشار إلى أن البنى

المتقدمة مثل ResNet-50 و DenseNet حققت دقة تشخيصية وصلت إلى 96.4% على مجموعات بيانات MRI و PET [7].

وفي سياق دمج بيانات PET و MRI، استكشف Woodward et al 2025 أساليب تشخيصية قائمة على الذكاء الاصطناعي عقب اعتماد تصوير Tauvid للكشف عن بروتين تاو. وحقق نموذج CNN ثلاثي الأبعاد الذي استخدمه دقة بلغت 97.2% في التمييز بين مرض الزهايمر والضعف الإدراكي البسيط، مما يبرز القيمة التشخيصية للمؤشرات الحيوية المرتبطة ببروتين تاو عند دمجها مع التعلم العميق [8]. وإلى جانب التصوير العصبي، حظيت المقاربات المعتمدة على المؤشرات الحيوية باهتمام متزايد، حيث استعرض Bhatia et al 2025 التقدم في اكتشاف المؤشرات الجزيئية والكيميائية العصبية باستخدام الذكاء الاصطناعي، واعتمدوا على خوارزمية آلة متجه الدعم SVM لتحليل بيانات السائل الدماغي الشوكي CSF و PET، محققين دقة بلغت 95.7% [9].

كما أكدت عدة مراجعات شاملة فعالية نماذج CNN في تشخيص مرض الزهايمر. فقد أشار Awang et al 2025 إلى أن الشبكات العصبية الالتفافية العميقة حققت دقة تراوحت بين 96.8% و 98.3% عند تطبيقها على بيانات MRI و fMRI ضمن إطار ADNI [10]. وفي السياق نفسه، اقترح Alsubaie et al 2024 بنية شبكة عصبية التلافية جديدة تُعرف باسم ConvADD لتشخيص مرض الزهايمر باستخدام صور الدماغ، حيث حقق النموذج دقة بلغت نحو 98% مع عدد معاملات منخفض نسبياً، مما يعكس كفاءة البنية المقترحة في استخراج الميزات التشخيصية من صور MRI [22]. وبالمثل، شدد Moraveji et al 2025 على مزايا النماذج الهجينة التي تجمع بين استخراج الميزات باستخدام CNN والمصنّفات التقليدية مثل SVM، لا سيما عند التعامل مع بيانات متعددة الوسائط تشمل PET, MRI, EEG، حيث تم تحقيق دقة بلغت 97.5% إلى جانب تحسين قابلية التفسير باستخدام تقنيات الذكاء الاصطناعي القابل للتفسير XAI [11].

كما تم استكشاف التنبؤ الشخصي بتطور مرض الزهايمر، حيث طور Koksalmis وآخرون 2025 نموذجاً تنبؤياً متعدد الطبقات يدمج بيانات MRI و PET والمؤشرات الحيوية السائلة والمعلومات الجينية، محققاً دقة بلغت 98.4% [12]. وفي المقابل، تناول Ehteshamzad

2024 نماذج تشخيصية قائمة على EEG ومقترنة بعلامات جينية، وحقق دقة بلغت 94.6% [13]. ولا تزال أساليب التعلّم الآلي التقليدية تلعب دورًا مهمًا في تشخيص مرض الزهايمر، إذ أظهر Abdalrada 2025 أن خوارزميات التجميع مثل XGBoost و Random Forest حققت دقة بلغت 95.3% عند تطبيقها على بيانات MRI و CSF والاختبارات النفسية العصبية [14]. وعلى الرغم من هذه النتائج الواعدة، فإن المقاربات العميقة متعددة الوسائط غالبًا ما تتطلب آليات معقدة لجمع البيانات، وموارد حسابية مرتفعة، ومراحل معالجة مسبقة مكثفة، مما يحدّ من قابليتها للتطبيق السريري. كما أن الوصول إلى بيانات متعددة الوسائط لا يكون متاحًا دائمًا في الممارسة السريرية اليومية، لا سيما في البيئات محدودة الموارد، الأمر الذي يعزز الحاجة إلى تطوير أطر تشخيصية تعتمد على بيانات MRI فقط.

ويُعد انخفاض التباين، ووجود الضجيج، وعدم تجانس شدة الإشارة من التحديات المهمة الأخرى في تصنيف مرض الزهايمر باستخدام صور الرنين المغناطيسي، حيث يمكن لهذه العوامل أن تحجب التفاصيل التشريحية الدقيقة وتؤثر سلبًا على استخراج الميزات العميقة. وعلى الرغم من تركيز العديد من الدراسات على تصميم بنى الشبكات العصبية، لم يحظَ تحسين الصور بشكل منهجي ومُكيّف خصيصًا لصور الدماغ بالاهتمام الكافي. وقد تم استكشاف أساليب تسوية الهستوغرام العالمية، إلا أنها قد تؤدي إلى تضخيم الضوضاء وتحسين مفرط لمناطق غير ذات صلة. في المقابل، تُعد تقنيات تحسين التباين التكيفية، مثل CLAHE، أكثر ملاءمة للتصوير الطبي، نظرًا لاعتمادها على المعالجة المحلية وفرض حدود على التباين للحد من تضخيم الضوضاء.

وبالتوازي مع ذلك، ازداد الاهتمام بأطر التصنيف الهجينة التي تجمع بين استخراج الميزات العميقة باستخدام CNN والمصنّفات التقليدية. إذ تعمل CNN كمستخرج قوي للميزات، بينما تتولى خوارزميات مثل SVM أو Random Forest أو Gradient Boosting عملية اتخاذ القرار النهائي. وتُظهر خوارزمية SVM على وجه الخصوص قدرة عالية على التعميم ومثانة في التعامل مع مجموعات البيانات المحدودة أو غير المتوازنة. وقد أظهرت عدة دراسات أن النماذج الهجينة CNN-SVM تتفوق على نماذج CNN المستقلة، خاصة في مهام التصوير الطبي التي تعاني من قلة العينات أو عدم توازن الفئات.

ومؤخرًا، تم دمج تقنيات الذكاء الاصطناعي القابل للتفسير، مثل Grad-CAM++، ضمن طرق تشخيص مرض الزهايمر، بهدف إظهار المناطق الدماغية المميزة التي تؤثر في قرارات النموذج، مما يعزز قابلية التفسير والثقة السريرية.

وبوجه عام، تشير الأدبيات العلمية إلى أنه على الرغم من النجاح الكبير الذي حققته نماذج التعلم العميق في تشخيص مرض الزهايمر، لا تزال هناك تحديات تتعلق بجودة البيانات، ومثانة النماذج، والتكلفة الحسابية. كما أن التأثير المشترك لتحسين التباين والتصنيف الهجين باستخدام CNN-SVM لم يُدرس بشكل موسّع في أطر تعتمد على MRI فقط لتحديد مراحل المرض. وانطلاقاً من هذه الملاحظات، تقترح هذه الدراسة منهجية قائمة على تحسين التباين باستخدام توزيع غاوسي ومتكاملة مع إطار CNN-SVM هجين لتصنيف مرض الزهايمر متعدد المراحل، بما يحقق توازناً بين دقة التشخيص، والكفاءة الحسابية، وقابلية التطبيق السريري.

3. مواد وطرائق البحث:

يصف هذا القسم الإطار المقترح لتصنيف مرض الزهايمر متعدد المراحل باستخدام التصوير بالرنين المغناطيسي MRI وتكامل المنهجية المقترحة بين تحسين التباين، واستخلاص الميزات العميقة باستخدام الشبكات العصبية الالتفافية CNNs، واستراتيجية تصنيف هجينة تعتمد على آلة متجه الدعم SVM.

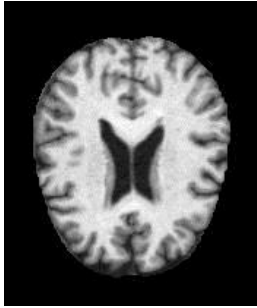
- وصف مجموعة البيانات Dataset Description

تم الحصول على مجموعة البيانات المستخدمة في هذه الدراسة من مستودع مفتوح الوصول مستضاف على منصة Kaggle وتحمل المجموعة عنوان Alzheimer's MRI Dataset، وهي متاحة للاستخدام العام دون قيود لأغراض البحث والتعليم. وتتضمن مجموعة البيانات ما يقارب 6,400 صورة رنين مغناطيسي للدماغ من نوع T1-weighted، تمثل مراحل مختلفة من تطور مرض الزهايمر. وقد صُنِّفت الصور إلى أربع فئات سريرية ذات صلة: غير مصاب بالخرف Non-Demented، خرف خفيف جداً Very Mild Demented، خرف خفيف Mild Demented، وخرف متوسط Moderate Demented ونتيج هذه الفئات إجراء تصنيف متعدد الأصناف ودراسة تطور المرض حيث يبين الجدول 1 مراحل التدهور المعرفي المتتالية.

الجدول (1) توزيع صور الرنين المغناطيسي عبر مراحل مرض الزهايمر.

الصنف	Number of Images
Non-Demented	640
Very Mild Demented	448
Mild Demented	179
Moderate Demented	13

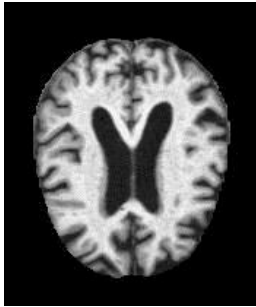
جميع الصور مقدمة على شكل مقاطع ثنائية الأبعاد بصيغ قياسية JPG/PNG، وتمت إعادة تحجيمها إلى دقة موحدة قدرها 128×128 بكسل لضمان الكفاءة الحسابية والتوافق مع نماذج التعلم العميق المقترحة كما يوضح الشكل 2 امثلة عن بيانات المرض في حالاته. كما جرى تسوية قيم البكسلات قبل التدريب لتحسين الاستقرار العددي.



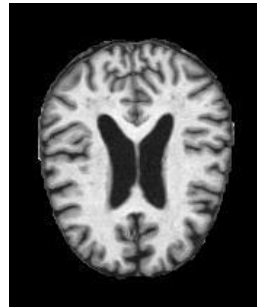
NonDemented



ModerateDemented



MildDeemenred



veryMildDemented

الشكل (2): مجموعة الصور لحالة المرض الاربعة

تُظهر مجموعة البيانات عدم توازن في الفئات، ولا سيما في فئة الخرف المتوسط، وهو ما يعكس التوزيع السريري الواقعي الشائع في دراسات مرض الزهايمر. ولتجنب تسرب البيانات، تم تقسيم البيانات عشوائياً إلى مجموعتي تدريب واختبار بنسبة 80:20، مع ضمان عدم وجود تداخل بين العينات.

وعلى الرغم من أن مجموعة البيانات ليست مجموعة سريرية أولية مثل ADNI أو OASIS، فقد استُخدمت على نطاق واسع في دراسات تصنيف مرض الزهايمر المعتمدة على التعلّم العميق، وتوفّر تنوعاً كافياً في الصور وحجماً مناسباً للمقارنة المرجعية لنماذج CNN كما تعرّز إتاحتها العامة من قابلية إعادة الإنتاج والشفافية.

- المعالجة المسبقة للصور وتحسين التباين

Image Preprocessing and Contrast Enhancement

تلعب المعالجة المسبقة دوراً حاسماً في تحليل صور MRI لمرض الزهايمر نظراً لوجود انخفاض في التباين، وضوضاء، وعدم تجانس في شدة الإشارة. ولتعزيز البنى التشريحية ذات الأهمية التشخيصية وتحسين عملية استخراج الميزات، تم تطبيق خطوات معالجة لتحسين التباين قبل تدريب النماذج.

- تسوية الهستوغرام Histogram Equalization

تُعد تسوية الهستوغرام HE تقنية عالمية لتحسين التباين، حيث تعيد توزيع قيم شدة البكسلات لتغطي مجال التدرج الرمادي بشكل أكثر تجانساً. ومن خلال تحسين التباين العام، تعزز HE وضوح البنى التشريحية في الصور ذات المجال الديناميكي المحدود. إلا أن هذه التقنية قد تؤدي إلى تضخيم الضجيج وتحسين مفرط لمناطق غير ذات صلة. في هذه الدراسة، حيث تبين المعادلة (1) القانون العام لتوزيع الهستوغرام كما تم شرحها من قبل [15].

$$S_k = T(r_k) = (L - 1) \sum_{j=0}^k p(r_j) \quad (1)$$

حيث r_k : مستوى اللون الرمادي الأساسي و $T(r_k)$: دالة التحويل التي تقوم بتحويل مستوى الرمادي الأصلي إلى مستوى جديد. و S_k : مستوى اللون الرمادي بعد التحويل، $p(r_j)$: التوزيع الاحتمالي للمستوى الحر r_j و L : عدد مستويات التدرج الرمادي في الصورة.

Gaussian-Based Contrast Limited Adaptive Histogram Equalization -GCLAHE

للتغلب على القيود المرتبطة بتقنية تسوية الهستوغرام التقليدية HE، تم اعتماد تقنية CLAHE المعتمدة على التوزيع الغاوسي، والمعروفة بـ GCLAHE في البداية، جرى تطبيق مرشح غاوسي على الصور بهدف تقليل الضجيج عالي التردد مع الحفاظ على البنية التشريحية الأساسية للصورة. بعد ذلك، تم تطبيق خوارزمية CLAHE على مناطق محلية صغيرة تُعرف بالبلاطات Tiles، مما يسمح بتحسين تكيفي للتباين اعتماداً على توزيع شدة البكسلات في صور الرنين المغناطيسي للدماغ المرتبطة بمرض الزهايمر.

لتحديد مقدار القطع في الهستوغرام ومنع التضخيم المفرط للتباين، يتم حساب الحد الأقصى المسموح لتكرار القيم الرمادية وفق المعادلة (2) كما وضحت في [16]:

$$MAX_{CL} = \frac{clip_{limit} \times total_pixels_in_tile}{L} \quad (2)$$

حيث يشير MAX_{CL} : إلى الحد الأعلى المسموح لتكرار مستوى رمادي معين داخل الهستوغرام المحلي، بينما L : عدد مستويات التدرج الرمادي للصورة و $clip_{limit}$: معامل التحكم الذي يحدد مقدار الحد من التباين داخل كل بلاطة.

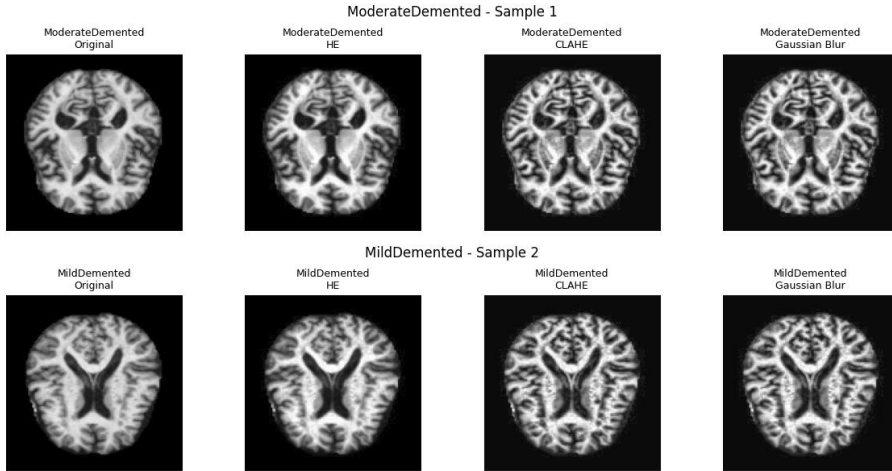
بعد تحديد هذا الحد، يتم تعديل التوزيع التكراري للهستوغرام بحيث لا تتجاوز أي قيمة التكرار الحد الأعلى المحدد. ويتم ذلك باستخدام عملية القص الموضحة في المعادلة (3) كما وضحت في [16]:

$$min(H, MAX_{CL}) = H^I \quad (3)$$

حيث تمثل H القيمة الأصلية للهستوغرام لمستوى رمادي معين، بينما تشير H^I إلى القيمة الجديدة بعد تطبيق عملية القص. وتعني هذه العملية أن أي قيمة تتجاوز الحد MAX_{CL} يتم تقليصها إلى هذا الحد، مما يمنع تضخيم التباين بشكل مفرط وتكمن أهمية هذا الإجراء:

1. يمنع تضخيم الضجيج في المناطق التي تحتوي على عدد كبير من البكسلات بنفس الشدة.
2. يحافظ على تفاصيل الصورة بشكل أفضل مقارنةً بـ HE.
3. يعطي تحكماً جزئياً في التباين بدلاً من التأثير على الصورة بأكملها.

يوضح الشكل (3) تسلسل مراحل معالجة البيانات، حيث يتم تطبيق خطوات التحسين تدريجيًا على الصور الطبية بهدف تحسين تباين الحواف والخصائص البنيوية لعدة فئات من مرض الزهايمر. وتساعد آلية تحديد التباين هذه في الحد من التضخيم المفرط للضجيج، مع إعادة توزيع القيم المقصودة بشكل متجانس للحفاظ على الاتساق الضوئي للصورة. إضافةً إلى ذلك، يتم استخدام الاستيفاء ثنائي الخلية Bilinear Interpolation بين البلاطات المجاورة لضمان انتقالات سلسلة بين المناطق المختلفة وتجنب ظهور آثار الحواف وقد أثبتت هذه المعالجة فعاليتها في إبراز التفاصيل التشريحية المهمة في صور الدماغ، مثل المادة الرمادية ومنطقة الحُصين، والتي تُعد مؤشرات مهمة في تحديد مراحل مرض الزهايمر [16].



الشكل (3) تطبيق HE و G-CLAHE على صورة رنين مغناطيسي للدماغ.

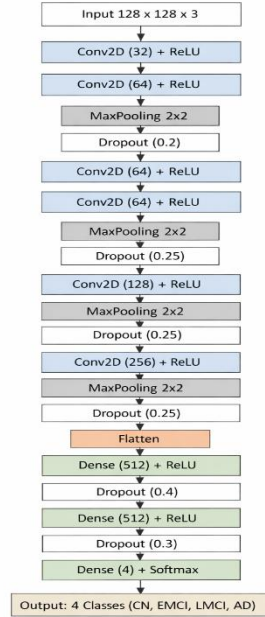
- التحليل النوعي لتقنيات تحسين الصور Qualitative Analysis وبالمقارنة البصرية والنوعية لتقنيات تحسين الصور المطبقة على عينات تمثيلية من صور MRI. تُظهر الصور الأصلية انخفاضًا في التباين وضعفًا في وضوح التفاصيل التشريحية. تعمل

تحليل صور الرنين المغناطيسي لمرض الزهايمر باستخدام تحسين التباين والشبكات العصبية

تسوية الهيستوغرام على تحسين التباين العام، إلا أنها قد تضخم الضجيج وتسبب تحسباً مفرطاً في بعض المناطق وبالمقابل، توفّر CLAHE تحسباً محلياً أكثر فاعلية للتباين مع الحفاظ على التفاصيل الدقيقة والتحكم في تضخيم الضجيج. ويساعد التموه الغاوسي في تقليل الضجيج الترددات العالية، إلا أنه قد يؤدي إلى فقدان حدة الحواف عند استخدامه بشكل منفصل. وبصورة عامة، يحقق النهج المقترح G-CLAHE توازناً فعالاً بين تحسين التباين وتقليل الضوضاء، مما يدعم استخراج ميزات أكثر موثوقة في مراحل التعلّم العميق اللاحقة.

- بنية الشبكة العصبية الالتفافية CNN Architecture

تم تصميم شبكة عصبية التلافية لاستخلاص ميزات عميقة تمييزية من صور MRI المحسنة. وتُعد CNN مناسبة بشكل خاص لتحليل الصور الطبية نظراً لقدرتها على تعلّم تمثيلات هرمية للميزات مباشرة من بيانات البكسل. وكما هو موضح في الشكل (4) ، تتكوّن البنية المقترحة من عدة طبقات التفاف مع دالة تنشيط ReLU، تليها طبقات Max-Pooling لتقليل الأبعاد المكانية والتعقيد الحسابي. وتُستخدم طبقات كاملة الاتصال لتجميع الميزات المستخرجة وتوليد تمثيلات مضغوطة لمهمة التصنيف النهائية. تم تدريب الشبكة باستخدام خوارزمية Adam بمعدل تعلّم 0.001، ودالة خسارة Sparse Categorical Cross-Entropy، ولمدة 50 دورة تدريبية وبحجم دفعة 32 صورة. كما استُخدم تقسيم للتحقق Validation Split لمراقبة قدرة التعميم أثناء التدريب [17].



الشكل (4): بنية الشبكة العصبية الالتفافية المقترحة.

- مصنف آلة متجه الدعم Support Vector Machine

تُعد آلة متجه الدعم SVM من خوارزميات التعلّم الخاضع للإشراف، حيث تبني حدود قرار من خلال تعظيم الهامش بين الفئات. وتُظهر فعالية عالية في فضاءات الميزات ذات الأبعاد المرتفعة، إلى جانب قدرة قوية على التعميم عند التعامل مع بيانات محدودة أو غير متوازنة [18].
تم هذه الدراسة استخدام SVM كمصنّف خارجي لتصنيف متجهات الميزات العميقة المستخرجة من CNN وتم اعتماد نواة دالة الأساس الشعاعي RBF لتمثيل العلاقات غير الخطية بين الميزات، مع اختيار المعاملات الفائقة تجريبياً لتحقيق أفضل أداء تصنيفي.

- الإطار الهجين CNN-SVM

لتحسين متانة التصنيف وحساسيته، تم اقتراح إطار هجين CNN-SVM، حيث تُستخدم CNN حصرياً لاستخلاص الميزات العميقة، بينما تتولى SVM مهمة التصنيف النهائي [19]. يستفاد من هذا النهج من قدرة CNN على تعلّم الميزات، ومن القوة التصنيفية العالية لـ SVM. ويُعد

الفصل بين مرحلتي استخراج الميزات والتصنيف مفيداً بشكل خاص في معالجة عدم توازن الفئات وتحسين الأداء في المراحل المرضية ضعيفة التمثيل.

- تدريب النموذج وتقسيم البيانات Model Training and Data Splitting

تم تقسيم البيانات إلى مجموعتي تدريب واختبار بنسبة 80:20 استُخدمت مجموعة التدريب لتحسين معاملات النموذج، بينما حُصصت مجموعة الاختبار لتقييم الأداء بشكل غير متحيز. ولم يُسمح بأي تداخل بين العينات لتجنب تسرب البيانات. كما تم تطبيق استراتيجيات التحقق المتقاطع أثناء التطوير لتعزيز المتانة والاستقرار.

