تعزيز الأمن السيبراني في الزمن الحقيقي:تأثير حجم تسلسلات استدعاءات النظام على اكتشاف الاقتحام

عمار مصطفى* كندة أبو قاسم ** محمد حجازية ***

(الإيداع : 30 نيسان 2025، القبول: 14 تموز 2025)

الملخص:

اكتسبت تقنية الحاويات في الآونة الأخيرة قيمة وشعبية هائلة، خاصةً بين المطورين وشركات التكنولوجيا. أصبح الكشف عن الهجمات الإلكترونية في الوقت الحقيقي مطلباً رئيسياً، خاصةً عندما تصبح الحاويات جزءاً من البيئة السحابية. تُستخدم أنظمة الكشف عن التسلل للبحث عن أي نشاط مشبوه يؤثر على التطبيقات المعبأة في حاويات، سواء كانت تعمل في بيئة سحابية أو بشكل مستقل. وبعبارة أخرى، فإن أي محاولات لتغيير سلوك الحاويات أو التطبيقات التي تعمل داخلها لن تساعد في الهجوم أوالكشف عن الشذوذ. تكمن مساهمة هذه الدراسة في طريق التحقيق في تأثير حجم تسلسل استدعاءات النظام على اكتشاف الهجوم. من خلال تحريك النوافذ بأحجام مختلفة. قيمت الدراسة على مجموعة بيانات LID-DS لدراسة تأثير التتبع الجزئي مقابل التتبع الكامل على اكتشاف الشذوذ. حللت الدراسة أيضًا كيفية تأثير التغييرات في أحجام النوافذ المنزلقة على أداء المصنف في الكشف عن الشذوذ. استخدمت مجموعة من خوارزميات التعلم الآلي. حققت خوارزمية الغابة العشوائية الغابة نجاحا في جميع المقاييس وأحجام النوافذ. حققت حجم النافذة w=14 أفضل أداء، أي أن حجم النافذة 14 هو حجم النافذة الأنسب لاكتشاف الهجمات.

الكلمات المفتاحية: تعلم الآلة، استدعاءات النظام ، أنظمة كشف الاقتحام، النافذة المنزلقة.

* طالب دراسات عليا (دكتوراه)، قسم هندسة الحاسبات والتحكم الالي، كلية الهندسة الكهربائية والميكانيكية، جامعة اللاذقية.

** أستاذ مساعد، قسم هندسة الحاسبات والتحكم الالي، كلية الهندسة الكهربائية والميكانيكية، جامعة اللاذقية، اللاذقية، سورية.

*** أستاذ، قسم هندسة الحاسبات والتحكم الالي، كلية الهندسة الكهربائية والميكانيكية، جامعة اللاذقية، اللاذقية، سورية.

Boosting real-time cybersecurity: The Influence of the size of system call sequences on intrusion detection

AMMAR MOUSTAFA * KINDA ABU KASSEM ** MOHAMMED HEJAZIEH ***

(Received: 30 April 2025, Accepted: 14 July 2025)

Abstract :

Container technology has recently gained tremendous value and popularity, especially among developers and technology companies. Detecting cyberattacks in real-time has become a key requirement, especially when containers become part of a cloud environment. Intrusion detection systems are used to look for suspicious activity affecting containerized applications, whether they are running in the cloud or standalone. In other words, any attempts to change the behavior of containers or the applications running inside them will not help with attack or anomaly detection. The contribution of this study lies in the way it investigates the effect of the size of the system call sequence on attack detection. By moving windows of different sizes. The study evaluated on the LID-DS dataset to study the effect of partial versus full trace on anomaly detection. The study also analyzed how changes in the sizes of the sliding windows affected the classifier's performance in anomaly detection. It used a combination of machine learning algorithms. The Random Forest Random Forest algorithm achieved success across all metrics and window sizes. The window size w=14 achieved the best performance and most suitable window size for detecting attacks.

Keywords: Machine Learning, System calls, Intrusion detection systems, sliding windows

* PhD Student, Computers and Automated Control Department, Faculty Mechanical & Electrical Engineering, LATAKIA University,

** Professor, Computers and Automated Control Department, Faculty of Mechanical & Electrical Engineering, LATAKIA University, Lattakia, Syria.

*** Professor, Computers and Automated Control Department, Faculty of Mechanical & Electrical Engineering, LATAKIA University, Lattakia, Syria.