- مقاييس التقييم Evaluation Metrics

تم تقييم أداء النماذج باستخدام مقاييس تصنيف متعددة الأصناف شائعة، شملت الدقة Accuracy، الدقة الإيجابية Precision، الاسترجاع Recall / Sensitivity، ومعيار F1- scour كما استُخدمت مصفوفات الالتباس لتحليل الأداء على مستوى كل فئة وأنماط الخطأ والتي تعتمد على أربع قيم أساسية هي True Positive-TP، و True Negative-TN، و False Positive-FP، و False Negative-FN يشير TP إلى عدد الحالات التي تنبأ النموذج بأنها إيجابية وكانت كذلك فعلياً، بينما يمثل TN عدد الحالات التي تنبأ النموذج بأنها سلبية وكانت صحيحة بالفعل. أما FP فيعبر عن الحالات التي تنبأ النموذج بأنها إيجابية رغم أنها سلبية في الواقع، في حين يشير FN إلى الحالات التي أخفق فيها النموذج في اكتشاف الحالة الإيجابية وتنبأ بأنها سلبية. إضافةً إلى ذلك، تم حساب منحنيات ROC والمساحة تحت المنحنى AUC لتقييم القدرة التمييزية للنماذج عبر عتبات تصنيف مختلفة. ونظرًا للأهمية السريرية للتشخيص المبكر، أُعطي اهتمام خاص لمقاييس الاسترجاع و F1 لتقليل الأخطاء السلبية الكاذبة كما وردت في [4, 6].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$precision = \frac{TP}{TP + FP} \quad (5)$$

$$recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1 - score = 2 * \frac{precision * recall}{precision + recall} \quad (7)$$

- النتائج والمناقشة Results and Discussion

يستعرض هذا القسم النتائج التجريبية التي تم الحصول عليها باستخدام الإطار المقترح، ويقدم مناقشة تفصيلية لأداء التصنيف. تم تقييم نموذجين مختلفين، هما: نموذج الشبكة العصبية الالتفافية المستقل CNN، والإطار الهجين آلة متجه الدعم CNN-SVM وتركز عملية التقييم على تصنيف مرض الزهايمر متعدد المراحل اعتماداً على صور الرنين المغناطيسي MRI، وذلك باستخدام عدة مقاييس أداء تشمل الدقة، والدقة الإيجابية، والاسترجاع، ومعيار F1-scoure، ومصفوفات الالتباس، وتحليل ROC-AUC.

- أداء نموذج الشبكة العصبية الالتفافية Standalone CNN

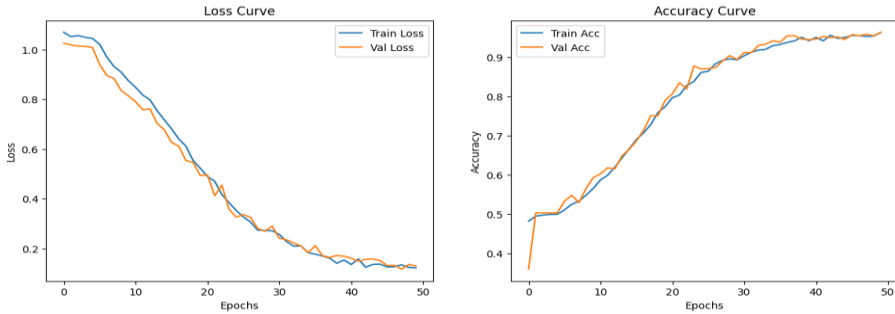
أظهر نموذج CNN المستقل قدرة عالية على تعلم تمثيلات تمييزية من صور الرنين المغناطيسي المحسنة بالتباين. وبعد تطبيق تقنيتي تسوية الهستوغرام HE والمعادلة التكميلية المحدودة على النقيض المعتمدة على التوزيع الغاوسي G-CLAHE، حقق النموذج دقة تصنيف كلية بلغت 96%، مما يدل على قدرة موثوقة في التمييز بين مراحل مرض الزهايمر المختلفة. وتلخص مقاييس الأداء الخاصة بكل فئة في الجدول (II) وتظهر النتائج قيماً مرتفعة ومتسقة لمقاييس الدقة والاسترجاع و F1-scour للفئات ذات التمثيل الكافي في البيانات. وعلى وجه الخصوص، حققت فئة غير مصاب بالخرف Non-Demented والخرف الخفيف جداً Very Mild Demented قيم F1 بلغت 0.94-0.97 على التوالي، مما يعكس حساسية النموذج للتغيرات المرضية المبكرة. كما أظهرت فئة الخرف الخفيف أداءً قوياً، مؤكدةً فعالية استخراج الميزات العميقة عند دمجها مع تحسين التباين. في المقابل، أظهرت فئة الخرف المتوسط Moderate Demented قيمة استرجاع أقل نسبياً بلغت 0.85، على الرغم من تحقيق دقة إيجابية كاملة. ويعكس هذا السلوك تأثير عدم توازن الفئات، حيث يحدّ العدد القليل من العينات من قدرة النموذج على اكتشاف جميع الحالات الحقيقية.

ومع ذلك، فإن غياب التنبؤات الإيجابية الخاطئة لهذه الفئة يشير إلى أن نموذج CNN تعلم ميزات عالية التخصص، وإن كان ذلك على حساب الحساسية.

الجدول: (II) أداء تصنيف نموذج CNN المقترح عبر مراحل مرض الزهايمر المختلفة.

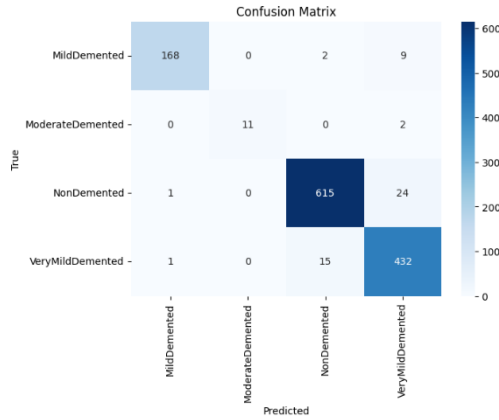
CLASS	Precision	Recall	F1-score	Support
MildDemented	0.99	0.94	0.96	179
ModerateDemented	1.00	0.85	0.92	13
NonDemented	0.97	0.96	0.97	640
VeryMildDemented	0.93	0.96	0.94	448
ACC			0.96	1280
Macro Avg	0.97	0.93	0.95	1280
Weighted Avg	0.96	0.96	0.96	1280

تُظهر منحنيات الخسارة والتحقق (الشكل 5) تقاربًا مستقرًا وفارقًا طفيفًا بين أداء التدريب والتحقق، مما يدل على عدم معاناة النموذج من فرط التكيف وقدرته الجيدة على التعميم على بيانات غير مرئية. ويؤكد ذلك فعالية استراتيجية المعالجة المسبقة والتدريب المعتمدة.



الشكل (5) منحنيات الخسارة والتدريب والدقة لنموذج الشبكة العصبية الالتفافية CNN .

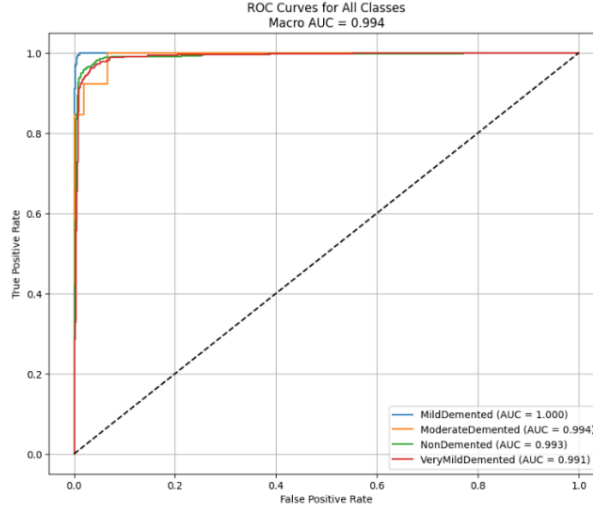
كما تؤكد مصفوفة الالتباس (الشكل 6) الأداء العالي للتصنيف عبر معظم الفئات. وتتركز حالات الخطأ أساسًا بين المراحل المرضية المتجاورة، مثل الخرف الخفيف جدًا والخرف الخفيف، وهو أمر متوقع نظرًا للطبيعة التدريجية والمستمرة للتدهور العصبي.



الشكل (6): مصفوفة الالتباس لنموذج الشبكة العصبية الالتفافية CNN المقترح لتصنيف مراحل مرض الزهايمر.

ويُظهر تحليل ROC-AUC الشكل 7 قدرة تمييزية عالية، حيث تجاوزت قيم 0.99 لجميع الفئات، مع متوسط كلي بلغ 0.994، مما يشير إلى توازن قوي بين الحساسية والخصوصية عبر مختلف عتبات التصنيف.

تحليل صور الرنين المغناطيسي لمرض الزهايمر باستخدام تحسين التباين والشبكات العصبية



الشكل (7): منحنيات خصائص التشغيل للمستقبل (ROC) وقيم المساحة تحت المنحنى (AUC) لتصنيف مرض الزهايمر متعدد الفئات باستخدام نموذج الشبكة العصبية الالتفافية (CNN).

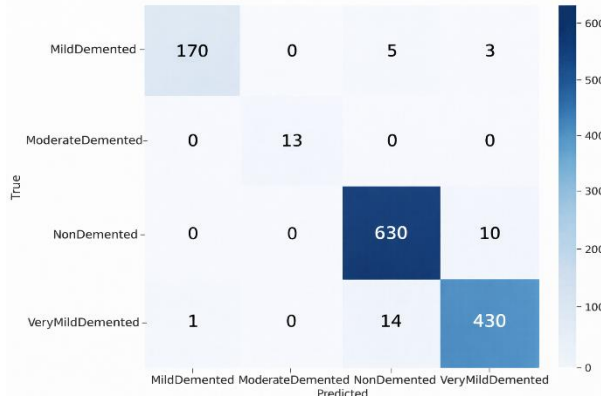
- أداء الإطار الهجين CNN-SVM

حقق الإطار الهجين CNN-SVM أداءً متفوقاً مقارنةً بنموذج CNN المستقل من أجل جميع مقاييس التقييم. فمن خلال استخدام CNN كمستخرج للميزات العميقة، وتكليف SVM بعملية التصنيف النهائية، بلغ النموذج الهجين دقة كلية قدرها 97% ومتوسط F1_score كلي بلغ 0.98. ويبين الجدول (III) نتائج النموذج الثاني.

الجدول (III): أداء تصنيف النموذج الهجين CNN-SVM.

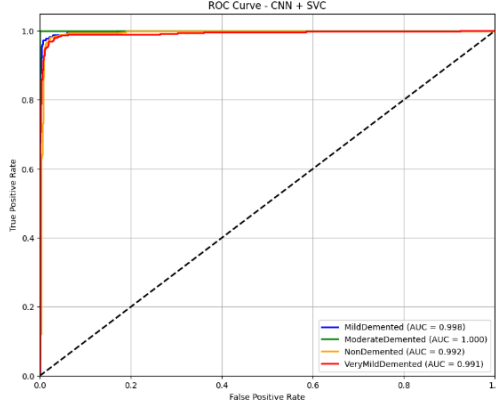
CLASS	PRECISIO	RECAL	F1-SCOR	SUPPOR
MildDemented	0.99	0.96	0.97	179
ModerateDemented	1.00	1.00	1.00	13
NonDemented	0.97	0.98	0.98	640
VeryMildDemented	0.97	0.97	0.97	448
ACC			0.97	1280
Macro Avg	0.98	0.98	0.98	1280
Weighted Avg	0.97	0.97	0.97	1280

ومن اللافت أن النهج الهجين حسّن بشكل كبير أداء فئة الخرف المتوسط ضعيفة التمثيل، وبرز هذا التحسن فعالية المصنّفات القائمة على تعظيم الهامش، مثل SVM، في التعامل مع عدم توازن الفئات عند العمل على تمثيلات ميزات عميقة عالية الأبعاد. وتُظهر مصفوفة الالتباس (الشكل 8) انخفاضًا واضحًا في معدلات الخطأ مقارنة بنموذج CNN المستقل، خاصة في المراحل المتقدمة من المرض. كما تُظهر معدلات عالية للتنبؤات الصحيحة لجميع الفئات، مع بقاء حالات الخطأ محصورة في المراحل المتجاورة، وهو ما يعكس التداخل السريري الطبيعي بين المراحل المرضية.



الشكل (8): مصفوفة الالتباس للإطار الهجين SVM-CNN في تصنيف مرض الزهايمر.

ويؤكد تحليل ROC-AUC (الشكل 9) متانة الإطار الهجين، حيث تراوحت قيم AUC بين 0.991 و 1.000، بمتوسط كلي بلغ 0.9954، مما يدل على سلوك تصنيفي أكثر اتساقاً مقارنة بالنموذج المستقل.



الشكل (9): تحليل منحنيات خصائص التشغيل للمستقبل (ROC) والمساحة تحت المنحنى (AUC) لتصنيف متعدد الفئات.

- التحليل المقارن بين نماذج CNN و CNN-SVM

يكشف التحليل المقارن بين النموذجين عن عدة ملاحظات رئيسية. فبينما يقدم نموذج CNN المستقل أداءً موثوقاً للفئات ذات التمثيل الكافي، تتخفف حساسيته عند التعامل مع الفئات محدودة البيانات. في المقابل، ينجح الإطار الهجين CNN-SVM في معالجة هذا القيد بفضل خاصية تعظيم الهامش في SVM، مما يؤدي إلى تحسين الاسترجاع والامتانة العامة. ومن منظور سريري، يُعد الاسترجاع مقياساً بالغ الأهمية، إذ قد تؤدي الأخطاء السلبية الخاطئة إلى تأخير التشخيص والتدخل العلاجي. وقد أظهر النموذج الهجين تحسناً متسقاً في الاسترجاع عبر جميع الفئات، لا سيما في المراحل المتقدمة من المرض، مما يجعله أكثر ملاءمة لأنظمة دعم القرار السريري.

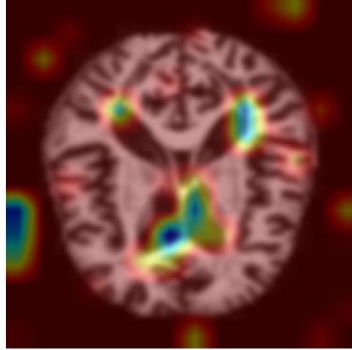
- تأثير تحسين التباين

تعزز تقنيات تحسين التباين أداء التصنيف. فقد أسهمت G-CLAHE في تحسين وضوح البنى التشريحية الدقيقة. وساعد تحسين التباين في تسهيل تعلم الميزات العميقة، مما انعكس إيجابياً على أداء كلا النموذجين. وبالمقارنة مع تسوية الهيستوغرام، قدّمت تقنية G-CLAHE المقترحة تحسناً أكثر توازناً من خلال تعزيز التباين المحلي مع الحد من تضخيم الضجيج، وهو أمر بالغ الأهمية في تحليل صور MRI حيث يتطلب التشخيص الدقيق الحفاظ على التفاصيل البنيوية الدقيقة.

- قابلية تفسير النموذج باستخدام Grad-CAM++

لتعزيز الشفافية والثقة السريرية، تم استخدام Grad-CAM++ لتصوير المناطق الدماغية التي تسهم في تنبؤات النموذج. وقد أظهرت خرائط التنشيط تركيزًا واضحًا على مناطق تشريحية ذات صلة سريرية، مثل الحُصين والفصوص الصدغية الإنسية (الشكل 10)، وهي مؤشرات حيوية معروفة لمرض الزهايمر.

Predicted: MildDemented | True: MildDemented



الشكل (10) تصوير Grad-CAM للمناطق التمييزية في حالة الخرف الخفيف.

ويشير هذا التوافق بين انتباه النموذج والأنماط العصبية المرضية المعروفة إلى أن الإطار المقترح يتعلم ميزات ذات معنى سريري حقيقي، وليس مجرد ارتباطات عشوائية. كما يعزز دمج تقنيات الذكاء الاصطناعي القابل للتفسير من قابلية اعتماد النموذج في البيئات السريرية [20,21].

- المناقشة في سياق الأعمال ذات الصلة

يوضح الجدول (IV) مقارنة بين عدد من الدراسات الحديثة التي تناولت تشخيص مرض الزهايمر باستخدام تقنيات الذكاء الاصطناعي والتعلم العميق، إضافةً إلى النماذج المقترحة في هذه الدراسة. وتهدف هذه المقارنة إلى إبراز موقع النموذج المقترح ضمن السياق البحثي الحالي، من حيث نوع البيانات المستخدمة، وبنية النموذج، ومستوى الأداء المحقق. كما تبيّن النتائج أن العديد من الدراسات الحديثة تعتمد على دمج مصادر بيانات متعددة مثل صور MRI و PET والبيانات السريرية أو الجينية، الأمر الذي يسهم غالبًا في تحسين دقة التنبؤ لكنه يزيد من تعقيد النموذج ومتطلبات المعالجة.

وتظهر المقارنة أن النموذج المقترح في هذه الدراسة يحقق توازنًا مناسبًا بين الدقة والمانعة والكفاءة الحسابية. فعلى الرغم من أن بعض الدراسات تعتمد على نماذج متعددة الوسائط أو شبكات ثلاثية

تحليل صور الرنين المغناطيسي لمرض الزهايمر باستخدام تحسين التباين والشبكات العصبية

الأبعاد معقدة، فإن النموذج المقترح يعتمد على بيانات MRI فقط مع تطبيق تقنيات تحسين الصور، مما يسمح بالحصول على أداء تنافسي مع الحفاظ على بساطة البنية وقابلية التطبيق العملي. ويُعد هذا الجانب مهماً في التطبيقات الطبية الواقعية، حيث إن تقليل تعقيد النظام ومتطلبات البيانات يسهم في تسهيل دمج النموذج ضمن أنظمة الدعم السريري واتخاذ القرار الطبي. كما تُظهر النتائج أن النموذج الهجين المقترح CNN + SVM يحقق تحسناً ملحوظاً مقارنة بالنموذج الأساسي المعتمد على CNN فقط، خاصة في التعامل مع الفئات الأقل تمثيلاً في البيانات. ويشير ذلك إلى أن الجمع بين تقنيات التعلم العميق وأساليب التعلم الآلي التقليدية يمكن أن يوفر أداءً أكثر استقراراً في مهام التصنيف الطبي.

الجدول (IV): مقارنة بين الدراسات الحديثة وأساليب الكشف عن مرض الزهايمر والنماذج المقترحة.

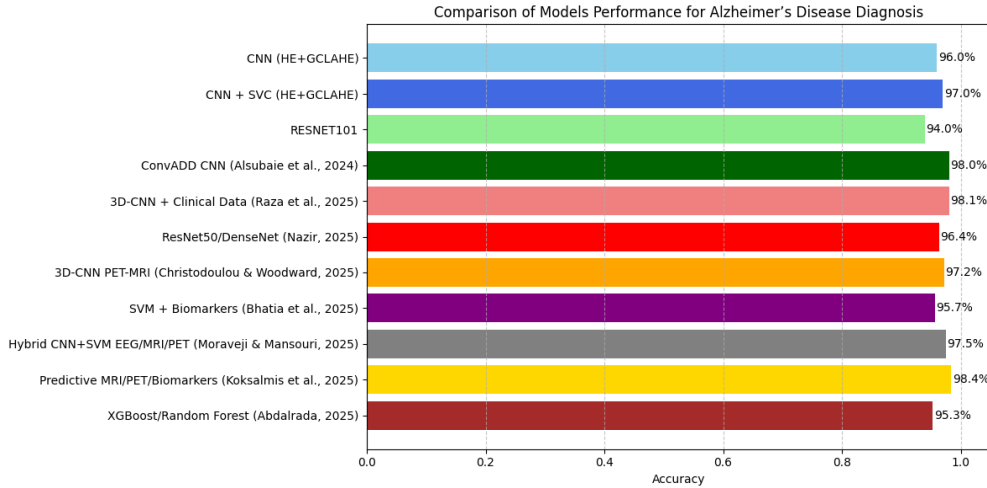
Study / Model	Data Type	Model/Tech nique	Acc(%)	Remarks
Raza et al., 2025	PET + MRI + Clinical	Multimodal 3D-CNN	98.1	Integration of genetic data improved predictive performance
Nazir, 2025	MRI + PET	ResNet-50 and DenseNet	96.4	Effective discrimination between early and advanced disease stages
Christodoulou & Woodward, 2025	PET-MRI + Tauvid	3D-CNN	97.2	Differentiation between Alzheimer's disease and mild cognitive impairment
Bhatia et al., 2025	CSF + PET	SVM with mass	95.7	Focus on molecular and

		spectrometry analysis		neurochemical biomarkers
Awang et al., 2025	MRI + fMRI	Deep CNN	96.8–98.3	Conducted within the ADNI framework
Moraveji & Mansouri, 2025	EEG + MRI + PET	Hybrid CNN–SVM	97.5	Incorporation of explainable AI (XAI) techniques
Koksalmis et al., 2025	MRI + PET + Fluid + Genomic	Multilayer predictive model	98.4	Support for personalized disease progression prediction
Ehteshamzad, 2024	EEG + Genetic data	Not specified	94.6	Non-invasive diagnostic tools
Abdalrada, 2025	MRI + CSF + Neuropsychological tests	XGBoost and Random Forest	95.3	A combination of deep learning and traditional methods
Alsubaie et al., 2024	MRI	ConvADD CNN	98.01	Alsubaie et al., 2024
Proposed Model I	MRI	CNN	96.0	Baseline model with G-CLAHE and HE preprocessing
Proposed Model II	MRI	CNN + SVM	97.0	Hybrid model with improved performance for underrepresented classes

تحليل صور الرنين المغناطيسي لمرض الزهايمر باستخدام تحسين التباين والشبكات العصبية

كما يوضح الشكل (11) مقارنة بصرية لدقة التصنيف بين النماذج المقترحة وعدد من الأساليب الحديثة، حيث يتبين أن نموذج CNN-SVM المقترح يظل منافسًا بقوة رغم اعتماده على وسيلة تصوير واحدة فقط.

وتؤكد هذه النتائج أن الإطار المقترح يوفر حلاً عملياً وقابلًا للتوسع لتشخيص مرض الزهايمر، من خلال تقليل الاعتماد على البيانات متعددة الوسائط مع الحفاظ على أداء تشخيصي قوي ومتوازن.



الشكل(11) : مقارنة دقة التصنيف بين النماذج المقترحة وعدد من الأساليب الحديثة في تشخيص مرض الزهايمر.

- القيود Limitations

على الرغم من النتائج الواعدة التي حققتها هذه الدراسة، إلا أنه يجب الإقرار بعدة قيود. أولاً، تعاني مجموعة البيانات من عدم توازن بين الفئات، لا سيما في المراحل المتقدمة من المرض، مما قد يؤثر على قدرة النموذج على التعميم، على الرغم من التخفيف الجزئي لهذا الأثر من خلال اعتماد التصنيف الهجين. ثانياً، يعتمد التحليل على مقاطع ثنائية الأبعاد من صور الرنين المغناطيسي بدلاً من البيانات الحجمية الكاملة، الأمر الذي قد يحد من الاستفادة من المعلومات البنيوية ثلاثية الأبعاد للدماغ. وأخيراً، لا تتضمن مجموعة البيانات معلومات طولية أو بيانات سريرية، مما يقيد إمكانية نمذجة تطور المرض عبر الزمن.