المقدمة

كيف يمكن لحجم تسلسل استدعاءات النظام أن يفتح المجال لاكتشاف الاقتحامات بكفاءة أكبر في البيئات المكوّنة من الحاويات؟ مع استمرار شهرة تقنيات الحاويات بين المطورين وشركات التكنولوجيا، أدى تكاملها في البيئات السحابية إلى زيادة الحاجة إلى آليات قوية للكشف عن الهجمات الإلكترونية في الزمن الحقيقي [A][B]. برزت أنظمة الكشف عن التسلل (IDS) كأداة حاسمة لتحديد الأنشطة المشبوهة داخل التطبيقات المكوّنة في حاويات، سواء كانت تعمل بشكل مستقل أو ضمن بنية تحتية سحابية. ومع ذلك، تطرح الطبيعة الديناميكية والمعقدة للحاويات تحديات كبيرة أمام مناهج الكشف عن الشذوذ التقليدية، لا سيما في التقاط أنماط الهجوم الخفية المضمنة في تسلسل استدعاءات النظام. يهدف هذا البحث إلى معالجة هذه التحديات من خلال استكشاف تأثير أحجام تسلسل استدعاءات النظام على أداء أنظمة كشف الأقتحام. باستخدام نهج النوافذ المنزلقة بأحجام متفاوتة ومجموعة بيانات LID-DS [C]، تقيّم هذه الدراسة تأثير الآثار الجزئية مقابل الآثار الكاملة على اكتشاف الهجمات وتستكشف كيفية تأثير الاختلافات في حجم النافذة على أداء المصنف. من خلال الاستفادة من خوارزميات تعلم الآلة، نسعى من خلال هذا البحث الى تحسين عملية اكتشاف الشذوذ في الزمن الحقيقي من خلال استكشاف الحجم الأمثل لتسلسلات استدعاء النظام الذي يوفر معلومات كافية لاكتشاف وبالنتيجة تعزيز كفاءة اكتشاف الهجمات في بيئات الحاويات.

1- مشكلة البحث:

تلخص مشكلة البحث في النقاط الآتية:

- تأثير حجم تسلسل استدعاء النظام:

✧ يؤثر على دقة الكشف، استهلاك الموارد، سرعة الكشف، القدرة على التكيف، تحليل العلاقات، ونسبة الإيجابيات الخاطئة في أنظمة الأمن السيبراني.

- التسلسلات الطويلة:

✧ توفر معلومات تفصيلية عن سلوك العمليات، مما يحسن دقة الكشف عن الحالات الشاذة، تساعد في تحليل العلاقات بين استدعاءات النظام، مما يكشف عن هجمات معقدة، تقلل الإيجابيات الخاطئة بسبب توفر بيانات كافية لتحديد الأنماط الطبيعية.

✧ العيوب: تتطلب معالجة بيانات كبيرة، مما يزيد استهلاك المعالج والذاكرة، ويبطئ الكشف، وقد تكون غير عملية في بيئات محدودة الموارد.

- التسلسلات القصيرة:

✧ تسهل الكشف السريع عن الحالات الشاذة، تفتقر إلى سياق كافٍ، مما يزيد الإيجابيات الخاطئة ويقلل فرص الكشف عن الإيجابيات الحقيقية، ولا توفر معلومات كافية عن تفاعل الاستدعاءات.

التحدي الأساسي:

إيجاد توازن بين حجم تسلسل استدعاء النظام وسرعة الكشف لتحقيق كشف فعال في الزمن الحقيقي عن الهجمات والحالات الشاذة.

2- الهدف من البحث:

نهدف من خلال البحث إلى التحقق ممايلي:

⊙ دراسة تأثير أحجام مختلفة لتسلسلات استدعاءات النظام باستخدام النافذة المنزلقة على دقة الكشف للشذوذ والهجمات في الزمن الحقيقي. والجدوى من استخدام كامل الاستدعاءات أو تجاهل البعض منها على كشف الهجمات

⊙ تقييم عملية استخدام جزء من التتبع أو التتبع بالكامل وتأثيره على عملية كشف الشذوذ.