4. الاستنتاجات Conclusions

قدّمت هذه الدراسة إطاراً فعالاً ومنتيناً لتصنيف مرض الزهايمر متعدد المراحل باستخدام التصوير بالرنين المغناطيسي MRI وتقنيات التعلّم العميق. ومن خلال دمج تحسين التباين مع استخلاص الميزات العميقة واستراتيجيات التصنيف الهجين، عالجت المنهجية المقترحة عدداً من التحديات الرئيسية في تشخيص مرض الزهايمر اعتماداً على صور MRI، بما في ذلك انخفاض تباين الصور، والفروق الدقيقة بين الفئات المرضية، وعدم توازن البيانات.

أظهرت النتائج التجريبية أن دمج تقنيات تحسين الصور، ولا سيما G-CLAHE، أسهم بشكل ملحوظ في تحسين وضوح البنى الدماغية ذات الأهمية التشخيصية، وعزّز قابلية تمييز الميزات العميقة. وقد حقق نموذج الشبكة العصبية الالتقافية المستقل CNN دقة تصنيف كلية بلغت 96%، مما يؤكد فعالية المعالجة المسبقة المحسّنة بالتباين عند دمجها مع استخلاص الميزات المعتمد على التعلّم العميق.

علاوة على ذلك، تفوّق الإطار الهجين المقترح CNN+SVM بشكل متسق على نموذج CNN المستقل، محققاً دقة كلية بلغت 97%، ومتوسط F1 كلي قدره 0.98، ومتوسط ROC-AUC بلغ 0.9954. وقد أظهر النهج الهجين متانة وحساسية أعلى، لا سيما في المراحل المرضية ضعيفة التمثيل، وتؤكد هذه النتائج أن الإطار المقترح يحقق توازناً ملائماً بين أداء التصنيف، والمتانة، والكفاءة الحسابية.

كما أسهم دمج تقنيات الذكاء الاصطناعي القابل للتفسير باستخدام Grad-CAM++ في تعزيز شفافية النموذج وأهميته السريرية. إذ أبرزت خرائط التنشيط مناطق تشرحية ذات معنى سريري واضح، مثل الحُصين والفصوص الصدغية الإنسية، المعروفة بتأثرها المبكر في مرض الزهايمر. ويعزز هذا التوافق بين تركيز النموذج والأنماط العصبية المرضية المعروفة قابلية تفسير النظام وإمكانية اعتماده سريريّاً.

وبوجه عام، تشير النتائج إلى أن الإطار المقترح المعتمد على صور MRI يوفّر حلاً موثوقاً وعملياً لتصنيف مراحل مرض الزهايمر باستخدام وسيلة تصوير واحدة فقط. وبالمقارنة مع المقاربات متعددة الوسائط الأكثر تعقيداً، يحقق هذا النهج أداءً تنافسياً مع متطلبات حسابية أقلّ وعمليات جمع بيانات أبسط.

5. الأعمال المستقبلية Future Work

يمكن استكشاف عدة اتجاهات لتطوير الإطار المقترح مستقبلاً. أولاً، يمكن دراسة دمج وسائل تصوير إضافية، مثل التصوير المقطعي بالإصدار البوزيتروني PET أو التصوير بالرنين المغناطيسي الوظيفي fMRI، إلى جانب المؤشرات الحيوية السريرية أو الجينية، بهدف تحسين دقة التشخيص وتقديم توصيف أكثر شمولية للمرض. ثانياً، قد يتيح توسيع المنهجية لتشمل شبكات CNN ثلاثية الأبعاد وتحليل صور MRI الحجمية الاستفادة من معلومات مكانية أغنى، مما قد يؤدي إلى تحسين إضافي في الأداء.

إضافة إلى ذلك، يمكن في الدراسات المستقبلية استكشاف تقنيات متقدمة لزيادة البيانات وإعادة توازن الفئات للتقليل من تأثير عدم التوازن، لا سيما في المراحل المتقدمة من المرض. كما أن دمج بيانات MRI الطولية قد يتيح نمذجة تطور المرض بمرور الزمن، ودعم التنبؤ المبكر بالتدهور المعرفي.

6. المراجع

- [1] Alzheimer's Association. (2025). 2025 Alzheimer's disease facts and figures. *Alzheimer's & Dementia*, 21(5), 1–XX. <https://alz-journals.onlinelibrary.wiley.com/doi/10.1002/alz.70235>
- [2] Rohani, M. M., Sharifi, S., & Durson, S. (2025). Deep learning in medical imaging for disease diagnosis. *World Journal of Advanced Research and Reviews*, 25(2), 2522–2526. <https://journalwjarr.com/node/772>
- [3] Zhou, J., Park, S., Dong, S., Tang, X., & Wei, X. (2025). Artificial intelligence-driven transformative applications in disease diagnosis technology. *Medical Review*, Advance online publication. <https://doi.org/10.1515/mr-2024-0097>
- [4] Sarraf, S., & Tofghi, G. (2019). DeepAD: Alzheimer's disease classification via deep convolutional neural networks using MRI and fMRI. *Cognitive Systems, Research*, 57, 147–159. <https://doi.org/10.1016/j.cogsys.2019.04.007>
- [5] Sarraf, S., & Tofghi, G. (2016). Classification of Alzheimer's Disease using fMRI Data and Deep Learning Convolutional Neural Networks. *arXiv preprint arXiv:1603.08631*. <https://arxiv.org/abs/1603.08631>
- [6] Raza, M. L., Hassan, S. T., Jamil, S., & Hyder, N. (2025). Advancements in deep learning for early diagnosis of Alzheimer's disease using multimodal neuroimaging: challenges and future

- directions. Frontiers in Neuroinformatics. <https://www.frontiersin.org/journals/neuroinformatics/articles/10.3389/fninf.2025.1557177/full>
- [7] Nazir, S. (2025). The future of Alzheimer's disease risk prediction: a systematic review. Neurological Sciences. <https://link.springer.com/article/10.1007/s10072-025-08167-x>
- [8] Christodoulou, R. C., & Woodward, A. (2025). Artificial Intelligence in Alzheimer's Disease Diagnosis and Prognosis Using PET-MRI: A Narrative Review of High-Impact Literature Post-Tauvid Approval. Frontiers in Aging Neuroscience (PMC). <https://pmc.ncbi.nlm.nih.gov/articles/PMC12387124/>
- [9] Bhatia, V., Chandel, A., Minhas, Y., & Kushawaha, S. K. (2025). Advances in biomarker discovery and diagnostics for Alzheimer's disease. Neurological Sciences. <https://link.springer.com/article/10.1007/s10072-025-08023-y>
- [10] Awang, M. K., Ali, G., & Faheem, M. (2025). Recent Advancements in Neuroimaging-Based Alzheimer's Disease Prediction Using Deep Learning Approaches in e-Health: A Systematic Review. Health Science Reports. <https://onlinelibrary.wiley.com/doi/abs/10.1002/hsr2.70802>
- [11] Moraveji, M., & Mansouri, N. (2025). Recent Advances in Computational and Machine-Learning Approaches for Alzheimer's Disease Classification: A Comprehensive Review. Archives of Computational Methods in Engineering. <https://link.springer.com/article/10.1007/s11831-025-10476-5>
- [12] Koksalmis, G. H., Soykan, B., & Brattain, L. J. (2025). Artificial Intelligence for Personalized Prediction of Alzheimer's Disease Progression: A Survey of Methods, Data Challenges, and Future Directions. arXiv preprint. <https://arxiv.org/abs/2504.21189>
- [13] Ehteshamzad, S. (2024). Assessing the Potential of EEG in Early Detection of Alzheimer's Disease: A Systematic Comprehensive Review (2000–2023). Alzheimer's Disease Research Journal. <https://journals.sagepub.com/doi/abs/10.3233/ADR-230159>
- [14] Abdalrada, A. S. (2025). Comprehensive Review of Machine Learning Approaches for Alzheimer's Disease Diagnosis and

- Prognosis. International Journal of Emerging Artificial Intelligence Applications. <https://ijeaa.cultechpub.com/index.php/ijeaa/article/view/25>
- [15] Chowdhury, J. H., et al. (2024). Simple Histogram Equalization Technique Improves Performance of VGG Models on Facial Emotion Recognition. *Algorithms*, 17(6), 238. <https://doi.org/10.3390/a17060238>
- [16] Namazi Nia, S., & Shih, F. Y. (2024). Medical X-Ray Image Enhancement Using Global Contrast-Limited Adaptive Histogram Equalization. *arXiv:2411.01373*. <https://arxiv.org/abs/2411.01373>
- [17] Zhou, Y., Zhang, M., Wang, T., & Li, H. (2024). A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*. <https://doi.org/10.1007/s10462-024-10721-6>
- [18] Guido, R., Ferrisi, S., Lofaro, D., & Conforti, D. (2024). An Overview on the Advancements of Support Vector Machine Models in Healthcare Applications: A Review. *Information*, 15(4), 235. <https://doi.org/10.3390/info15040235>
- [19] Anwar, S., Rehman, A., et al. (2024). Hybrid CNN-SVM Model for Enhanced Early Detection of Chronic Kidney Disease. *Alexandria Engineering Journal*. <https://www.sciencedirect.com/science/article/abs/pii/S174680942401142X>
- [20] Chattopadhyay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018). Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*.
- [21] Ahmed, S., Lee, J., & Park, S. (2024). Explainable Deep Learning for Alzheimer's Disease Classification Using Grad-CAM++ on MRI Images. *Journal of Medical Imaging and Health Informatics*, 14(3), 213-223. <https://doi.org/10.1166/jmihi.2024.23312>.
- [22] Alsubaie, M. G., Luo, S., & Shaukat, K. (2024). ConvADD: Exploring a Novel CNN Architecture for Alzheimer's Disease Detection. *International Journal of Advanced Computer Science and Applications*, 15(4), 300–313. <https://doi.org/10.14569/IJACSA.2024.0150431>

التنبؤ بحركة المرور ضمن المدن الذكية باستخدام خوارزميات التجميع والتعلم العميق

م. خالد قندقجي د. ماهر عباس د. أسماء شعاع

المخلص

تعد المدن الذكية من أهم تطبيقات انترنت الأشياء، حيث تتميز بتنوعها فهي عبارة عن تكامل مجموعة من التطبيقات المختلفة كنظم التنقل الذكية، نظم كشف التلوث وغيرها. تساهم نظم التنقل الذكية وبمساعدة تقنيات الذكاء الصناعي في تنظيم حركة المرور واكتشاف الازدحام مما يوفر المعلومات اللازمة لسائقي المركبات والمشاة في معرفة حالة الطرق، الحوادث الموجودة واختيار المسارات الأفضل للتنقل. قمنا في هذا البحث باستخدام مجموعة البيانات المقدمة من مشروع citypulse وذلك بهدف تطوير نموذج هجين يستخدم خوارزميات التجميع لتحديد الطرق المتشابهة بحركة المرور ومن ثم استخدام البيانات الناتجة في عملية التنبؤ، وبيّنت الدراسة تفوق خوارزمية المحولات على غيرها من خوارزميات التعلم العميق الأخرى، فيما تساهم خوارزميات التجميع في تحسين الدقة، تقليل الزمن وتخفيف الكلفة الحسابية اللازمة لبناء النماذج اللازمة للتنبؤ بحركة المرور المستقبلية من خلال اكتشاف نماذج الحركة بين الطرق المختلفة. وأظهرت خوارزمية DBSCAN أداءً أفضل من باقي الخوارزميات الأخرى في عملية تحسين التنبؤ لاعتمادها على تجميع البيانات بالاعتماد على الكثافة وقدرتها على فترة النقاط الشاذة مما يحسن عمل أداء خوارزميات التعلم العميق.

الكلمات المفتاحية: حركة المرور، التنبؤ، المدن الذكية، خوارزميات التجميع، تحليل البيانات.

م. خالد قندقجي طالب دراسات عليا (دكتوراه) كلية الهندسة المعلوماتية قسم هندسة الشبكات والنظم الحاسوبية جامعة حمص

د. ماهر عباس استاذ مساعد في كلية الهندسة المعلوماتية قسم هندسة الشبكات والنظم الحاسوبية في جامعة حمص

د. أسماء شعار مدرس في كلية الهندسة المعلوماتية قسم هندسة البرمجيات ونظم المعلومات
جامعة حمص

Traffic Flow Forecasting in Smart Cities using Clustering and Deep Learning Algorithms

Abstract

Smart cities represent one of the most significant applications of the Internet of Things (IoT), characterized by their diversity and integration of multiple systems, such as intelligent transportation and pollution detection systems. Intelligent transportation systems with help of artificial intelligence (AI) techniques play a crucial role in traffic regulation and congestion detection, providing real-time information to drivers and pedestrians about road conditions, accidents, and optimal route selection.

In this study, we used a dataset provided by the CityPulse project to develop a hybrid model that employs clustering algorithms to identify roads with similar traffic patterns and leverages their data for predictive modeling. Our findings demonstrate that Transformer-based algorithms outperform other Deep Learning methods, while clustering techniques contribute to improving accuracy, reducing computational time, and lowering the cost of building traffic flow prediction models by uncovering movement patterns across different roadways.

Notably, the DBSCAN algorithm exhibited superior performance in enhancing prediction accuracy compared to other clustering methods. This can be attributed to its density-based clustering approach and its ability to

filter out outliers, thereby improving the effectiveness of deep learning models in traffic flow forecasting.

KEYWORDS: traffic flow, predicting, smart cities, clustering algorithms, data analysis.

1- مقدمة:

أدى الانفجار السكاني الهائل لزيادة متطلبات وسائل النقل والتي بدورها أدت للكثير من المشاكل كالازدحام المروري، الحوادث وزيادة التلوث. يؤثر الازدحام المروري بشكل عام على حركة المرور حيث يُسبب التأخير وإضاعة الوقت بالإضافة لتعطيل عمل بعض المركبات الأخرى كسيارات الشرطة والإسعاف، مما يؤدي إلى خسائر بشرية ومادية للدول تصل لمعدلات عالية حيث وصلت لـ 80% لدى البعض [2].

ساهم ظهور شبكات انترنت الأشياء (Internet of Things) بتغيير نمط حياة الأفراد حيث باتت هذه الشبكات جزءاً لا يتجزأ في مختلف تطبيقات الحياة. حيث تستخدم في المدن الذكية (Smart Cities)، المجال الصحي وغيرها [30]. ومما ساعد على انتشار هذه الشبكات ظهور الذكاء الصناعي وتطبيقاته القادرة على تحليل البيانات الناتجة عن هذه الشبكات [4]. تعد نظم التنقل الذكية (Intelligent Transportation Systems) ITS من أهم تطبيقات شبكات انترنت الأشياء حيث تستفيد هذه النظم من البيانات التي تولدها هذه الشبكات من خلال تحليلها وذلك باستخدام تقنيات تعلم الآلة (Machine Learning) والتعلم العميق (Deep Learning) لتأمين متطلبات التنقل الآمن عن طريق تحديد المسار الأفضل، إخطار السائقين بالحوادث والازدحام المروري، تنظيم إشارات المرور، تزويد السائقين بمعلومات تخص مواقف السيارات [14]، اكتشاف الشذوذ في حركة المرور [19] وغيرها من التطبيقات الأخرى [9]. حيث تستفيد هذه النظم من

شبكات انترنت الأشياء لتحصيل البيانات المختلفة ضمن المدينة كعدد المركبات، سرعة الحركة، حالة الطقس، التلوث ضمن الجو، معلومات مواقف السيارات وغيرها.

2- مشكلة البحث:

تتأثر حركة المرور ضمن طريق ما بمجموعة من العوامل كحركة الطرق الأخرى، سرعة حركة المركبات، التلوث وغيرها. الأمر الذي يجعل دراسة تأثير هذه الخواص أمراً هاماً في عملية التنبؤ. تتطلب نظم التنقل الذكية نماذج ذات دقة تنبؤ عالية من جهة وذات كلفة حسابية أقل من جهة أخرى الأمر الذي يتطلب بناء نماذج هجينة تقلل كلفة العمليات الحسابية الخاصة بالنماذج اللازمة للتنبؤ بحركة المرور. وذلك بهدف تحديد أوقات الذروة والفراغ ضمن الطرق المدروسة بهدف تحسين حركة المرور ضمن المدينة.

3- هدف البحث:

يهدف البحث إلى دراسة تأثير حركة الطرق المتشابهة على عملية التنبؤ في حركة المرور ضمن الطريق المدروس، ومن ثم دمج النماذج السابقة مع خوارزميات التجميع بهدف إنشاء نماذج أكثر دقة في عملية التنبؤ وأخف كلفة حسابية تتناسب مع بيئة تطبيقات انترنت الأشياء.

4- أهمية البحث:

تساعد نظم التنقل الذكية في تقديم العديد من الفوائد التي يمكننا الاستفادة منها كعملية ضبط زمن إشارات المرور، تنظيم حركة المركبات ضمن المدن وتخفيف الازدحام ومساعدة الجميع للوصول للهدف بأسرع وقت. تساعد عملية الاختيار الأمثل لطرق استقبال البيانات من الحساسات، معالجة وتخزين البيانات وتحليلها على تقديم هذه الوظائف بسهولة ويسر مما يساهم في تحقيق الاستفادة من هذه النظم.

5- السلاسل الزمنية (Time Series):

تُعرف السلسلة الزمنية بأنها مجموعة من نقاط البيانات المرتبة زمنياً، بتعبير آخر هي عبارة عن تسلسل زمني من العينات التي تخص المتحول المدروس [13]، يتم تسجيل هذه النقاط خلال فواصل زمنية متساوية (كل دقيقة، ساعة، يوم...). وتستخدم في الكثير من التطبيقات اليومية في حياتنا كتسجيل استهلاك الطاقة الكهربائية، مراقبة درجة حرارة الجو وغيرها الكثير. [13]. وتتكون السلاسل الزمنية بشكل عام من المركبات التالية:

- 1- قيم السلسلة الزمنية: تمثل السلسلة الزمنية المسجلة.
- 2- الاتجاه (Trend): يمثل التغير الحاصل في السلسلة الزمنية ويمكن أن يكون يتصاعد أو يتنازل حسب طبيعة السلسلة المدروسة.
- 3- الفصلية أو الموسمية (Seasonality): تلتقط هذه المركبة التغيرات الدورية للسلسلة خلال فواصل زمنية ثابتة.
- 4- الخطأ (Residual): تعبر عن الأخطاء الحاصلة وتسمى بـ (White Noise). وهذه المركبة لا يمكن استخدامها كدخل لعمليات التنبؤ، التحليل.

6- الدراسات المرجعية:

6-1 النماذج التقليدية المعتمدة على خوارزميات الانحدار (Regression):

قام الباحثون في [14, 18] بالتنبؤ بحركة المرور المستقبلية باستخدام خوارزميات الانحدار وقدمت هذه الدراسات معدل خطأ مرتفع من أجل التنبؤ بالحركة فيما حسنت الدراسة [18] معدل الخطأ الناتج في بعض الخوارزميات بإضافة خواص أخرى للتنبؤ كسرعة الحركة التي تساهم أيضاً في تقليل حجم النافذة للحصول على نماذج أفضل وبكلفة حسابية أقل.

6-2 نماذج التعلم العميق المعتمدة على الشبكات المتكررة (RNN):

توجد العديد من الدراسات التي تستخدم هذه الفئة من الشبكات في تحليل السلاسل الزمنية كما في [2] التي اقترحت نموذجاً للتنبؤ بالتلوث ضمن المدن الذكية ويحتوي النموذج المقترح على عدد من الطبقات التي تبدأ بمعالجة البيانات ثم مجموعة من طبقات شبكة الذاكرة الطويلة قصيرة الأمد (LSTM) والتي تدرس العلاقة بين سلاسل الدخل وأخيراً طبقة الخرج التي تمثل قيم التنبؤ. حقق النموذج السابق تحسناً على دقة التنبؤ من خلال تقليل الخطأ الناتج بنسبة وصلت لـ 36% من أجل معيار RMSE عن نماذج RNN التقليدية، لكن يعاني النموذج المقترح من كلفة حسابية عالية حيث يحتاج لضعف زمن تنفيذ خوارزمية RNN وهو غير مناسب مع طبيعة تحليل بيانات شبكات انترنت الأشياء. واقترح الباحثون في [3] دراسة مجموعة من النماذج المستخدمة في التنبؤ بالقيم المستمرة كشبكات LSTM، الغابات العشوائية (Random Forest) بالإضافة لخوارزمية التعبئة (Bagging) والتي حققت أقل معدل خطأ من باقي الخوارزميات السابقة، تعاني النماذج المستخدمة من قيم خطأ عالية كما تم استخدام الخوارزميات السابقة بشكل عام دون ضبط قيم البارامترات لتقليل معدل الخطأ.

6-3 نماذج أخرى معتمدة على خوارزميات التجميع، التعلم العميق أو كليهما:

واقترح الباحثون في [4, 5] نظام تنقل ذكي معتمد على مرحلتين الأولى استخدام خوارزميات التجميع لتحديد الطرق المتشابه بشكل ديناميكي حيث قارنت الدراسة بين خوارزميات K-medoids، التجميع الهرمي، التجميع المعتمد على الكثافة والانتشار التقريبي التي أظهرت أداءً أفضل من الخوارزميات السابقة، ومن ثم قامت الدراسة السابقة بدراسة تأثير التنبؤ بحركة المرور ضمن طريق بالاعتماد على شبكات LSTM بالاعتماد على عملية التجميع السابقة وأظهرت الدراسة تحقيق معدل خطأ قليل. لم تقم الدراسة السابقة بالتنبؤ بحركة المرور بشكل واسع واقتصرت على بيانات يوم واحد للاختبار كما ويجعل الأسلوب الديناميكي لهذا النموذج مشكلة خاصة مع استخدام شبكات انترنت الأشياء التي تعاني في بعض الحالات من مشاكل في الاتصال. واقترح

الباحثون في [6] نموذجاً للتنبؤ بحركة المرور بالاعتماد على تجميع الطرق ضمن بيان فرعي بالاعتماد على زمن التنقل بين الطرق والطريق الأقصر ومن ثم استخدام هذه البيانات مع الزمان والمكان لتنفيذ عملية التنبؤ. يعاني النموذج السابق من التعقيد ولا يحقق معدل خطأ قليل مقارنة بدراسات أخرى. وقام الباحثون في [7] بمقارنة أداء مجموعة من خوارزميات التجميع على السلاسل الزمنية باستخدام عوامل تشابه كالمسافة الإقليدية والفرق الزمني الديناميكي (Dynamic Time Warping) وأظهرت الدراسة تحقيق عدد عناقيد أقل من الدراسة [4] حيث تم تجميع البيانات على شكل فواصل زمنية مختلفة على شكل أيام، ساعات، و 30 دقيقة وركزت الدراسة على خوارزميات التجميع فقط. واقترح الباحثون في [8] بناء نظام تنقل ذكي معتمد على خوارزميات التجميع والتنبؤ بحركة المرور باستخدام مزيج ما بين الشبكات العصبونية التلافيفية CNN (Convolutional Neural Networks) وشبكات LSTM، تستفيد الدراسة من استخدام التعلم المتقل (Transfer Learning) لتخفيف عبء التدريب. ولا يحقق النموذج المقترح نتائج تنبؤ جيدة من أجل معدل الخطأ. يوضح الجدول [1] ملخص التقنيات ومجموعات البيانات المستخدمة في الدراسات السابقة:

الجدول (1) ملخص الدراسات المرجعية السابقة

المؤلفين	الخواص المستخدمة	الخوارزميات المستخدمة	سلبيات النموذج
Shouaib et al [14]	الغازات المنبعثة (pollution)	Regression + DL	مكلف جداً بسبب حجم النافذة الكبير للتنبؤ
Kondakji et al [18]	الغازات المنبعثة + سرعة الحركة	Regression only	أخف من الدراسة السابقة بسبب استخدام خواص أخرى في عملية التنبؤ.

مكلف جداً.	RNN only	الغازات المنبعثة	Bello et al [2]
مكلف جداً ودقة قليلة	Regression + LSTM	عدد مركبات الطريق المدروس	Alkarim et al [3]
تعتمد على القيم المتولدة من شبكات IoT بشكل لحظي مما قد يعطل عمل النموذج عند حدوث فقدان بيانات	Clustering + LSTM	عدد مركبات الطرق المتشابهة	Wang [4,5]
معقد ومكلف جداً ويحقق أداءً ضعيفاً مقارنة بدراسات أخرى	محولات + شبكات البيان العصبونية Graph Neural Networks (GNN)	زمن التنقل ضمن الطرق وإحداثيات الطريق	Han et al [6]
قامت بتقييم خوارزميات التجميع دون استخدام الناتج في عملية التنبؤ.	Clustering only	عدد مركبات في الطرق	Toshniwal et al [7]
رغم تخفيف عبء التدريب إلا أن النموذج يعاني من نسبة خطأ عالية. بالإضافة لعدم تقييم أداء خوارزميات التجميع	Clustering + CNN + LSTM with Transfer Learning	عدد المركبات وسرعة الحركة	Zou et al [8]
يحقق النموذج دقة عالية في التنبؤ، ويخفف الكلفة الحسابية بما	Clustering + RNNs + Transformers	عدد المركبات في الطرق المتشابهة	Ours

يتناسب مع شبكات انترنت الأشياء.			
------------------------------------	--	--	--

7- مواد وطرق البحث:

يستعرض هذا القسم توصيف مجموعة البيانات وخوارزميات التجميع والتعلم العميق المستخدمة في الدراسة وآلية تقييم كل منها للحصول على أفضل نتيجة تنبؤ.

7-1 مجموعة البيانات المستخدمة في الدراسة:

تم استخدام مجموعة البيانات من مشروع [9] citypulse EU FP7 لمدينة Aarhus في الدنمارك، وذلك باستخدام مجموعة من الحساسات يبلغ عددها 136 حساس مثبتة في بدايات ونهايات الطرق في المدينة بهدف تجميع بيانات تخص عدد المركبات التي تعبر الطريق، متوسط سرعتها بالإضافة لنسب بعض الغازات في الجو كـ (CO₂, NO₂, ...) خلال فواصل زمنية تبلغ 5 دقائق، يتم تخزين البيانات ضمن ملفات بلاحقة CSV. لكل من حركة المرور مع سرعة الحركة وملفات أخرى لنسب الغازات في الجو لكل طريق عن طريق استخدام بنية سحابية تربط الحساسات مع بنية التخزين وذلك لمدة شهرين متتابعين. بلغ عدد التقارير المتولدة 449 تقرير لكافة الطرق ضمن المدينة وذلك عن طريقة بنية لتجميع البيانات. يوضح الجدول [2] عينة من بيانات حركة المرور المسجلة باستخدام الحساسات.

الجدول (2) عينة من مجموعة بيانات حركة المرور المستخدمة [9]

REPORT_ID	avgSpeed	vehicleCount	TIMESTAMP
158324	50	5	2014-08-01T07:50:00
158324	50	6	2014-08-01T07:55:00
158324	60	4	2014-08-01T08:00:00

يشير حقل report id لمعرفة يخصص الطريق المدروس، كما ويوضح الجدول [3] عينة من بيانات التلوث المستخدمة والتي يتم جمع قراءتها أيضاً باستخدام الحساسات:

الجدول (3) عينة من مجموعة البيانات التلوث المستخدمة [9]

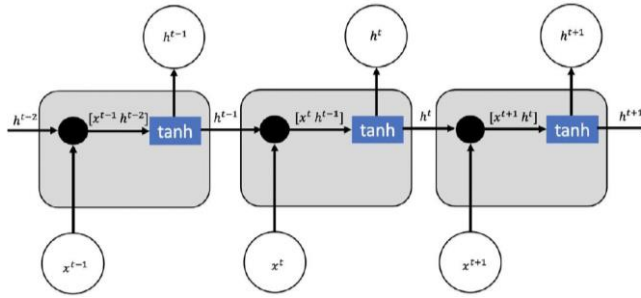
REPORT_ID	latitude	longitude	NO ₂	SO ₂	CO ₂	PM	O ₃	TIMESTAMP
158324	56.2317	10.10498	87	44	49	94	101	2014-08-01T07:50:00
158324	56.2317	10.10498	86	47	48	97	106	2014-08-01T07:55:00
158324	56.2317	10.10498	85	42	49	95	107	2014-08-01T08:00:00

كما وتحوي مجموعة البيانات السابقة معلومات أخرى تخص الطقس، الأحداث في المدينة والمناسبات الاجتماعية والفعاليات ضمنها وحالة مواقف السيارات.

7-2 خوارزميات التعلم العميق المستخدمة في الدراسة:

7-2-1 الشبكات العصبونية المتكررة:

تستخدم الشبكات العصبونية المتكررة (Recurrent Neural Networks) RNN لتحليل البيانات المتسلسلة كمعالجة اللغات الطبيعية والسلاسل الزمنية [10]، وتعتمد على مبدأ مشاركة القيم السابقة في عملية التنبؤ ضمن طبقات الشبكة مما يساهم في عملية تحسين التنبؤ عن الطرق التقليدية الأخرى [1]. يوضح الشكل [1] بنية شبكة RNN التقليدية:



شكل (1) بنية شبكة RNN التقليدية [10]

7-2-2 شبكات الذاكرة الطويلة قصيرة المدى LSTM:

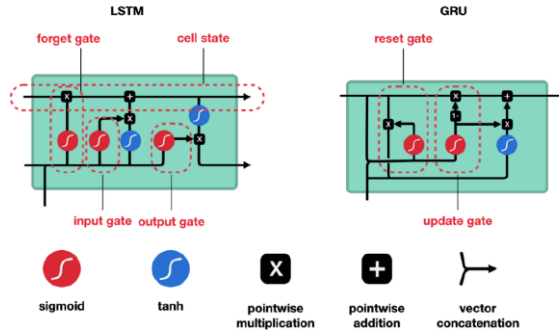
تعاني شبكات RNN التقليدية من مجموعة من المشاكل أهمها مشكلة العلاقة بين القيم السابقة للبيانات والقيم الحالية على المدى الطويل [17]، ولذلك ظهرت هذه الشبكات والتي قامت بتطوير الوحدات السابقة بإضافة خلية ذاكرة إليها مع وحدات لقراءة، نسيان وإضافة قيمة إلى هذه الخلية [1]. تحتوي كل وحدة LSTM على البوابات التالية:

- 1- بوابة الدخل (input gate): تحدد كمية المعلومات التي نريد قراءتها من الدخل الحالي والحالة السابقة [1].
- 2- بوابة النسيان (forget gate): تساعد هذه البوابة تحديد كمية المعلومات التي نحتاج لنسيانها من الذاكرة [1]، حيث نقوم بتطبيق تابع تفعيل على المدخلات وعندما يكون خرج هذا التابع مساوي لـ (1) فإنه يتم تمرير قيمة الخلية السابقة بالكامل وإلا لن تتم عملية النقل [17].
- 3- بوابة الخرج (output gate): تحدد هذه البوابة كمية المعلومات التي يجب استخدامها من القيمة الحالية لتوليد خرج البوابة [1].

3-2-7 وحدات إرجاع البوابة (GRU Gated Recurrent Units):

تعتبر هذه الوحدات مبسطة عن LSTM حيث تستخدم بوابتين فقط بدلاً من ثلاثة وتقوم بتمرير المعلومات المناسبة ضمن قيمة الحالة بدلاً من خلية الذاكرة [1]، وتتكون من:

- 1- بوابة إعادة التعيين (reset gate): تحدد مقدار المعلومات المخفية الواجب اعتبارها ضمن قيمة الحالة عند الوسمة الزمنية الحالية [1].
- 2- بوابة التحديث (update gate): تحدد بوابة التحديث المعلومات التي يجب تجاهلها والمعلومات الجديدة التي يجب تخزينها بتعبير آخر تساعد في تحديد المعلومات التي يجب نقلها للمستقبل [17]. أخيراً تعتبر وحدات GRU أقل كلفة حسابية من وحدات LSTM التي تتفوق من ناحية الدقة على حساب الكلفة. يوضح الشكل [2] بنية كل من وحدتي LSTM و GRU:



شكل (2) وحدات LSTM, GRU [1]

4-2-7 المحولات (Transformers) [1, 15]:

تعتبر المحولات (Transformers) من أهم نماذج التعلم العميق التي صُممت في البداية لمعالجة اللغة الطبيعية، لكنها أثبتت فعاليتها مؤخراً في التعامل مع السلاسل الزمنية. تتميز المحولات

بقدرتها على فهم العلاقات المعقدة بين عناصر السلسلة، حتى لو كانت بعيدة عن بعضها في الترتيب الزمني.

تعاني نماذج RNN بأنواعها المختلفة من عدم قدرتها على تحديد العلاقة بين القيم السابقة البعيدة في السلسلة مع القيم المستقبلية مما يضعف أداء هذه الخوارزميات وهنا تأتي ميزة المحولات التي تتجاوز هذه المشكلة من خلال:

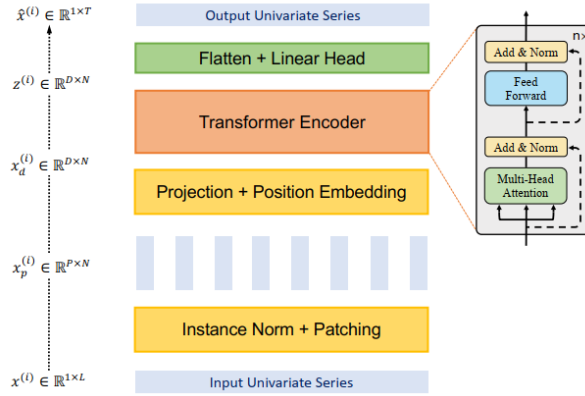
- عدم اعتمادها على الترتيب التتابعي، بل تنظر إلى السلسلة كاملة دفعة واحدة. بتعبير آخر تقوم المحولات بإضافة معلومات زمنية للمدخلات من خلال ما يعرف بالترميز الموضعي (Positional Encoding) مما يساهم في تعريف النقاط مع اللحظة الزمنية الخاصة بها على عكس خوارزميات RNN التي تعتمد على الترتيب المتتابع للبيانات.
- قدرتها على التركيز على الأجزاء الأهم من السلسلة عبر آلية الانتباه (Attention).
- تعمل بكفاءة عالية على الأجهزة الحديثة لأنها تُمكن من التوازي في المعالجة.

يتكون نموذج المحول من مكونين رئيسيين وهما:

1. المشفر (Encoder): يأخذ بيانات السلسلة الزمنية ويحوّلها إلى تمثيل داخلي غني من خلال عدة طبقات.
2. المفكك (Decoder): يأخذ التمثيل الناتج من المشفر ويستخدمه لتوليد التنبؤات المستقبلية خطوة بخطوة.

كما وتستخدم المحولات ما يعرف بالانتباه الذاتي (Self-Attention) والذي يعتبر من أقوى المميزات الموجودة في المحولات والتي تتمثل في قدرتها على التركيز على النقاط المهمة في السلسلة، بغض النظر عن موقعها. على سبيل المثال، عند التنبؤ بدرجة الحرارة غدًا، قد يولي

النموذج اهتماماً خاصاً لنمط حدث قبل أسبوعين إذا كان مشابهاً للحالة الحالية. يوضح الشكل [3] المخطط العام لنماذج المحولات.



شكل (3) البنية العامة لنموذج المحولات من أجل السلاسل الزمنية [15].

7-3 خوارزميات التجميع وعوامل تقييمها المستخدمة في الدراسة:

تُستخدم خوارزميات التجميع بهدف تشكيل مجموعات من نقاط البيانات من خلال وضعها ضمن عناقيد (Clusters) بحيث يكون التشابه بين النقاط الموجودة في نفس المجموعة عالياً ويكون التشابه أقل مع نقاط البيانات الموجودة في المجموعات الأخرى [10]. وتقسم هذه الخوارزميات لـ:

7-3-1 خوارزميات التقسيم (Partitioning Algorithms):

من أهم الخوارزميات المستخدمة في هذا النوع هي K-Means, K-Medoids ويكمن الفرق بينهما أن خوارزمية K-Means تختار المراكز الابتدائية بشكل عشوائي مقارنة بالخوارزمية الأخرى التي تختار المراكز من نقاط البيانات الموجودة [10]. كما ويمكننا إعادة تشغيل الخوارزمية بمراكز ابتدائية مختلفة من خلال استخدام معامل (n_init). ثم تقوم الخوارزميتين وبشكل تكراري اسناد النقاط للمراكز الأقرب لها، تليها عملية تحديث المراكز حتى تستقر المراكز. ومن

الخوارزميات الأخرى المستخدمة خوارزمية الانتشار التقريبي (Affinity Propagation) [4]، والفرق بين هذه الخوارزمية والخوارزميتين السابقتين أنها لا تحتاج عدد العناقيد كدخل لها، وهنا تكون جميع نقاط البيانات مرشحة لتكون مركز عنقود وبنفس الوقت تبحث عن مركز جديد لها. في كل مرحلة من مراحل عمل الخوارزمية تقوم كل نقطة أو غرض من مجموعة البيانات بحساب المسؤولية (Responsibility) والإتاحة (Availability) والتي تخص كل زوج من النقاط الموجودة ضمن مجموعة البيانات. تعبر المسؤولية بين نقطتين $r(i, k)$ عن وزن يدل على مقدار مناسبة (k) لتكون مركز للعقدة (i) أما الإتاحة بين النقطتين $a(i, k)$ تعبر عن وزن يمثل مقدار مناسبة (i) لتتنتمي للمركز الذي تتواجد فيه العقدة (k). وتكون قيمة العنقود لكل نقطة تمثل القيمة العظمى لمجموع القيمتين السابقتين.

7-3-2 طرائق التجميع الهرمي (Hierarchical Clustering) [10]:

تعمل هذه الطرق على تجميع البيانات بشكل هرمي (بتعبير آخر على شكل شجرة من العناقيد). ويتكون هذا النوع من طريقتين رئيسيتين (Top - Down) أو (Bottom - Up)، ومن أمثلة النوع الأخير خوارزمية Agglomerative Clustering والتي تبدأ باعتبار كل نقطة تشكل عنقوداً مستقلاً عن العقد الأخرى، وبشكل متتالي يتم جمع نقاط البيانات مع بعضها البعض الأقرب فالأقرب بناءً على مقياس تشابه معين حتى يتحقق شرط معين أو تصبح جميع النقاط ضمن عنقود وحيد. وتعتمد هذه الخوارزمية في عملها على نواة تحدد طبيعة التجميع بناءً على القيم العظمى أو الصغرى للتشابه أو القيمة المتوسطة للتشابه وتحدد باستخدام البارامتر (linkage).

7-3-3 التجميع المعتمد على الكثافة (Density Based Clustering) [10]:

تولد الخوارزميات السابقة عناقيد كروية بشكل عام، ولإيجاد عناقيد بإشكال أخرى نلجأ لاستخدام أنواع أخرى من الخوارزميات وأشهرها DBSCAN (Density-Based Spatial Clustering)

(of Applications with Noise) والتي تستخدم مجموعة البيانات ونصف القطر المعبر عن مجال الجوار لكل غرض وعتبة الكثافة. وتعمل الخوارزمية كما يلي:

1- تبدأ الخوارزمية بتحديد النقاط الأساسية وهي النقاط التي لديها على الأقل MinPts من النقاط في نطاق ϵ .

2- إذا كانت النقطة أساسية، يتم توسيع المجموعة عن طريق ضم جميع الجيران الذين هم أيضاً نقاط أساسية أو نقاط حدودية أما النقاط التي لا تنتمي إلى أي مجموعة تُعتبر ضجيج.

8- التجارب والنتائج ومناقشتها:

8-1 تجهيز البيانات:

قبل تطبيق الخوارزميات السابقة لابد لنا من القيام بمجموعة من المراحل للتأكد من أن البيانات خالية من المشاكل وذلك باستخدام مكتبة تحليل البيانات بلغة بايثون (Pandas) [11]، تعد مشكلة البيانات المفقودة من أهم مشاكل بيانات انترنت الأشياء والتي تحدث نتيجة عدم تمكن الحساسات من إرسال البيانات نتيجة مشكلة في الاتصال أو تم إهمال القراءة بسبب ضجيج البيانات. توجد العديد من الحلول لاستبدال القيم المفقودة في مجموعة البيانات وسنقوم باستخدام طريقة الإكمال الخطي (Linear Interpolation) والتي تعتمد على وجود علاقة خطية بين القيمة المفقودة والقيم السابقة واللاحقة لها وأظهرت هذه الطريقة أداءً أفضل من طرق أخرى كما في [18] بحيث تكون قيمة الخطأ ما بين القيم الفعلية والقيم المستبدلة أقل ما يمكن. تساهم عملية إدخال القراءات السابقة للحساسات في تحسين عملية التنبؤ (إدخال عامل الزمن) حيث حسنت هذه الآلية عملية التنبؤ كما في [14,18]. يوضح الجدول [4] شكل مجموعة البيانات من أجل قيمة النافذة تساوي 1 على سبيل المثال، بتعبير آخر بفرض أننا حصلنا على القراءات في لحظة ما (t) ونريد

التنبؤ بحركة المرور في هذه اللحظة فإننا نستخدم قراءات الحساس المسجلة في هذه اللحظة الحالية واللحظة السابقة (t-1).

الجدول (4) التنبؤ بحركة المرور بالاعتماد على حركة الطرق المجاورة

Avg speed	Road 1 (t)	Road 2 (t)	Road 3 (t)	Road 1 (t-1)	Road 2 (t-1)	Road 3 (t-1)	...	vehicleCount (output)
47	10	0	3	4	4	3	...	5
50	19	8	5	10	0	3	...	6
60	3	6	2	3	6	2	...	4

بعد عملية التهيئة السابقة نقوم بتقسيم مجموعة البيانات لمجموعتين (80% من البيانات للتدريب و20% للاختبار على الترتيب وذلك بحسب طبيعة بيانات السلاسل الزمنية). أما من أجل تجميع الطرق المتشابهة ضمن عناقيد (clusters). سنقوم بدراسة تأثير تجميع البيانات بشكل ربع يوم، نصف يوم، يومي لحركة المرور. يوضح الشكل [4] الشكل الجديد للبيانات والذي يحوي متوسط الحركة المكافئ لكل طريق بشكل يومي من أجل الطرق المدروسة على حدا بحيث يكون معرف الطريق هو دليل كل سطر على سبيل المثال وذلك لاستخدامها في عملية التجميع.

day	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	...
road_id											
158324	5.085062	3.769097	3.335938	5.237847	5.136522	3.366551	3.221354	4.853299	3.748264	3.427951	...
158355	5.261411	4.083333	3.619792	5.904514	5.555652	3.571924	3.428819	5.477431	3.980035	4.010417	...
158386	3.447484	2.450751	2.488715	3.436632	3.615523	2.086207	2.117391	3.280105	2.488294	2.208333	...

الشكل (4) متوسط حركة الطرق اليومية المستخدمة في التجميع

8-2 التنبؤ بحركة المرور بالاعتماد على حركة المرور في جميع طرق المدينة:

8-2-1 ضبط بارامترات النماذج المدروسة:

توجد العديد من البارامترات التي يمكن ضبطها أثناء الدراسة وتم اختيار أهم البارامترات التي تؤثر على أداء الخوارزميات كحجم الإطار (النافذة) التي يجب علينا استخدامها لتوليد البيانات اللازمة للتنبؤ وبعض البارامترات المشتركة كعدد الطبقات، عدد العصبونات وغيرها. تم تجربة فضاء من القيم الممكنة لكل بارامتر بحيث تحقق الخوارزميات أفضل أداء لها. يوضح الجدول [5] البارامترات التي تحقق من خلالها كل خوارزمية الأداء الأفضل لها. وللحصول على هذه القيم قمنا بالبحث ضمن مجموعة من القيم الممكنة لكل من هذه البارامترات والتي تحقق أفضل أداء ممكن لكل خوارزمية.

الجدول (5) البارامترات التي تحقق أفضل أداء لكل من الخوارزميات المدروسة

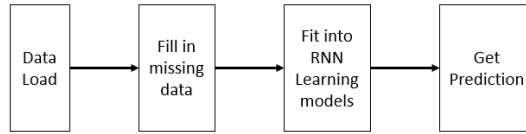
حجم النافذة	حجم التجميع (batch size)	معدل التعلم	عدد العصبونات	معدل dropout	عدد الطبقات المخفية	الخوارزمية
11	32	0.0002	200	0.2	2	LSTM
31	64	0.0002	200	0.1	2	GRU
28	16	0.0001	200	0.2	1	RNN

تساهم طبقة dropout في منع النموذج من مواسمة البيانات بشكل كبير (Overfitting) وذلك من خلال إهمال نسبة من خرج الطبقات الأخرى أثناء التدريب مما يساهم في تحسين أداء الشبكات ونقوم بتحديد نسبة العقد التي نريد إهمالها من خلال قيمة احتمال. أخيراً، يوضح الجدول [6] قيم البارامترات الخاصة بخوارزمية المحولات.

الجدول (6) البارامترات المستخدمة من أجل المحولات

المعامل	فضاء التمثيل	عدد رؤوس الانتباه	عدد طبقات المحول	dropout	معدل التعلم	حجم التجميع	حجم النافذة
القيمة	64	4	1	0.1	0.001	64	22

أخيراً يوضح الشكل [5] النموذج العام المستخدم في التجربة الأولى:



شكل (5) المخطط الصندوقي للتجربة الأولى

8-2-2 مقاييس التقييم:

لتقييم أداء هذه الخوارزميات سنستخدم مجموعة من المعايير التي تقوم بتقييم أداء خوارزميات التعلم العميق كما في [14] وهي (Root Mean Squared Error) (RMSE) وهو عبارة عن الجذر التربيعي لمتوسط الفروق بين القيم الحقيقية (O_i) والقيم الناتجة عن التنبؤ (P_i) ويعطى بالمعادلة التالية:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (O_i - P_i)^2} \quad (1) \quad [14]$$

يمثل المعيار السابق الجذر التربيعي لمربع الفروق بين القيم الفعلية والقيمة الناتجة عن التنبؤ وبالتالي يوضح مدى قرب القيم الناتجة عن التنبؤ من القيم الفعلية. كما ويمكن استخدام معامل Mean Absolute Error (MAE) والذي يعبر عن المتوسط الحسابي للقيمة المطلقة للفروق بين القيم الحقيقية والقيم الناتجة عن التنبؤ.

$$MAE = \frac{1}{n} \sum_{i=1}^n |O_i - P_i| \quad (2) \quad [14]$$

أخيراً يمكن استخدام معامل R2 ويعطى بالمعادلة التالية:

$$R^2 = 1 - \frac{\sum_{i=1}^n (O_i - P_i)^2}{\sum_{i=1}^n (O_i - \bar{O})^2}, \text{ where } \bar{O} = \frac{1}{n} \sum_{i=1}^n O_i \quad (3) \quad [14]$$

يسمى المعيار السابق بمركبة التحديد ويوضح المعامل السابق مدى مناسبة البيانات لنموذج الانحدار أو مدى جودة تفسير البيانات الموجودة لدينا باستخدام متحول الدخل وكلما كانت القيمة أكبر يدل على أداء أفضل للخوارزمية.

3-2-8 النتائج:

تحقق المحولات أداءً أفضل من باقي الخوارزميات المدروسة الأخرى بفضل آلية الانتباه والتي تقوم بدراسة السلسلة ككل ويهتم بتموضع النقاط زمنياً مقارنةً بالتتابع المتولد في خوارزميات RNN الأخرى، فيما تحقق خوارزمية RNN أداءً أسوأ من باقي الخوارزميات بسبب معانيتها من التبعيات الطويلة في السلسلة الأمر الذي يضعف أداءها. فيما تحتاج خوارزمية GRU لعدد نوافذ أكثر لتحقيق دقة أفضل وقريبة من خوارزمية LSTM التي تتفوق على الخوارزمية السابقة. يوضح الجدول [7] تقييم أداء الخوارزميات المدروسة من أجل عوامل التقييم المختلفة.

الجدول (7) تقييم أداء خوارزميات التعلم العميق بالاعتماد على بيانات جميع الطرق

R2	MAE	RMSE	الخوارزمية
0.7504	1.6711	2.5734	LSTM
0.7233	1.8027	2.664	GRU
0.6123	2.1558	3.0451	RNN

0.8292	1.3917	2.1395	Transformer
--------	--------	--------	-------------

8-3 التنبؤ بحركة المرور بالاعتماد على حركة المرور في الطرق المتشابهة:

تتكون هذه التجربة من مرحلتين الأولى تشمل تحديد الطرق المتشابهة بحركة المرور وذلك باستخدام خوارزميات التجميع ودراسة تأثير المعاملات وآلية التجميع الزمنية للبيانات. أما في المرحلة الثانية فنقوم باستخدام بيانات الطرق المتشابهة في عملية التنبؤ بدلاً من الاعتماد على بيانات كافة الطرق في المدينة.

8-3-1 ضبط بارامترات خوارزميات التجميع:

لأجل مقارنة أداء الخوارزميات مع بعضها البعض سنقوم باستخدام عدد العناقيد الناتجة عن خوارزمية الانتشار التقريبي ونلاحظ أداء الخوارزميات الأخرى عند تجميع البيانات ضمن نفس العدد من العناقيد. فيما تم استخدام أفضل البارامترات التي تحقق أفضل قيم من أجل مقياس التقييم المستخدم. حيث يوضح الجدول [8] البارامترات المستخدمة من أجل خوارزميات التجميع المدروسة.

الجدول (8) البارامترات المستخدمة لكل من خوارزميات التجميع المدروسة:

المعاملات	الخوارزمية
init= kmeans++, n_init = auto	K-Means
linkage = average	التجميع الهرمي
-	الانتشار التقريبي
eps = 1, min_samples = 10	DBSCAN

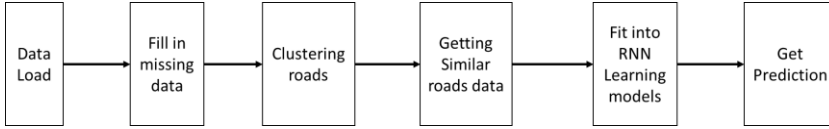
8-3-2 مقاييس التقييم:

يتم تقييم أداء خوارزميات التجميع من خلال عدة عوامل أشهرها معامل Silhouette Coefficient [4] وكلما كانت قيمة المعامل أكبر كلما كانت جودة العناقيد أفضل والذي يحسب كما يلي:

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}} \quad (4) \quad [4]$$

- $a(i)$: متوسط المسافة بين النقطة والنقاط في مجموعتها.
- $b(i)$: أقل متوسط مسافة بين النقطة وأقرب مجموعة أخرى.

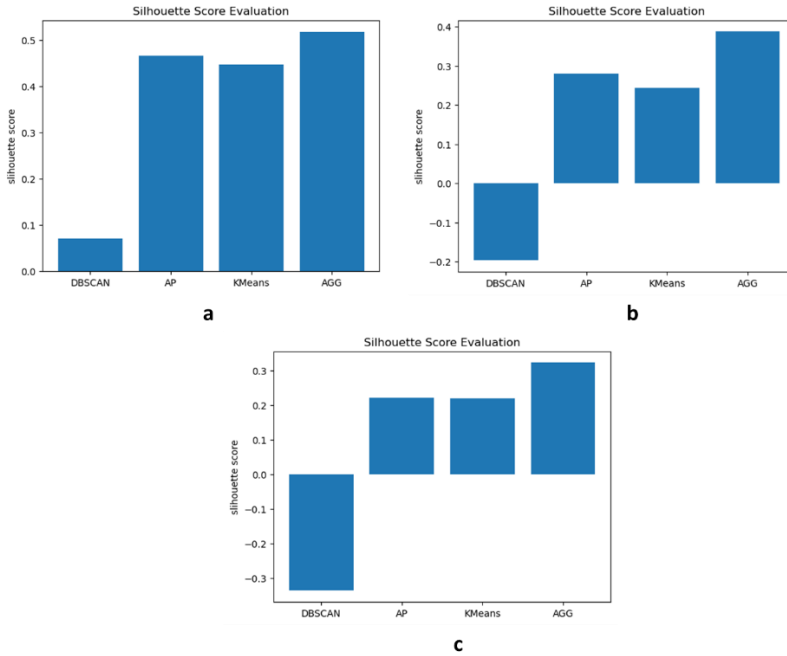
يوضح الشكل [6] المخطط الصندوقي لبناء نموذج للتنبؤ معتمد على خوارزميات التجميع والتعلم العميق.



الشكل (6) المخطط الصندوقي للتجربة الثانية

8-3-3 تقييم أداء خوارزميات التجميع لتحديد نماذج الحركة ضمن الطرق:

يوضح الشكل [7] نتائج أداء كل من الخوارزميات المدروسة عند استخدام عدد العناقيد الناتج عن خوارزمية الانتشار التقريبي:



الشكل (7) تقييم أداء خوارزميات التجميع عند استخدام عدد عناقيد موحد

a. تجميع يومي b. تجميع نصف يوم c. تجميع ربع يوم

نلاحظ أن أداء الخوارزميات يتأثر بحسب طبيعة التجميع حيث تحقق الخوارزميات أفضل أداء لها عند تجميع البيانات بشكل يومي وينخفض مع تجميع البيانات بالاعتماد على متوسط الحركة بشكل نصف يومي أو ربع يومي وذلك بشكل مشابه للدراسة [7] تحقق آلية التجميع بالاعتماد على عدد العناقيد الناتجة عن خوارزمية الانتشار التقريبي أداءً أفضل من استخدام عدد العناقيد المستخدمة في [7] فيما تحقق DBSCAN أداءً أقل بسبب عزلها للضجيج مما يقلل قيمة المعامل مما يزيد من تباعد النقاط عن المجموعات الأخرى. كما وتحقق طريقة تجميع الطرق بناءً على حركة الطرق بشكل يومي ونصف يومي تحسن أداء بعض الخوارزميات عن التجميع الديناميكي

المستخدمة في [4]. سنقوم باستخدام نفس المعاملات المستخدمة في الجدولين [4,5] سابقاً من أجل خوارزميات التعلم العميق.

8-3-4 تحسين عملية التنبؤ بحركة المرور باستخدام خوارزميات التجميع:

سنحدث عن تأثير استخدام بيانات الطرق المتشابهة في عملية التنبؤ بدلاً من استخدام كافة الطرق الموجودة في المدينة، كما توضح الجداول [9, 10, 11, 12].

الجدول (9) ناتج تقييم خوارزمية LSTM مع استخدام خوارزميات التجميع

R2	MAE	RMSE	الخوارزمية (LSTM)
0.7542	1.6773	2.5004	Agglomerative Clustering
0.7513	1.6732	2.5154	Affinity Propagation
0.7292	1.7727	2.6244	Kmeans
0.54336	1.19575	1.8323	DBSCAN

الجدول (10) ناتج تقييم خوارزمية GRU مع استخدام خوارزميات التجميع

R2	MAE	RMSE	الخوارزمية (GRU)
0.7587	1.6763	2.4780	Agglomerative Clustering
0.7554	1.6817	2.4952	Affinity Propagation
0.7127	1.8238	2.7039	Kmeans
0.5503	1.2218	1.8188	DBSCAN

الجدول (11) ناتج تقييم خوارزمية RNN مع استخدام خوارزميات التجميع

R2	MAE	RMSE	الخوارزمية (RNN)
0.7145	1.8495	2.6953	Agglomerative Clustering
0.7087	1.8445	2.7228	Affinity Propagation
0.6493	1.9824	2.9875	Kmeans

0.5407	1.2721	1.8378	DBSCAN
--------	--------	--------	--------

الجدول (12) ناتج تقييم خوارزمية المحولات مع استخدام خوارزميات التجميع

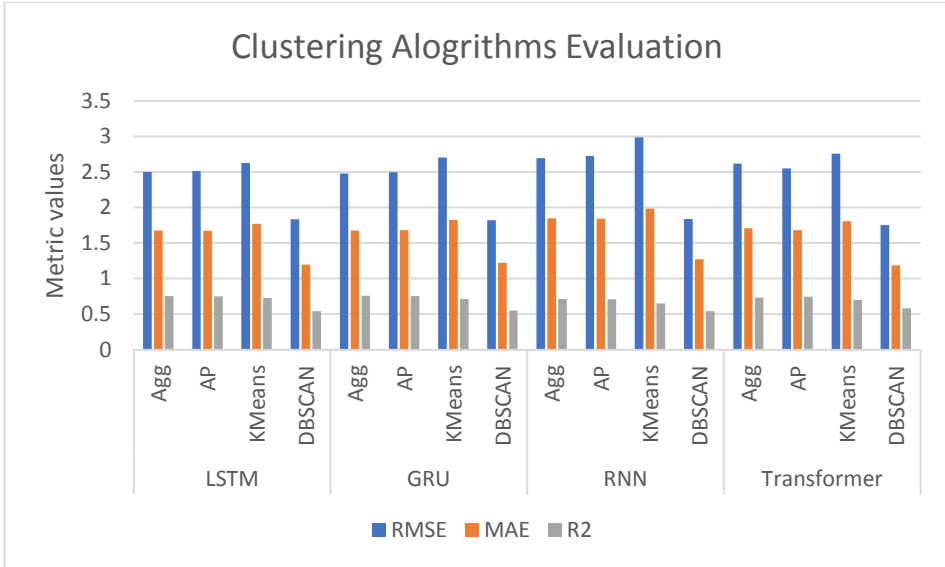
R2	MAE	RMSE	الخوارزمية (Transformer)
0.7307	1.7089	2.6175	Agglomerative Clustering
0.7442	1.6791	2.5514	Affinity Propagation
0.7016	1.8061	2.7555	Kmeans
0.5821	1.1870	1.7530	DBSCAN

يوضح الجدول [13] عدد الطرق المستخدمة في التنبؤ بالاعتماد والناتج عن كل خوارزمية:

الجدول (13) عدد الطرق المستخدمة في عملية التنبؤ بعد تطبيق خوارزميات التجميع

عدد الطرق المستخدمة في التنبؤ	الخوارزمية
448	No clustering
68	Agglomerative Clustering
68	Affinity Propagation
35	Kmeans
265	DBSCAN

يظهر الشكل [8] تقييم أداء تأثير خوارزميات التجميع على أداء خوارزميات التعلم العميق:



الشكل (8) تأثير استخدام خوارزميات التجميع على أداء خوارزميات التعلم العميق

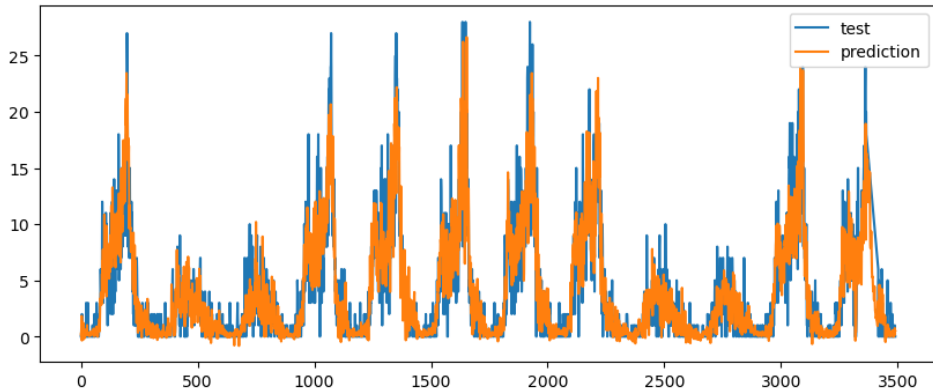
نلاحظ أن استخدام خوارزمية DBSCAN تساهم في تقليل الخطأ الناتج عن التنبؤ لكل من معاملات (RMSE, MAE) بسبب طريقة التجميع المعتمدة على الكثافة على عكس الخوارزميات الأخرى وكذلك قدرتها على التخلص من الضجيج. فيما تحقق الخوارزمية السابقة قيمة R2 أقل من الخوارزميات الأخرى بسبب تقارب البيانات من بعضها البعض مما قلل قيمة التباين في معامل R2 على عكس الخوارزميات الأخرى التي تملك تبايناً أعلى من أجل هذا المعامل بسبب بعض الضجيج وطريقة التجميع المعتمدة على المسافة. كما وتساهم خوارزميات التجميع في تقليل كلفة وتعقيد بناء نماذج التعلم العميق للتنبؤ بحركة المرور مما يتناسب مع تطبيقات انترنت الأشياء. يوضح الجدول [14] كيفية مساهمة خوارزميات التجميع من خلال تقليل عدد البارامترات التي يتم تدريبها من جهة وتخفيف العمليات الحسابية الأساسية على الأرقام الحقيقية (Floating Point Operations) FLOPs وذلك من أجل خوارزمية LSTM:

الجدول (14) تأثير استخدام خوارزميات التجميع على الكلفة الحسابية لبناء نماذج التعلم

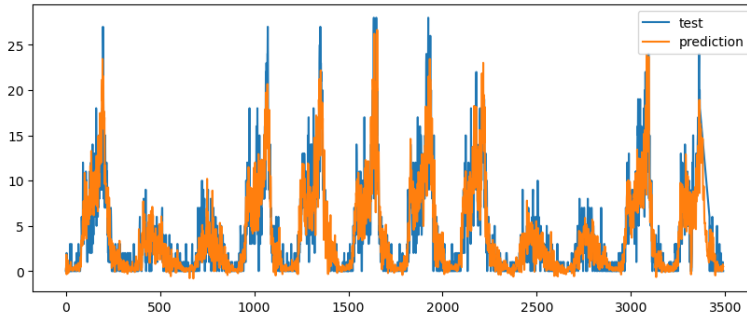
العميق

الخوارزمية	عدد البارامترات التي يتم تدريبها	عدد العمليات الحسابية (FLOPs)	نسبة تخفيف الكلفة الحسابية
No clustering	842,601	296,531,232	-
AGG-Clustering	539,401	189,804,832	35.98%
Affinity Propagation	539,401	189,804,832	35.98%
Kmeans	513,001	180,512,032	39.12%
DBSCAN	697,001	245,280,032	17.28%

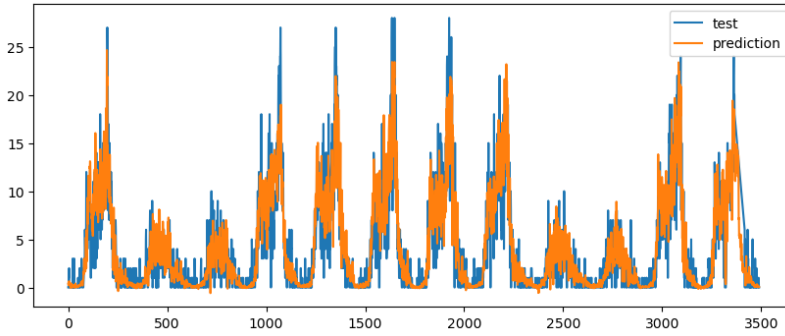
توضح الأشكال [9, 10, 11, 12] شكل الإشارة الناتجة عن استخدام المحولات مع خوارزميات التجميع المختلفة في عملية التنبؤ:



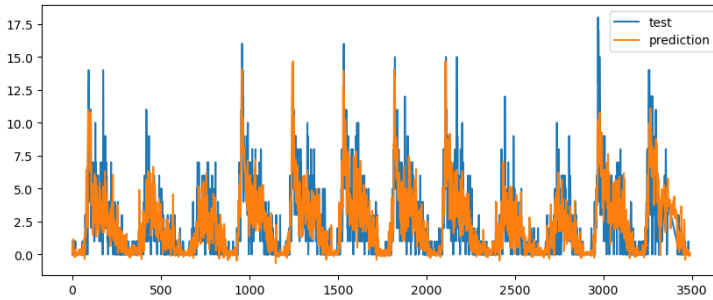
الشكل (9) إشارة التنبؤ الناتجة عن خوارزمية المحولات مع AGG clustering



الشكل (10) إشارة التنبؤ الناتجة عن خوارزمية المحولات مع AP clustering



الشكل (11) إشارة التنبؤ الناتجة عن خوارزمية المحولات مع KMeans clustering



الشكل (11) إشارة التنبؤ الناتجة عن خوارزمية المحولات مع DBSCAN clustering

9- مناقشة النتائج:

تساهم خوارزميات التجميع والتعلم العميق في تحسين جودة التنبؤ بحركة المرور حيث حققت الخوارزميات المعتمدة على المحولات أداءً أفضل من خوارزميات RNN التقليدية، فيما تساعد خوارزميات التجميع على تحسين أداء بعض الخوارزميات من جهة وتخفف تعقيد هذه الخوارزميات من جهة أخرى مع الحفاظ على دقة مقبولة وهذا ما يتناسب مع بيئة انترنت الأشياء التي تتطلب نماذج قليلة الكلفة حسابياً. تحقق النماذج المعتمدة على خوارزمية DBSCAN أداءً أفضل في التنبؤ من الخوارزميات الأخرى نظراً لآلية التجميع المعتمدة على الكثافة بما يشبه حركة المرور في الطرق التي تعتمد على كثافة المركبات في الطريق وإزالة الضجيج وعدم التقيد بشكل محدد للعناقيد الناتجة عن التجميع على عكس الخوارزميات الأخرى التي تفرض تشكيل المجموعات على شكل كروي مما يحقق دقة أفضل في عملية التنبؤ.

10- المقترحات والتوصيات المستقبلية:

توجد العديد من الخواص الأخرى التي يمكن دراستها في مجموعة البيانات وتأثيرها على حركة المرور كبيانات الطقس، المناسبات الاجتماعية في المدينة، مواقف السيارات وغيرها.. كما وتعاني مجموعة البيانات من كونها صغيرة نسبياً لذا يمكن دراسة تأثير النماذج على مجموعة بيانات أكبر وهذا يساهم في تحسين أداء خوارزميات التعلم العميق التي تعتمد على وجود كميات كبيرة من البيانات. أخيراً يمكن بناء لوحة تحكم (Dashboard) لعرض تفاصيل حركة المرور ضمن المدينة وتوضيح اختيار المسار الأقصر بناءً على التنبؤ بحركة المرور المستقبلية.

11- المراجع:

[1] Joseph, M., 2022. Modern Time Series Forecasting with Python: Explore industry-ready time series forecasting using modern machine learning and deep learning. Packt Publishing Ltd.

- [2] BELLO, R.D., SULAIMON, H.A., ABDULRAHMAN, M.L. and AHMAD, M.A., 2020. DEEP RECURRENT NEURAL NETWORK FOR POLLUTION FORECASTING IN SMART CITIES. BIMA JOURNAL OF SCIENCE AND TECHNOLOGY (2536-6041), 4(02), pp.199-205.
- [3] Alkarim, A.S., Al-Ghamdi, A.S.A.M. and Ragab, M., 2024. Ensemble Learning-based Algorithms for Traffic Flow Prediction in Smart Traffic Systems. Engineering, Technology & Applied Science Research, 14(2), pp.13090-13094.
- [4] Wang, Z., Thulasiraman, P. and Thulasiram, R., 2019, July. A dynamic traffic awareness system for urban driving. In 2019 international conference on Internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData) (pp. 945-952). IEEE.
- [5] Wang, Z. and Thulasiraman, P., 2019, November. Foreseeing congestion using LSTM on urban traffic flow clusters. In 2019 6th international conference on systems and informatics (ICSAI) (pp. 768-774). IEEE.
- [6] Han, S.Y., Sun, Q.W., Zhao, Q., Han, R.Z. and Chen, Y.H., 2022. Traffic forecasting based on integration of adaptive subgraph

reformulation and spatio-temporal deep learning model. Electronics, 11(6), p.861.

[7] Toshniwal, D., Chaturvedi, N., Parida, M., Garg, A., Choudhary, C. and Choudhary, Y., 2020. Application of clustering algorithms for spatio-temporal analysis of urban traffic data. Transportation Research Procedia, 48, pp.1046–1059.

[8] Zou, X. and Chung, E., 2024. Traffic prediction via clustering and deep transfer learning with limited data. Computer-Aided Civil and Infrastructure Engineering.

[9] CityPulse Dataset Collection · A collection of semantically annotated datasets for the CityPulse EU FP7 Project <http://iot.ee.surrey.ac.uk:8080/index.html> last access 12/7/2024

[10] Jiawei, H., Jian, P. and Hanghang, T., 2023. Data mining: concepts and techniques. Morgan kaufmann.

[11] VanderPlas, J., 2022. Python data science handbook (No. PUBDB–2023–07774). O'Reilly.

[12] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, pp.2825–2830.

[13] Peixeiro, M., 2022. Time series forecasting in python. Simon and Schuster.

[14] Shouaib, M., Metwally, K. and Badran, K., 2023. An Enhanced Time-dependent Traffic Flow Prediction in Smart Cities. *Advances in Electrical & Computer Engineering*, 23(3).

[15] Nie, Y., Nguyen, N.H., Sinthong, P. and Kalagnanam, J., 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

[16] م. آلاء السباعي، د. محسن حسين، د. وسيم رمضان، 2022. أنظمة تحليل بيانات الزمن الحقيقي الضخمة الناتجة عن انترنت الأشياء. مجلة جامعة حمص-سلسلة العلوم الهندسية الميكانيكية والكهربائية والمعلوماتية، 44(14).

[17] ميلاد وزان، ترجمة الدكتور علاء طعيمة، التعلم العميق المبادئ والمفاهيم والأساليب 2024، كلية علوم الحاسوب وتكنولوجيا المعلومات، جامعة القادسية.

[18] م. خالد قندقجي، د. ماهر عباس، د. أسماء شعار، 2025. تحليل حركة المرور ضمن المدن الذكية. مجلة جامعة حمص-سلسلة العلوم الهندسية الميكانيكية والكهربائية والمعلوماتية، 47(2).

[19] م. حلا إبراهيم، د. رانيا لطفي، د. أسماء شعار، 2023. تحليل السلاسل الزمنية لاكتشاف الشذوذ ضمن حركة المرور. مجلة جامعة حمص-سلسلة العلوم الهندسية الميكانيكية والكهربائية والمعلوماتية، 45(20).

مراجعة لموازنة حمل الخوادم في الشبكات المعرفة برمجياً

م.ميساء الضاهر¹ د.ماهر عباس² د.محسن عبود³

الملخص

تقدم هذه المراجعة تحليلاً شاملاً للعديد من الدراسات البحثية التي تركز على تقنيات موازنة أحمال الخوادم في سياق الشبكات المعرفة بالبرمجيات (SDN). عرضت بعض هذه الدراسات منهجيات وخوارزميات متنوعة لتحسين كفاءة النظام والحفاظ على استقراره في ظل ظروف حمل متغيرة، بينما قارنت دراسات أخرى هذه المنهجيات والأساليب لتحديد الحل الأكثر فعالية. تناقش هذه المقالة بنية الشبكة المستخدمة والمقاييس الرئيسية التي درست في كل مقالة، مثل زمن الاستجابة، والإنتاجية، وزمن الوصول، واستخدام الموارد، وهي مؤشرات أساسية للأداء في بيئات الشبكات المعرفة بالبرمجيات. كما تسلط المراجعة الضوء على الإعدادات التجريبية وأدوات المحاكاة الشائعة الاستخدام في الأدبيات، مما يوفر رؤى حول كيفية تقييم الباحثين لتقنياتهم المقترحة والتحقق من صحتها. بالإضافة إلى ذلك، يتناول البحث التوجهات المستقبلية التي اقترحتها الدراسات لتسليط الضوء على مجالات الابتكار المحتملة في هذا المجال، بما في ذلك دمج الذكاء الاصطناعي، والتعلم الآلي، وآليات اتخاذ القرار الذكية لتعزيز الأداء في ظروف الشبكة الديناميكية. ثم تقدم هذه الورقة ملخصاً واضحاً للدراسات البحثية في جدول منظم يسهل الفهم ويدعم جهود البحث المستقبلية في موازنة أحمال الخوادم ضمن بيئة الشبكات المعرفة بالبرمجيات. تهدف هذه المراجعة إلى أن تكون مرجعاً قيماً للأكاديميين والممارسين الذين يسعون إلى تحسين توزيع الأحمال وتحسين أداء الأنظمة القائمة على الشبكات المعرفة بالبرمجيات (SDN).

¹ م.ميساء الضاهر طالبة دكتوراه في كلية الهندسة المعلوماتية، قسم هندسة الشبكات والنظم الحاسوبية، جامعة حمص.
² د.ماهر عباس أستاذ مساعد في كلية الهندسة المعلوماتية، قسم هندسة الشبكات والنظم الحاسوبية، جامعة حمص.
³ د.محسن عبود مدرس في كلية الهندسة المعلوماتية، قسم هندسة الشبكات والنظم الحاسوبية، جامعة حمص.

الكلمات المفتاحية: الشبكات المعرفة برمجياً ، موازنة حمل الخادم، الخوارزميات، أداء الشبكة.

Review of Server Load Balancing in Software Defined Networks

Mayssaa Aldaher¹, Dr. Maher Abbas², Dr. Mohssen Abboud³

¹ PhD Student, Department of Network Engineering and Computer Systems, Homs University, Syria (E-mail: maldaher@homs-univ.edu.sy).

² Assistant Professor, Department of Network Engineering and Computer Systems, Homs University, Syria (E-mail: mabbaas@homs-univ.edu.sy)

³Lecturer, Department of Network Engineering and Computer Systems, Homs University, Syria (E-mail: mabboud@homs-univ.edu.sy).

Abstract—*This review provides a comprehensive analysis of many research studies focused on server load balancing techniques in the context of software-defined networks (SDNs). Some of these studies presented diverse methodologies and algorithms used to improve system efficiency and maintain stability under variable load conditions, while others compared these methodologies and approaches to identify the most effective solutions. Our article discusses the topology used and the key metrics studied in each article, such as response time, throughput, latency, and resource utilization, which are essential indicators of performance in SDN environments. The review also highlights the experimental setups and simulation tools commonly used in the literature, offering insights into how researchers evaluate and validate their proposed techniques. In addition, the research addresses future directions suggested by the studies to highlight potential areas of innovation in this field, including the integration of artificial intelligence, machine learning, and intelligent decision-making mechanisms to enhance performance in dynamic network conditions. This paper then provides a clear summary of the research studies in a structured table that facilitates understanding and supports future research efforts in server load balancing within an SDN environment. The review aims to serve as a valuable reference for academics and practitioners seeking to optimize load distribution and improve the performance of SDN-based systems.*

Keywords: *Software Defined Network; Server Load Balancing; Algorithms; Network Performance*

I. INTRODUCTION

Load balancing is now an irreplaceable element of contemporary network architectures, which guarantees that the load is efficiently shared among the servers and eliminates a bottleneck that leads to performance degradation. In software-defined networks (SDNs), load balancing is even more important because the control and data planes are decoupled, and users can centrally control traffic flows using their controllers. This programmability also opens the doors to adaptive and intelligent mechanisms that are stronger than the tools of traditional, static load balancers.

The load balancing of servers in SDN is a thorny issue that is yet to be resolved. Most of the current methods use simplified topologies, are based on test cases or are based on early algorithms to simulate the dynamics and heterogeneity of real traffic. Others are more concentrated on the link-level or routing-level balancing and the optimization of the servers is not fully explored. Moreover, there are no standardized assessment datasets and benchmarking procedures, making the comparison of findings across studies a challenging issue on which the applicability of suggested solutions can be restricted. These flaws highlight the urgency of critical reviews that find trends, limitations, and recommend directions of future innovation.

Recent literature points to a number of new trends such as the use of fuzzy logic, machine learning, and adaptive decision-making models to enhance response time, throughput and the overall quality of service. Nevertheless, these methods also tend to add some complexity, computational load or reliance on artificial data, which lowers their feasibility. Not the least, scalability and energy efficiency are seldom viewed as high-priority values, although they are the focus of SDN-enabled data centres and cloud services. These gaps need to be found and the performance of the current algorithms against them should be assessed in order to advance towards more robust and scalable solutions.

The work is valuable in that it synthesizes and critically evaluates various methods in SDN to load balance servers, with specific focus to the mechanisms employed, the performance measurements taken, and the experimentation employed. It aims to clarify researchers and practitioners

about the existing limitations and innovations in this field of rapid development by expressing the unresolved issues and outlining the emerging opportunities in the area.

II. SOFTWARE-DEFINED NETWORKING

SDN represents a networking paradigm that separates the control plane from the data plane, facilitating network evolution, interoperability, and scalability. This separation is the foundation of SDN and is made possible by decoupling switching components [1]. Software-defined networks were developed to achieve factory-independent control from a single logical point, facilitate innovation, and enable simple programmatic control of the network data path. Separating forwarding devices from control logic allows for easier deployment of new protocols and applications, clear network visualization and management, and the integration of various middle boxes into programmatic control. Instead of enforcing policies and implementing protocols on dispersed devices, the network is reduced to simple forwarding devices and a network controller (or controllers) that performs decision-making [2][37]. Notably, one of the most prominent examples of middlebox functions implemented using SDN is load balancers [3][4].

A. Software-Defined Networking Architecture

The SDN model consists of three layers: the infrastructure layer, the control layer, and the application layer, which stack on top of each other, as shown in Figure 1.

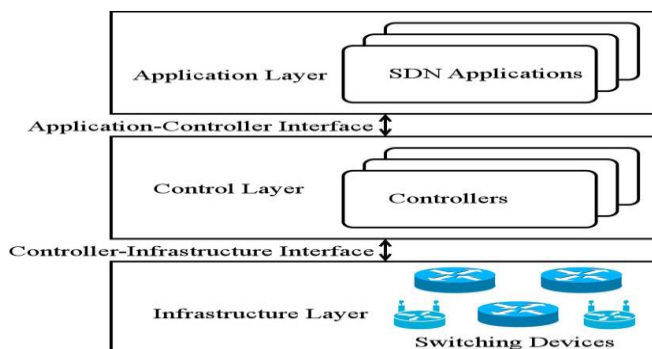


Fig. 1. SDN architecture [5]

- **The infrastructure layer:** Also called the data layer, it consists of switching devices (such as switches, routers, etc.). The functions of switching devices are: First, they are responsible for collecting network status, temporarily storing it on local devices, and sending it to controllers. Network status may include information such as network topology, traffic statistics, and network usage. Second, they are responsible for processing packets based on rules provided by the controller [5].
- **The control layer:** Represents the intelligence of the network. This approach enables the controller to retain a comprehensive awareness of the network' state within the infrastructure layer [1].
- **The Application Layer:** Contains SDN applications designed to meet user requirements. Through the programmability inherent in the control layer facilitates application-level access to, and governance of, the switching infrastructure [5].

The abstraction of the layers described above makes it easier for programmers to work on a single network layer, rather than thousands of different physical devices, through application programming interfaces (APIs) [1]. External management systems or network services may wish to extract information about the infrastructure or control an aspect of network behaviour or policy. This is done through the Northbound Application Interface (NAI) [2].

An important aspect of SDN is the connection between the data plane and the control plane, where forwarding elements are controlled by an open interface. It is important that this connection remains accessible and secure. The OpenFlow protocol can be viewed as one possible implementation of switch-controller interactions, as it defines the connection between switching devices and the network controller and is called the Southbound Interface protocol [2].

B. OpenFlow Protocol

OpenFlow is an open standard protocol specifically designed for SDN networks, allowing communication between the data and control planes,

as shown in Figure 2. OpenFlow switches and controllers can communicate using the OpenFlow protocol over a secure channel. The protocol defines various messages, such as receiving a packet, sending a packet, modifying the routing table, and obtaining statistics, which can be exchanged between the switch and controller [1].

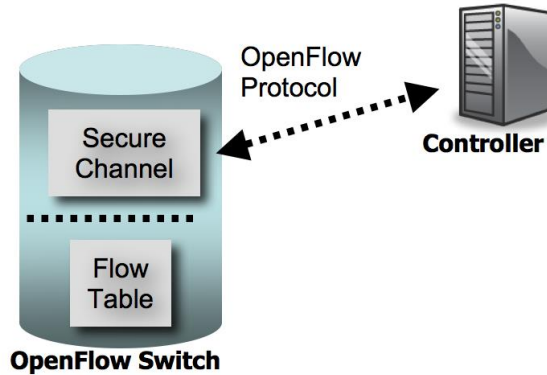


Fig. 2. Communication of an OpenFlow switch with a controller using the OpenFlow protocol [6]

The OpenFlow protocol was initially developed at Stanford University in 2008 to enable researchers to run experimental protocols on campus networks. OpenFlow is now being added as a feature to commercial networking devices, providing a standard, vendor-free interface for accessing Ethernet switches, routers, and wireless access points. Furthermore, these so-called "OpenFlow-enabled" devices allow access without requiring manufacturers to reveal the internal workings of their products. Therefore, the OpenFlow protocol makes it easy to deploy innovative routing and switching protocols [1].

C. Controllers

The controller acts as the main unit responsible for overseeing the key functions of the control plane. It creates updates and removes flow entries in the flow tables on the switch. In addition to its basic functionality, it can be extended to perform additional critical tasks such as routing, network access [6] and load balancing [3][4]. It can serve one or multiple switches,

depending on the network design. Currently, there are many controllers available, open source and based on various programming languages [6][38]. For example, POX and Ryu are based on Python, while Java is used in platforms such as Floodlight and OpenDaylight [7].

D. Load Balancing in Software Defined Networks

Load balancing is one of the fundamental pillars that contribute to improving the performance and efficiency of software-defined networks (SDNs). With the rapid expansion of Internet-based services, such as cloud computing, the Internet of Things, and time-sensitive applications, load balancing has become a critical component for ensuring network stability, optimizing resource utilization, and maintaining low response times while minimizing data loss. This fundamental importance of load balancing has prompted researchers to explore it in depth by developing multiple algorithms and methods aimed at improving the distribution of data traffic within networks.

In this context, one study [8] classified load balancing in SDNs based on the data level into two main categories:

- [1] Server load balancers: These focus on distributing data traffic among available servers to improve their performance.
- [2] Link load balancers: These distribute data traffic across different network links to ensure optimal bandwidth utilization and reduce congestion on network links.

In this article, we will adopt a comprehensive perspective on the studies that have addressed server load balancers as a key area. The reason behind this focus is the pivotal role servers play in processing data traffic and their direct impact on quality of service. Servers are considered the core component of modern networks, and distributing the load on them contributes to improved resource utilization and reduced overall system response time.

We will explain and analyze several methodologies, techniques, and algorithms used in research to achieve server load balancing. What distinguishes this study from others is that it provides a comprehensive view of each method by analyzing it from all possible dimensions. We have addressed each algorithm from all technical aspects, including: the

mechanism of action, how the algorithm was designed, the steps it follows to achieve load balancing, the topology used, the parameters measured, the tools used from network simulators and performance measurement tools, the algorithms compared with them, the future prospects, and the points mentioned by each study to improve performance or expand the use of the algorithm in future. The goal is to provide researchers with a deep understanding that will help them choose or improve the methods used in server load balancing within software-defined networks. This article will identify current challenges and future opportunities in this field, providing a starting point for future research.

Before delving into previous research studies, we will review the basic load balancing algorithms, which are often used as benchmarks in most research. We will also review the most used performance metrics for evaluating the efficiency of algorithms.

Basic Load Balancing Algorithms

- Random Algorithm

In this algorithm, the Load Balancer arbitrarily chooses one server node for forwarding the request from client nodes [9].

- Round Robin Algorithm

With this method, the load balancer distributes incoming requests across multiple servers using a round robin strategy [9].

- Weighted Round Robin Algorithm

The Weighted Round Robin maintains a cyclical request distribution similar to the traditional Round Robin, but allocates a larger portion of the requests to servers with higher specifications, based on predefined weights [9].

The Most Used Performance Metrics

- Response Time

It refers the duration required to handle and reply to incoming requests [10].

- Throughput

It is the amount of data transferred per second [11].

- Transaction Rate

It expresses the number of requests processed per second [11].

III. EXPLORING STRATEGIES FOR SERVER LOAD BALANCING IN SDN

In this article, we will conduct a comprehensive analysis and review of many scientific papers that address the topic of server load balancing in software-defined networks. This information will then be summarized in a clear and organized table that facilitates reviewing the main points of each study.

Hamed et al. [12] proposed a load balancing technique between servers in an SDN architecture that distributes traffic across available servers based on the least number of active TCP server connections (connection-based). The proposed algorithm was compared with round-robin and random in two environments: the first using Mininet with a topology of 3 hosts, one of which works as a client and the rest as web servers, with an OpenFlow switch and a Ryu controller as shown in Figure 3.

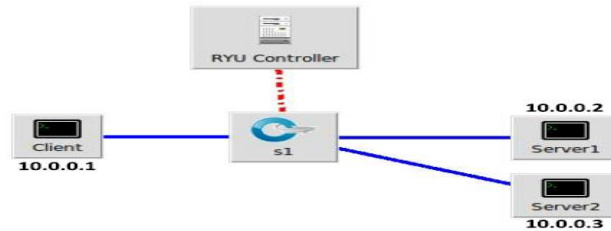


Fig. 3. First topology used in [12,13]

The second using a Raspberry Pi testbed with Open vSwitch installed to be used as an OpenFlow switch, with two laptops, one as a client and the other as a controller, running Ubuntu, respectively, with two Raspberry Pi kits running Linux as web servers, and which are all connected to an OpenFlow-enabled Raspberry Pi switch as shown in Figure 4.

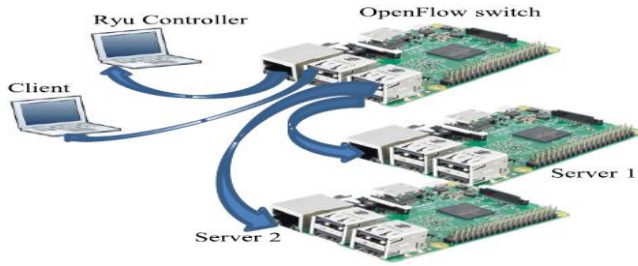


Fig. 4. Second topology used in [12,13]

To measure throughput, number of errors, and average response time, the HTTPerf and Openload tools were used. The results and experiments showed that the proposed algorithm is better than the round-robin and random strategies in terms of the measured parameters.

Hamed et al. [13] aimed to develop a new strategy to load balancing using SDN. A bandwidth-based load balancing approach for web servers is proposed, which distributes network requests among a group of servers based on the servers' bandwidth consumption. The proposed BWBLB algorithm selects the server that consumes the least bandwidth during the last 14 seconds to handle requests. The proposed technique was compared with round-robin and connection-based load balancing algorithms in two environments: the first using a Mininet emulation environment with a topology consisting of an OpenFlow switch and a Ryu controller with three hosts, one of which acts as a client and the rest as web servers, as shown in Figure 3.

The second environment used a Raspberry Pi with Open Vswitch installed to turn it into a real-time OpenFlow switch with four hosts, one of which is used as a client generating HTTP requests, and the other is used to run the load balancing application (the Ryu controller), Both run Ubuntu, with two Raspberry Pi kits used as HTTP servers to respond to client requests. The servers, the client, and the controller are all connected to an OpenFlow-enabled Raspberry Pi switch, as shown in Figure 4.

To measure throughput and average response time, the HTTPerf and OpenLoad tools were used, and the results showed that the proposed load

balancing algorithm improves throughput and response time for web servers.

Sroya et al. [14] implemented the least delay dynamic weighted round robin (LDDWRR) load balancing algorithm in an SDN environment, where the link delay was taken into account as there is a link between each server and an OpenFlow switch and a different delay was assigned to each link. The dynamic weight was then determined according to the delay so that the server with the least delay handles more weight than the other servers with the highest delay and thus handles a larger number of requests. To test the LDDWRR algorithm, a Mininet topology was created consisting of 3 web servers, one client, an OpenFlow switch and a POX controller as shown in Figure 5.

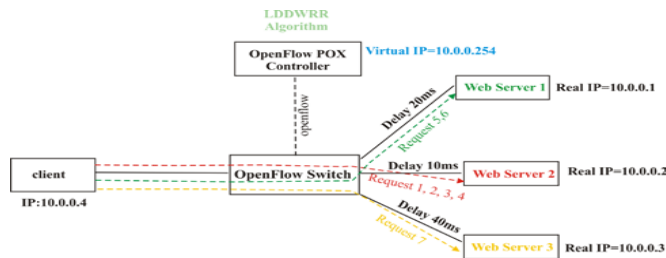


Fig. 5. Mininet topology used in [14]

For load testing, the siege tool was used and the results showed that the LDDWRR algorithm is better than the Round Robin algorithm based on transfer rate, throughput and response time.

Zhong et al. [15] designed a load balancing algorithm in an SDN environment, where the controller was used to obtain the response times of each server via Packet-Out messages, which were then to select the server demonstrating the shortest or most consistent response time. To evaluate the proposed scheme, a topology consisting of an Open Vswitch, a Floodlight controller, and three virtual machines acting as servers was used, while another virtual machine used 30 clients to access the servers. The proposed approach was compared with random, round robin, and least connection algorithms, and the results showed that the proposed algorithm (LBBSRT) yielded a better load balancing effect and processed requests with minimal average server response times. Memory usage and CPU

utilization rates were also extracted for each server, and a slight difference was observed in the memory and CPU utilization of the three servers in the proposed approach, i.e., the resources of all servers were utilized almost equally compared to other algorithms.

Li et al. [16] proposed a fuzzy logic-based load balancing strategy (LBSFL) for SDN networks. Initially, the correlation between several parameters affecting load balancing (processor utilization, memory utilization, and I/O utilization) is analyzed, and the load of multiple virtual servers is obtained using a fuzzy logic algorithm. Then, the load status of the virtual servers is examined in real time, and the lightest virtual server is selected to handle the request. If necessary, the server's sleep/restart policy is adjusted. To validate the proposed load balancing algorithm, a Mininet topology with an OpenFlow switch and an OpenDaylight controller with four web servers and a client is used, as shown in Figure 6.

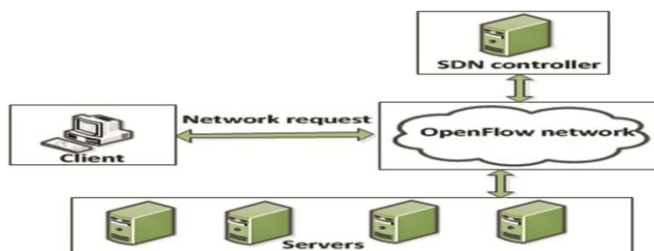


Fig. 6. Test network in [16]

The Iperf tool was used to test the network performance. The efficiency of LBSFL was compared with weighted round robin (WRR) and weighted least connections (WLC) algorithms. Experiments showed that LBSFL had the lowest average server response times, and that LBSFL had a better load balancing effect in terms of processor utilization, memory utilization, and I/O utilization than other algorithms.

Joshi et al. [17] compared different load balancing algorithms among a set of servers in a software-defined networking environment. The compared algorithms were the round-robin algorithm, the server load balancing algorithm, in which the server with the lowest CPU load is selected to handle the next request; and the flow statistics-based load

balancing algorithm, in which the server with the fewest flow connections is selected to handle the next request. These algorithms were also proposed in other studies. These algorithms were compared using a Mininet emulator with a topology consisting of a POX controller (as a load balancer), an OpenFlow switch, and a group of hosts, some acting as clients and others as servers. The results were tested using the following tools: Siege and load balancer sniper. The findings indicated that the server load balancing algorithm outperformed others in both throughput and transaction rate. Regarding response time, the flow statistics-based load balancing algorithm showed better response time compared to other algorithms.

Soleimanzadeh et al. [18] proposed a new load balancing mechanism for web server farms based on a centralized SDN controller that periodically collects information about switch port traffic and server response time and subsequently directs each incoming flow to the most suitable web server. The suggested algorithm is structured around three main modules: one that monitors sever response times and another that tracks traffic on switch ports, both operating at consistent intervals of two seconds. The third component is the best server locator, which selects the most suitable server based on the statistics obtained using the previous two components. To evaluate the performance, a Mininet emulator topology consisting of 30 client nodes, four server nodes, one Floodlight controller, and one switch was used. To generate HTTP requests, a custom application was written in .NET Core. The proposed SD-WLB approach was compared with random and round robin algorithms and the load balancing algorithm based on server response time (LBBSRT). The proposed strategy achieved higher throughput and lower response time compared to other load balancing methods. The processor and memory utilization rates for each server were extracted during the execution of each algorithm. It was found that there was a slight difference in memory and processor utilization for the four servers in the proposed approach. This means that the resources of all servers were used almost equally, and no server used its resources significantly different from the others, meaning that the

available resources were used almost equally. As future work for this paper, the following are proposed:

- Consider server load and status and incorporate these statistics into the workflow to make better decisions.
- Consider the time intervals for aggregated server response time and switch port traffic, as well as the weights α and β used in the equation to calculate the best server dynamically and adaptively.
- Leverage previously recorded metrics-such as server response times, switch port utilization, and various network parameters-as input datasets for machine learning algorithms.

Montazerolghaem et al. [19] presented a new architecture for SIP (Session Initiation Protocol) networks, where an SDN-based framework was introduced to perform load balancing between SIP servers, i.e., controlling SIP overload through an SDN approach instead of using traditional load balancers. The SIP is a signaling protocol for handling a variety of applications, including Voice over Internet Protocol (VoIP) and Instant Messaging (IM), for establishing and terminating calls. SIP architecture primarily involves two essential elements: user agents and servers. Within this protocol, user agents initiate connection requests directed to servers. However, due to the finite processing capacity of SIP servers, a high volume of user agents can potentially lead to server congestion. Therefore, the proposed framework was evaluated using three load balancing algorithms: random, round robin, and least request, with a Mininet topology consisting of a Floodlight controller, a switch, three servers, and a number of user agents as shown in Figure 7.

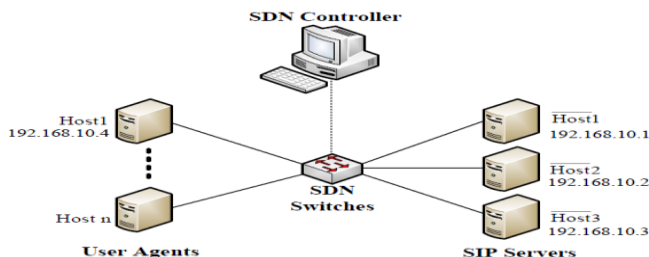


Fig.7. Topology used in [19]

Performance evaluation relied on two primary indicators: average response time and throughput. The results showed that the least request method has a lower average response time and higher throughput than random and round robin.

Aguilar et al. [20] evaluated how effective Software-Defined Networking (SDN) is in enhancing load balancing mechanisms. The proposed load balancer implemented three different load balancing algorithms: random, round robin, and least bandwidth. To evaluate the effectiveness of the load balancer, a Mininet topology consisting of a switch, a POX controller, a host acting as a client, and a varying number of servers was used. Load generation and response time measurement were conducted using Apache ab tool. Performance testing demonstrated that the server farm, when managed by the SDN-based load balancer, maintained response times under 0.1 seconds—well within acceptable user thresholds—highlighting the superior adaptability and effectiveness of SDN-driven load balancing over traditional hardware-based solutions. The experiments also showed that the least bandwidth method had a generally low response time compared to other algorithms.

Linn et al. [21] compared the Round Robin and Weighted Round Robin load balancing strategies in software-defined networks. A topology consisting of a POX controller, an OpenFlow switch, 16 clients, and three servers used as simple web servers was used using the Mininet emulator as shown in Figure 8.

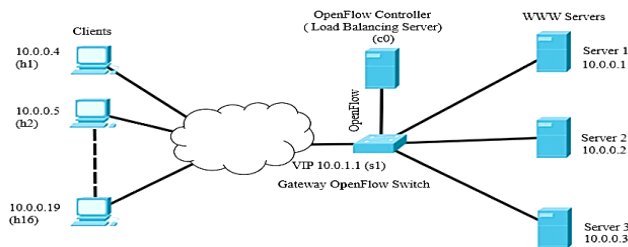


Fig. 8. Topology used in [21]

To measure the average server response time, the OpenLoad tool was used. Experiments showed that the average server response time for the Weighted Round Robin algorithm is smaller than that for the Round Robin algorithm.

Faraj et al. [22] applied the least packet load algorithm, which is used in traditional load balancers using a POX controller to distribute the load among servers in an SDN environment by directing client requests to the servers that are currently receiving the least number of packets. The proposed algorithm was compared with random, round robin, weighted round robin, least connection, and least bandwidth algorithms using a Mininet emulator with a three-server topology and a different number of clients similar to Figure 9.

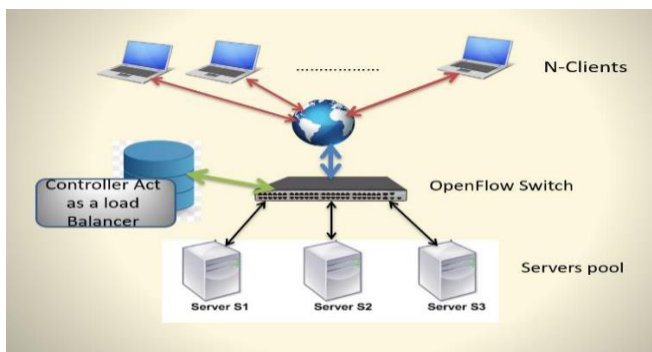


Fig. 9. Network topology used in [22]

The Iperf tool was used to generate TCP and UDP traffic from clients to servers and then calculate the throughput on each server. The results showed that the proposed algorithm outperforms all other algorithms in terms of throughput, especially when the network load is high.

Utomo et al. [23] implemented load balancing on a software-defined network using the Naïve Bayes algorithm. The Naïve Bayes algorithm was designed to obtain the server status based on data received from the Psutil agent and distribute traffic based on the current server status. After the algorithm receives input variables (processor usage, memory usage, disk usage) from the agent, computational operations are performed on them to convert them into a linguistic form based on a pre-defined dataset of 27

states to obtain the server status. After obtaining the server status, traffic is distributed based on the least resource usage of the server by the controller. To test the performance of the Naïve Bayes algorithm, a Mininet topology was used consisting of a POX controller, one switch, three hosts acting as servers, and one host acting as a client, as shown in Figure 10.

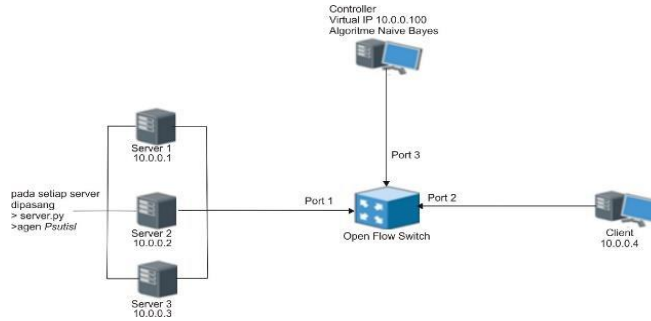


Fig.10. Network topology used in [23]

The test results showed that the Naïve Bayes algorithm performs better than the Fuzzy algorithm in terms of response time, throughput, memory usage, and CPU usage, as the Fuzzy algorithm contains more complex calculations compared to the Naïve Bayes algorithm.

Fancy et al. [24] introduced the TA-ASLB algorithm, a traffic-aware and adaptive approach designed to distribute incoming flows efficiently across servers within a data centre environment. The proposed load balancing operates based on flow size, checking the type of incoming flow and classifying it as a mice flow or an elephant flow. The algorithm is used only for elephant flows. The suggested approach is structured into three phases: monitoring network traffic, making adaptive decisions, and selecting the appropriate server. In the traffic monitoring stage, the controller monitors the entire topology. The proposed method takes into account three important details: the size of each flow, the remaining bandwidth on all links to the servers, and the computational capacity of all available servers. In the second stage, if the flow size is less than a threshold of 10 KB, a decision is made to forego the use of complex load balancing algorithms. Therefore, load balancing is carried out using the

round-robin technique. When the data flow exceeds a predefined threshold, the TA-ASLB mechanism is activated. In the final step, server selection is carried out using the TA-ASLB algorithm. This method evaluates two main factors: the bandwidth of the links leading to the server and the server's own processing capacity. The algorithm aims to identify nodes with the greatest remaining bandwidth capacity to ensure optimal data transmission. To achieve this, the remaining bandwidth of each link is estimated by analyzing its current traffic load. The server that offers both the greatest available bandwidth and the highest capacity is then selected. The algorithm was implemented using the Mininet emulator, creating a Fat-Tree topology containing seven switches and eight hosts, two of which act as clients and the rest as servers, with a Floodlight controller, as illustrated in Figure 11.

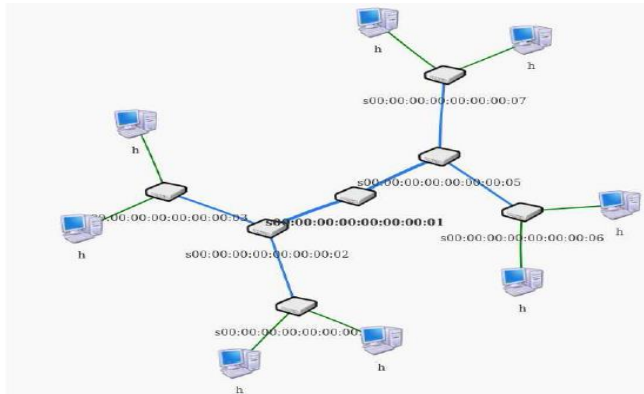


Fig. 11. Network topology used in [24]

To verify the effectiveness of the proposed algorithm, it was compared with round-robin, weighted round-robin, and statistics method based on the remaining bandwidth of the node in terms of throughput measured using the Iperf command and round-trip time measured using the Ping command. Compared to alternative algorithms, the proposed solution achieved higher data transfer rates and lower. The paper suggests future work on:

- Using neural networks in the server selection process, which can suggest optimal solutions, making it a better choice in the future.

- Dynamically determining the threshold based on the current network status, rather than using a static flow classification method.

It explained that this allows for an accurate model to be created that focuses on the current status of machines, such as the number of flows currently waiting to be processed, the number of flows already processed, server capacity, remaining bandwidth on the links connected to the servers, queue length, arrival rate, etc. Focusing on these issues improves the efficiency of current load balancing systems.

Krishna et al. [25] proposed a machine learning-based load balancing system. The architecture includes a Ryu controller, a selector unit, multiple clients, and a set of servers. The controller receives the IP address of the optimal server from the selector module and accordingly forwards the request to the optimal server in the network. The selector is where the trained machine learning model is integrated. It takes trip time, RAM usage, and CPU usage as inputs and outputs the best type of requests that the server can handle with those given statistics. The model was trained on a dataset containing 200 data points, where each data point is a set containing trip time, RAM usage, CPU usage, and a flag that is either 1 or 0 corresponding to the request type (media or text). However, it was not mentioned how the dataset was generated or created. The machine learning model employs logistic regression to train and predict the optimal request type to be handled based on the given features. Clients may request data in the form of media or plain text. The server runs two processes in parallel. One is responsible for processing the request received from the client by the controller, and the second process involves sending server information to the selector module, which uses this information as input to determine the best IP address for any type of request. The experimental network setup includes a controller and five hosts connected to a single switch, where three hosts function as clients and the remaining two serve as servers. As for the results, they only showed that their machine learning model demonstrated an accuracy of 88% in selecting the optimal server for a given type of request. Since the machine learning model, they used in their

paper took trip time, RAM usage, and CPU usage as inputs, the researchers suggest as future work for this paper is to improve the model by adding more features and considering a greater number of request types.

Sobih et al. [26] evaluated the performance of a set of load balancing algorithms in software-defined networks. These algorithms include the random algorithm, the Round Robin algorithm, the weighted Round Robin algorithm, the least-delay weighted dynamic Round Robin algorithm, the least-connection algorithm, and the weighted least-connection algorithm. Evaluation of the six algorithms focused on transaction rate and response time in a network setup where all servers shared identical configurations. Then, measurements of response time and transaction rate were conducted after modifying network conditions through the insertion of variable delays on the links connecting the switch to the server. These parameters were then measured by changing the server specifications, specifically their CPU speed. In the final scenario, the performance of the algorithms was evaluated with an additional load directed to one of the servers only. All of these scenarios were implemented by building a network consisting of a POX controller, an OpenFlow switch, three hosts acting as web servers, and several hosts acting as clients, as shown in the figure 12.

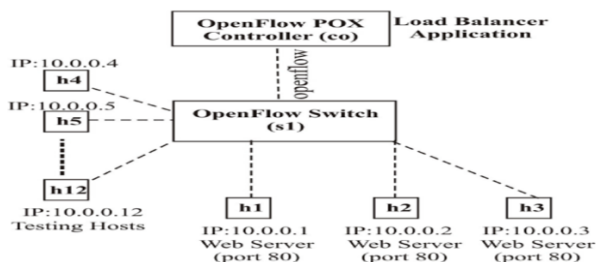


Fig. 12. The network used in [26]

To evaluate web server performance, the Httpperf utility was employed to simulate HTTP traffic by generating request loads. Experiments have shown that the response time of static algorithms is less than the response time of dynamic algorithms when the number of requests in the network is small. However, when the network conditions are fixed and predetermined, the static algorithms (Round Robin, Least Delay, Weighted Round Robin,

Weighted Round Robin) that take these conditions into account can outperform the two dynamic algorithms. While the random algorithm was unable to provide good performance in any case when the number of requests per second is large, it provided relatively good performance when the number of requests per second is small (less than 60 requests in the applied scenarios). As for the two dynamic algorithms (least connections, weighted least connections), they can adapt to different network conditions and provide relatively good performance and can outperform the rest of the static algorithms under fluctuating network conditions.

Alssaheli et al. [27] analysed the performance of SDN in balancing the load among a group of servers using the POX controller, which pre-contains the IP_loadbalancer component, which is a simple load balancer that randomly directs new flows to the server addresses in three scenarios with different numbers of servers and hosts, as shown in Figures 13, 14, and 15.

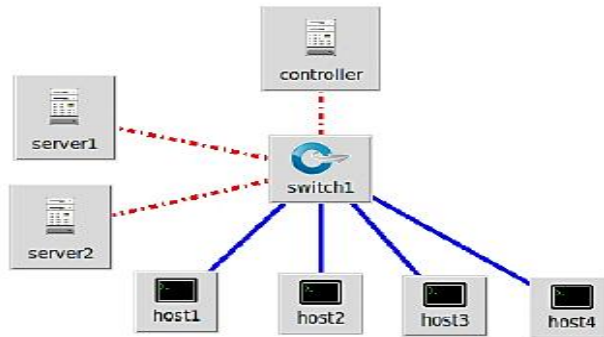


Fig.13. First scenario in [27]

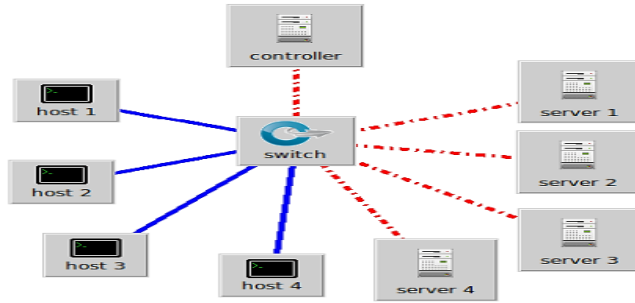


Fig. 14. Second scenario in [27]

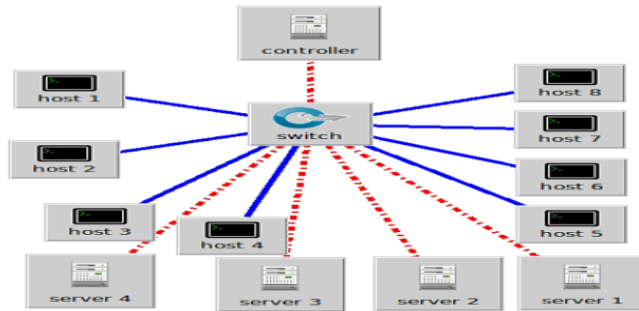


Fig. 15. Third scenario in [27]

These scenarios were implemented using the Mininet emulator, and in each scenario, throughput, delay, and Jitter were calculated. The findings indicated that employing SDN for load balancing leads to a better scheduling mechanism.

Vishwakarma et al. [28] compared different load balancing algorithms in an SDN based data center network such as round robin, random, weighted round robin, least connection and equal load using Mininet emulator with a topology of 4 legacy switches and an OpenFlow switch connected to POX with a group of hosts some of which are working as servers and others as clients as in Figure 16.

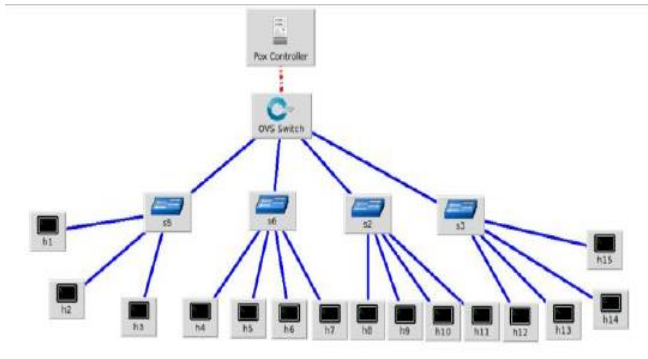


Fig. 16. Test topology used in [28]

The network's performance was evaluated through the deployment of the Siege testing utility. The test focused on the parameters of throughput, average response time and transaction rate. The results showed that the weighted round robin method outperforms all other presented algorithms. Given that average response time, transaction rate and throughput were used as performance metrics, the researchers proposed exploring additional parameters such as bandwidth usage and server utilization as part of future work.

Hamdan et al. [29] proposed to implement load balancing in SDN using linear regression. The dataset for modeling the linear regression function contains thirty samples. The constructed dataset includes key server metrics, specifically memory and CPU usage. The model uses processor and memory usage as input variables to forecast the number of requests for each server. The model is trained to associate resource consumption levels with the corresponding request load per server. The controller selects the server with the fewest requests to process the request. To read resource metrics from the server, this research employs the Psutil util. Server metrics are continuously transmitted to the controller, and the resource information collected via Psutil is stored in the dataset, where the last five samples in the dataset are updated every five seconds. For testing purposes, an SDN topology using a Mininet emulator was used. It consists of a POX controller acting as a load balancer using linear regression

calculations, another POX controller for forwarding, six switches, six clients, and two servers with varying specifications, as shown in Figure 17.

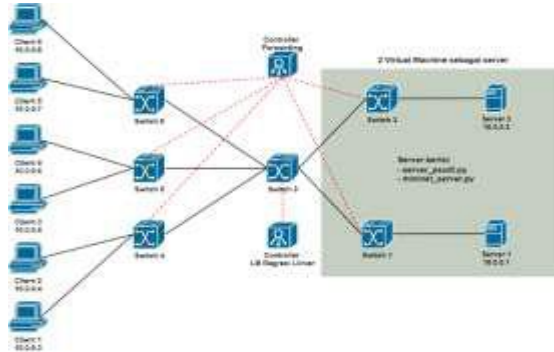


Fig. 17. SDN topology used in [29]

The HTTPPerf tool was run on one of the clients. Testing was performed in scenarios with rates of 100, 150, 200, 250, and 300 requests per second. Each scenario was executed three times, with a total of 5,000, 7,500, and 10,000 requests. This research compared linear regression with a round robin algorithm, and the results showed that linear regression and round robin performed relatively similarly in response time testing when the request rate was low. As the request rate increased, linear regression outperformed round robin in response time performance. In throughput testing, linear regression and round robin have a similar pattern as the response time test. When the rate used is small, linear regression and round robin have relatively similar performance. However, linear regression outperforms round robin when the rate used is higher. In the CPU utilization test, linear regression performs better than round robin. During the linear regression analysis, CPU usage peaked at 75 percent. Meanwhile, round robin reached 98 percent. As future work for this paper, the following are proposed:

- Increase the accuracy of linear regression calculations by adding more samples to the dataset.
- Improving the program in terms of the time when linear Regression calculations are performed so that they are computed only if there is a change in the server's information.

Malavika et al. [30] focused on extending SD-WLB and LBBSRT by implementing the closed-loop control theory (CLCT) (this theory means measuring the controlled variable and comparing it to a setpoint that represents the desired performance [35]) with two threshold values L and U to adaptively adjust the timing of server probe. Both SD-WLB and LBBSRT probe servers in the server farm at a uniform and specific time interval to retrieve the factors of server response time and switch port traffic volume. Therefore, the task of the proposed framework LBBNNC was how the controller recovers these factors at variable time intervals by defining the application delay tolerance as threshold values L and U to distinguish the server load status as low, medium, and high based on the maximum response time difference:

$MRTD = MAX(\text{response time of all the servers}) - MIN(\text{response time of all the servers})$

between servers in a server farm that were less than L, between L and U, and higher than U to adaptively adjust the probe time to measure the server response and switch port traffic volume. The values of L and U were adjusted using a neural network then feeds the L and U values into a probe time generation algorithm, which, based on the L, U, and MRTD values, determines the new probe time for the servers in the next iteration. When traffic increases, the probe time is reduced, and when traffic decreases, the probe time is increased. To evaluate the performance of the proposed LBBNNC framework, a Mininet topology with one switch and one controller was used, with four hosts acting as clients and three hosts acting as Apache web servers. For testing purposes, the Httperf tool was used, using 50,000 requests at a request rate of 10 requests per second to 100 requests per second, increasing the request rate by 5 requests per second each time. The proposed scheme proved to be superior to round robin, LBBSRT, and SD-WLB in terms of the achieved request rate, average response rate, average response time, and number of produced errors.

Malbašić et al. [31] proposed a new load balancing scheme for a hybrid SDN environment. The scheme relied on continuous monitoring of server

load indicators and implementing multi-parameter metrics (CPU load, I/O read, I/O write, link upload, link download) to schedule connections. This scheme used the SNMP protocol mechanism to collect information about the resources used by the server to schedule requests. A simulated environment incorporating virtual servers and network components using EVE-NG platform [36] to facilitate testing and experimentation. The following virtual components were implemented: a Linux OVS router as an SDN-aware switch; six virtual machines with Linux Ubuntu servers (two representing web servers, three representing clients, and one representing a POX controller); and the last component represents a Mikrotik ROS router plus an internet connection, as shown in Figure 18.

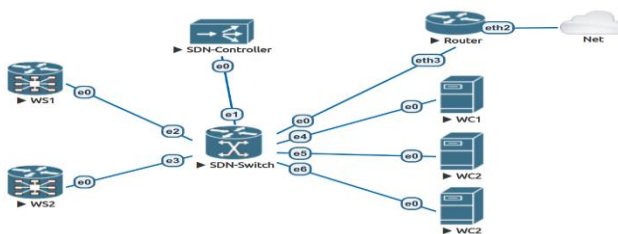


Fig. 18. Testing environment used in [31]

The proposed scheme was compared with Random, Round Robin, and LBBSRT algorithms, and the test results showed that the proposed mechanism yielded the best results in terms of balancing CPU load on servers and made better load balancing decisions than the tested mechanisms.

Singh et al. [32] implemented a Round Robin load balancing algorithm in SDN to overcome the limitations of traditional network load balancers using a Mininet topology consisting of a POX controller, an OpenFlow switch, six hosts acting as web servers, and several hosts acting as clients. Siege and Autobench were used for performance testing. The experiments showed 100% availability of services with an average of 0.028 seconds in the shortest transfer and a response time of less than 1 second.

Vani et al. [33] presented a dynamic load balancing algorithm for software-defined networks based on server response time and content

mapping. When performing load balancing among a group of servers, the controller directs requests to the respective servers in the server pool based on the load balancer result for the requested content type and response time. Server pools are organized based on content type, including dedicated pools for video, audio, images, and textual data. The proposed architecture consists of three main modules: request classification module, server monitoring module, and dynamic load balancing module. The request classification module is designed to determine the nature of incoming requests, categorizing them into formats like video, audio, image, or text. Once this information is extracted, it is sent to the load balancing module. To ensure real-time oversight, the server monitoring module is designed so that each server within the pool transmits its load metrics to the controller at 5-milliseconds intervals. Information such as CPU usage, memory usage, requests per second, time per request, transfer rate, and waiting time are sent to the controller. The dynamic load balancing module is implemented using content-based routing. When a client request is received, the content is analysed and the server with the lowest response time in each server cluster is found. The controller installs the flow based on the requested content and the server with the lowest response time. For example, if the request is related to images, it is redirected to the image server pool, etc. The experiment was conducted in a data center network where a number of different clients and web servers, such as Apach2, Ngnix, and simpleHTTPserver, were connected to the Ryu controller via a Zodiac-FX switch in real time, as shown in Figure 19.

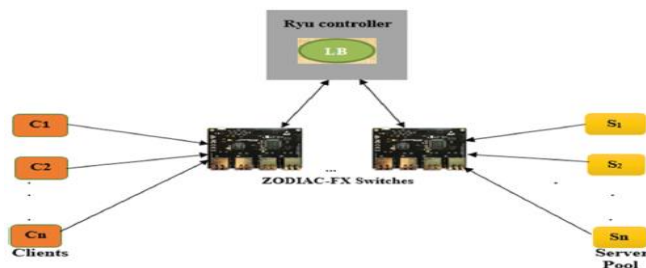


Fig. 19. Data center network used in [33]

The proposed mechanism was compared with different techniques such as round robin, random, LBBSRT (load-balancing scheme based on server response time) and SD-WLB (Web load balancing based on server statistics) based on various metrics such as average response time, transfer rate, time per request, requests per second, and waiting time. The proposed mechanism has shown better performance compared to other techniques.

Fazal et al. [34] proposed an adaptive approach to load balancing between servers for SDN-based data centre networks by identifying an optimal server and choosing the best path to the selected server. To do this, the proposed approach is divided into five modules: the first module is the probe interval generator, which determines the time interval after which servers will be checked for their performance and is inspired by the LBBCLCT (Load Balancing Based on Closed Loop Control Theory) approach. The second module is the server performance monitor, which monitors server response times and stores them in a local database, and is inspired by the LBBSRT approach. The third module is the server selector, which reads the local database of server response times and selects the server with the lowest response time as the optimal server. The fourth module is the switch port aggregator, which checks switches for the traffic volume passing through its ports and stores the data in a local database and is originally inspired by the SD-WLB approach. The fifth module, known as the path selector, will take the specified server and traffic data from the database and apply the ant colony algorithm on the network to determine the most efficient route between the sender and the most suitable server. The proposed approach was tested in a Mininet environment with a fat-tree topology, with 31 switches and 125 hosts, 12 of which were configured as servers, and the rest as hosts with a Floodlight controller, as shown in Figure 20.

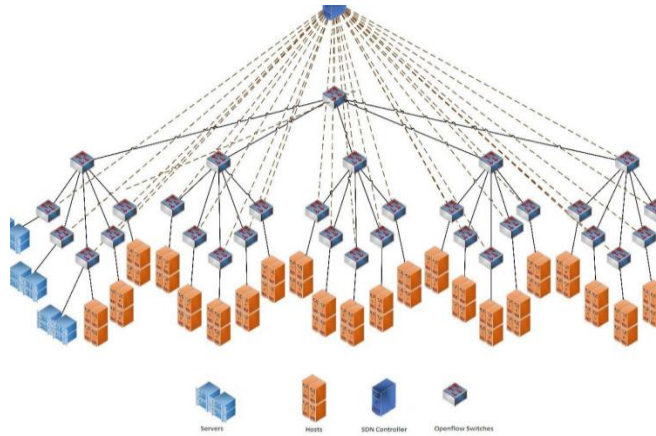


Fig. 20. Fat tree topology used in [34]

To verify the effectiveness of the proposed algorithm (SD-ALB), it was compared with round-robin, SD-WLB, LBBST, and LBCLCT algorithms. The results showed that the proposed approach outperforms in terms of throughput and server response time, but it requires a lot of computational resources.

After reviewing the articles and analyzing their methodologies, the Table 1 summarizes their key aspects to facilitate comparison and identify important research trends:

TABLE 1 SUMMARY OF PREVIOUS RESEARCH STUDIES ON SERVER LOAD BALANCING IN SOFTWARE DEFINED NETWORKS (SDN)

Reference	Year	Objective	Controller Used	Network Topology	Algorithms Compared With	Tools Used	Measured Parameters	Future Prospects
[12] Hamed et al.	2017	Proposing a technique for load	Ryu	Switch, controller, 3 hosts (1	Random, Round Robin	Mininet, Raspberry	Throughput, Response time,	–

		balancing between servers in an SDN architecture based on the least number of active server TCP connections		client, 2 servers)		Pi, Httpf, Open Load	Number of errors	
[13] Hamed et al.	2017	Bandwidth-based load balancing (BWBLB) approach for web servers	Ryu	Switch, controller, 3 hosts (1 client, 2 servers)	Round Robin, Connection-based	Mini net, Raspberry Pi, Httpf, Open Load	Through put, Response time	-
[14] Sroya et al.	2017	Implementing Least Delay Dynamic Weighted Round Robin (LDDWRR) algorithm in SDN	POX	3 web servers, 1 client, OpenFlow switch, controller	Round Robin	Mini net, Siege	Response time, Through put, Transaction rate	Using multiple controllers
[15] Zhong et al.	2018	Load balancing based on server response time (LBBSRT)	Floodlight	Open vSwitch, controller, 3 servers, 30 clients	Random, Round Robin, Least Connection	-	Response time, CPU & Memory utilization	Energy saving
[16] Li et al.	2018	Fuzzy logic-based load	OpenDaylight	OpenFlow switch, controller,	WRR, WLC	Mini net, Iperf	Response time, CPU,	-

		balancing (LBSFL)		4 servers, 1 client			Memory, I/O utilization	
[17] Joshi et al.	2019	Comparison of algorithms (Round Robin, Server Load Balancing, Flow statistics-based)	POX	Switch, controller, hosts as clients/servers	Round Robin, Server load, Flow-based	Mini net, Siege, Sniper	Through put, Response time, Transaction rate	Suggest Ryu/multi-controller
[18] Soleimanzadeh et al.	2019	SD-WLB: centralized mechanism using server stats and switch port traffic	Floodlight	30 clients, 4 servers, 1 controller, 1 switch	Random, Round Robin, LBBSRT	Mini net, Custom .NET Core app	Through put, Response time, CPU, Memory utilization	AI integration, adaptive intervals
[19] Montazerolghaem et al.	2019	SDN-based framework for SIP server load balancing	Floodlight	Controller, switch, 3 servers, user agents	Random, Round Robin, Least Request	Mini net	Response time, Through put	Practical implementation
[20] Aguilar et al.	2019	Effectiveness of SDN in load balancing	POX	Switch, controller, 1 client, servers	Random, Round Robin, Least Bandwidth	Mini net, Apache ab	Response time	-
[21] Linn et al.	2019	Comparison of Round Robin and Weighted	POX	Controller, OpenFlow switch,	Round Robin, WRR	Mini net, Open Load	Response time	-

		Round Robin		16 clients, 3 servers				
[22] Faraj et al.	2020	Least packet load algorithm in SDN	POX	Controller , 3 servers, varying clients	Random, RR, WRR, Least Conn., Least BW	Mini net, Iperf	Through put	Multi-controll er, DDoS prevent ion
[23] Utomo et al.	2020	Naïve Bayes algorithm for server status-based balancing	POX	Controller , 1 switch, 3 servers, 1 client	Fuzzy	Mini net, Psuti l	Respon se time, Through put, CPU, Memory	–
[24] Fancy et al.	2021	TA-ASLB: traffic-aware adaptive load balancing	Floodlight	Fat-tree, 7 switches, 8 hosts (2 clients, 6 servers)	RR, WRR, Bandwid th-based	Mini net, Ping, Iperf	Through put, RTT	Neural networ ks, adaptiv e threshol ds
[25] Krishna et al.	2021	Machine learning-based load balancing	Ryu	Controller , switch, 5 hosts (3 clients, 2 servers)	–	Mini net	Accurac y 88%	More features , request types
[26] Sobih et al.	2021	Evaluation of six load balancing algorithms	POX	Controller , OpenFlo w switch, 3 servers, clients	Random, RR, WRR, LDDWR, LC, WLC	Mini net, Httpe rf	Transact ion rate, Respon se time	–
[27] Alssaheli et al.	2022	Performan ce of SDN using POX IP_loadbal ancer	POX	Scenarios with varying servers & hosts	–	Mini net	Through put, RTT, Jitter	AI-based algorith ms

[28] Vishwakarma et al.	2022	Comparison of algorithms in SDN data centers	POX	4 legacy switches + OpenFlow switch, clients & servers	RR, Random, WRR, LC, Equal Load	Mininet, Siege	Response time, Transaction rate, Throughput	Include bandwidth & server utilization
[29] Hamdan et al.	2022	Linear regression for load balancing	POX	2 controllers, 6 switches, 6 clients, 2 servers	RR	Mininet, Psutil, Httpperf	Response time, Throughput, CPU utilization	Larger dataset, smarter execution timing
[30] Malavika et al.	2022	Closed-loop control theory (LBBNNC) for adaptive probing	Ryu	Switch, controller, 4 clients, 3 servers	RR, SD-WLB, LBBSRT	Mininet, Httpperf	Response rate, Response time, Errors, Request rate	Cloud computing applications
[31] Malbašić et al.	2022	Hybrid SDN with multi-parameter metrics	POX	Linux OVS router, 6 VMs (2 servers, 3 clients, 1 controller) + Mikrotik router	Random, RR, LBBSRT	EVE-NG	Transactions/sec, Avg transaction time, CPU balance	New weight distribution models
[32] Singh et al.	2022	Round Robin implementation in SDN	POX	Controller, OpenFlow switch, 6 servers, clients	-	Mininet, Siege, Autobench	Response time, Transfer rate	New LB technologies

[33] Vani et al.	2023	Dynamic LB based on server response & content mapping	Ryu	Zodiac-FX switch, clients, servers, controller	RR, Random, LBBSRT, SD-WLB	–	Response time, Transfer rate, Requests/sec, Waiting time	–
[34] Fazal et al.	2023	Adaptive load balancing (SD-ALB) with optimal path selection	Floodlight	Fat-tree, 31 switches, 125 hosts (12 servers)	RR, SD-WLB, LBBSRT, LBCLC T	Mini net, Iperf	Throughput, Response time	Deep reinforcement learning

IV. CRITICAL ANALYSIS OF PREVIOUS STUDIES: RESEARCH LIMITATIONS IN SERVER LOAD BALANCING IN AN SDN ENVIRONMENT

Many studies have employed artificial intelligence techniques in the server selection process within SDN networks, with the aim of improving load distribution and increasing performance efficiency. However, the lack of a benchmark dataset that simulates realistic environments poses a significant challenge. This lack has prompted many researchers to generate synthetic datasets specifically designed to meet the study's requirements, which may limit the generalizability of results outside of experimental settings.

It is also noted that the proposed algorithms are often compared to the underlying algorithms. This is due to the unavailability of source code for most algorithms, which hinders comprehensive evaluation within the same test environment.

From a design perspective, most studies rely on a simplified topology that includes a single OpenFlow switch connected to servers, clients, and a controller. Despite the simplicity of this model, it is considered functionally appropriate because the switch directly connected to the servers is responsible for receiving flow balancing rules from the controller

and distributing requests based on them. Furthermore, some modern high-performance switches support connecting more than 20 servers, enhancing the realism of the proposed model without the need for more complex topologies.

Although there has been growing interest in load balancing concepts within the SDN environment in recent studies for 2024 and 2025, most of this research focuses on improving quality of service or managing routing without directly addressing the server load balancing problem itself. This highlights a clear research gap and indicates the need for more specialized studies on this vital aspect, especially with the increasing reliance of cloud and virtual infrastructure on SDN servers.

V. CONCLUSION AND FUTURE RECOMMENDATIONS

In conclusion, server load balancing in software-defined networks becomes one of the essential elements to ensure stability, scalability, and efficiency in the present infrastructures. This work makes a contribution by reviewing and analyzing a broad variety of algorithms, showing the advantages of dynamic and intelligent approaches compared to more traditional approaches of static approach, and the weaknesses, including the limited scalability of the approach, use of simplified experimental topologies, and lack of standardized benchmarking practices.

In practical terms, the summary of existing research offers useful information to network engineers, cloud data center operators, and SDN developers, who have to choose or create load balancing solutions depending on the performance requirements, in terms of response time, throughput and resource usage. The review, however, is limited by the fact that the studies reviewed are heterogeneous and they tend to use various testbeds, tools, and metrics, and such direct comparison with the results is not easily possible.

Going forward, the focus of future studies must be put on development of standardized evaluation frameworks, empirical benchmarking data, and adapting algorithms that would be capable of dealing with scalability and energy efficiency in addition to performance. Synthesis of artificial intelligence and machine learning into SDN controllers are also a

promising avenue particularly in an environment with dynamic and unpredictable traffic. To make sure that SDN-based systems are strong and sustainable as the demands on the global networks increase, the multi-controller architecture and energy-conscious designs deserve further attention as well.

REFERENCES

- [1].F. ONGARO, "Enhancing Quality of Service in Software-Defined Networks", UNIVERSITY OF BOLOGNA, 2014.
- [2].B.N. Astuto, M. Mendonça, X.N. Nguyen, K. Obraczka, T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks", Communications Surveys and Tutorials, IEEE Communications Society, 2014.
- [3].R. Wang, D. Butnariu, J. Rexford, "OpenFlow-Based Server Load Balancing Gone Wild", Princeton University, 2011.
- [4].N. Handigol, S. Seetharaman, M. Flajslik, N. McKeown, R. Johari, "Plug-n-serve: Load-balancing web traffic using OpenFlow". ACM SIGCOMM Demo, Barcelona, Spain 2009.
- [5].W. Xia, Y. Wen, C.H. Foh, D. Niyato, H. Xie, "A Survey on Software-Defined Networking", IEEE COMMUNICATION SURVEYS & TUTORIALS, 2015.
- [6].V. Khatri, "Analysis of Openflow Protocol in Local Area Networks", M.S. thesis, Tampere Univ. of Technology, 2013.
- [7].G. Altangerel, T. Chuluuntsetseg, D. Yamkhin, "Performance analysis of SDN controllers: POX, Floodlight and OpenDaylight", the 13th International Forum on Strategic Technology, 2018.
- [8].M.C. Saxena, M. Sabharwal, P. Bajaj, "Review of SDN-based load-balancing methods, issues, challenges, and roadmap", International Journal of Electrical and Computer Engineering Systems, 2023.
- [9].S. K. Vishwakarma, H. Pavithra, "Performance Analysis of different Load Balancing Algorithms in SDN Based Data Center Networks", International Journal of Engineering Research & Technology, 2022.
- [10]. Httpperf, "httpperf", <https://github.com/httpperf/httpperf?tab=readme-ov-file>
- [11]. Btfak, "sniper", <https://github.com/btfak/sniper>
- [12]. M.I. Hamed, B.M. ElHalawany, M.M. Fouda, A.S. Tag Eldien, "Performance Analysis of Applying Load Balancing Strategies on

- Different SDN Environments", Benha Journal of Applied Sciences, Egypt, 2017.
- [13]. M.I. Hamed, B.M. ElHalawany, M.M. Fouda, A.S. Tag Eldien, " A new approach for server-based load balancing using software-defined Networking", The 8th IEEE International Conference on Intelligent Computing and Information Systems, 2017.
- [14]. M. S. Sroya, V. Singh, " LDDWRR: Least Delay Dynamic Weighted Round-Robin Load Balancing in Software Defined Networking", International Journal of Advanced Research in Computer Science, 2017.
- [15]. H. Zhong, Y. Fang, J. Cui, "Reprint of LBBSRT: An efficient SDN load balancing scheme based on server response time", Future Generation Computer Systems, 2018.
- [16]. G. Li, T. Gao, Z. Zhang, Y. Chen, " Fuzzy Logic Load-Balancing Strategy Based on Software-Defined Networking", Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST) 2018, Springer Nature Switzerland AG, as part of WiCON 2017.
- [17]. N. Joshi, D. Gupta, " A Comparative Study on Load Balancing Algorithms in Software Defined Networking", Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST), Springer Nature Switzerland AG, 2019.
- [18]. K. Soleimanzadeh, M. Ahmadi, M. Nassiri, "SD-WLB: An SDN-Aided Mechanism for Web Load Balancing Based on Server Statistics", ETRI Journal, 2019.
- [19]. A. Montazerolghaem, "SIP Server Load Balancing Based on SDN", Quchan University of Technology, Iran, 2019.
- [20]. L. C. F. P. Aguilar, D. M. Batista, " Effectiveness of Implementing Load Balancing via SDN ", Brazilian Symposium on Computer Networks and Distributed Systems, 2019.
- [21]. A. S. Linn, S. H. Win, S. T. Win, "Server Load Balancing in Software Defined Networking", National Journal of Parallel and Soft Computing, 2019.

- [22]. M. K. Faraj, A. Al-Saadi, R. J. Al-Bahadili, " An Investigation of Using Traffic Load in SDN Based Load Balancing", Iraqi Journal of Computer Communication Control and System Engineering, 2020.
- [23]. A. A. Utomo, P. H. Trisnawan, R. Primananda, "Server Load Balancing Mechanism Using Naive Bayes Method with Psutils Agent on Software Defined Network", Journal of Information Technology and Computer Science Development, 2020.
- [24]. C. Fancy, M. Pushpalatha, "Traffic-aware adaptive server load balancing for software defined networks", International Journal of Electrical and Computer Engineering Systems, 2021.
- [25]. P. M. Krishna, S. S. Rao, S. S. Bhurat, T. Venkatesh, K. A. Vani, " Machine Learning to Balance Web Applications using SDN", International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences, 2021.
- [26]. M. Sobih, I. Khalil, "Performance evaluation of controllers in software defined networks", Tartous University Journal for Research and Scientific Studies, 2021.
- [27]. O. M. A. Alssaheli, Z. Z. Abidin, N. A. Zakaria, Z. A. Abas, " Software Defined Network based Load Balancing for Network Performance Evaluation", International Journal of Advanced Computer Science and Applications, 2022.
- [28]. S. K. Vishwakarma, H. Pavithra, " Performance Analysis of different Load Balancing Algorithms in SDN Based Data Center Networks", International Journal of Engineering Research & Technology, 2022.
- [29]. A. A. Hamdan, P. H. Trisnawan, F. A. Bakhtiar, "Implementation of Load Balancing using Linear Regression Method on Software Defined Network", Journal of Information Technology and Computer Science Development, 2022.
- [30]. R. Malavika, M. L. Valarmathi, "Adaptive Server Load Balancing in SDN Using PID Neural Network Controller", Computer Systems Science & Engineering, 2022.
- [31]. T. Malbašić, P. D. Bojović, Z. Bojović, J. Šuh, D. Vujošević, "Hybrid SDN Networks: A multi-parameter server load balancing scheme", Journal of Network and Systems Management, 2022.

- [32]. I. T. Singh, T. R. Singh, "Server Load Balancing with Round Robin Technique in SDN", International Conference on Decision Aid Sciences and Applications (DASA), 2022.
- [33]. K. A. Vani, K. N. RamaMohanBabu, " An Intelligent Server load balancing based on Multi-criteria decision-making in SDN", International Journal of Electrical and Computer Engineering Systems, 2023.
- [34]. R. Fazal, S. M. M. Gilani, M. J. Khalid, " SD-ALB: Software Defined Adaptive load balancing in Data Center Network", KIET Journal of Computing & Information Sciences, 2023.
- [35]. ScienceDirect, "Closed Loop Control", <https://www.sciencedirect.com/topics/engineering/closed-loop-control>.
- [36]. EVE-NG, "Emulated Virtual Environment Next Generation", <https://www.eve-ng.net>.
- [37]. Y. Ghazi, A. Mustafa, "Improving Single-Controller Software-Defined Networks Architecture Using Named Data Networks Architecture", Homs University Journal, 2025.
- [38]. M. Abboud, M. Aldaher, " Performance Analysis in software defined networks (SDNs)", Homs University Journal, 2019.