3- الدراسات المرجعية:

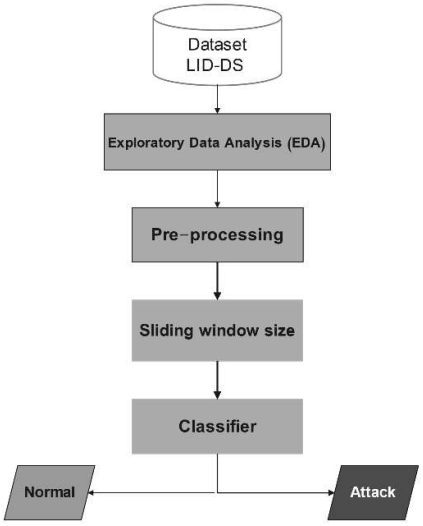
اقترح الباحث[D] آلية تجزئة ديناميكية تعتمد على استخراج تسلسلات قصيرة مميزة لتعزيز الدقة وتقليل التأخير الزمني، مما يعالج قصور النوافذ الثابتة في الأنظمة محدودة الموارد . بين الباحث[E] أهمية استخدام نوافذ أكبر (12–14) لضمان تمثيل الأنماط الموزعة زمنياً في أنظمة التحكم الصناعي، مع مراعاة الأداء الحسابي والذاكرة، كما لاحظ الباحث[F] أن ضبط حجم النافذة والعتبة بشكل تجريبي يؤثر تأثيراً كبيراً على حساسية النظام وسرعة الكشف. يتناول البحث[G] تحديات كشف الاقتحام القائم على المضيف (HIDS) في بيئات الحاويات (Containers) باستخدام استدعاءات النظام، المشكلة الرئيسية تتمحور حول عدم قدرة الأنظمة الحالية على التعامل بفعالية مع خصائص بيئات الحاويات وضعف الأداء في الزمن الحقيقي، الأنظمة التقليدية مثل تلك القائمة على التتبع الكامل تتطلب انتظار اكتمال الهجوم، مما يؤخر الكشف ويجعلها غير مناسبة لبيئات الزمن الحقيقي. اقترح الباحثون نظام HIDS جديداً قائماً على التحليل غير المراقب للكشف عن الانحرافات في الحاويات من خلال تحليل متعدد الخصائص لاستدعاءات النظام، تمثيل الاستدعاءات باستخدام هيكلية قائمة على الرسوم البيانية، التركيز على السياق. الحل المقترح يُعد تقدماً ملحوظاً في كشف الانحرافات في بيئات الحاويات، خاصة بفضل تحليل السياق واستخدام خصائص متعددة، لكنه يحتاج إلى تحسينات للتعامل مع الهجمات الخفية (Heartbleed) والهجمات المعتمدة على المعرفة المسبقة، بالإضافة إلى اختباره على بيانات حقيقية لتأكيد فعاليته في بيئات الإنتاج. ركز الباحث[A] على تطوير نظام كشف الاقتحام مصمم خصيصاً لبيئات الحوسبة السحابية من خلال تحديد تقنيات تحليل استدعاءات النظام يستخدم هذا النظام تكرار استدعاءات النظام N–grams كخصائص مميزة .لتحسين الأداء، استخدم نموذجان للتعلم الجماعي، وهما التعلم الجماعي القائم على التصويت والتعلم الجماعيXGBoost، لتدريب واختبار البيانات. قُيم النظام المقترح باستخدام مجموعة بيانات اكتشاف الاقتحام(LID–DS) ، مما يدل على أداء كبير مقارنة بالطرائق الحديثة الموجودة. حقق النظام تحسينات كبيرة في دقة الكشف مع زيادة بنسبة 4٪ لنموذج المجموعة القائمة على التصويت وزيادة بنسبة 6٪ لنموذج المجموعة XGBoost بالإضافة إلى ذلك، لوحظ انخفاضٌ في معدل الإيجابيات الكاذبة بنسبة 0.9٪ و 0.8٪ لهذه النماذج. تعمق الباحث[H] في المجال المعقد لاكتشاف الشذوذ غير الخاضع للإشراف داخل بيئات حاويات Docker، نشرت في هذه الدراسة مجموعة من 17 نموذجاً، تشمل 4 نماذج تقليدية لتعلم الآلة (PCA و LOF و Isolation Forest و One Class SVM) و13 نموذجاً مخصصاً للتعلم العميق، تستكشف الدراسة تأثير أنواع البيانات المختلفة وأحجام النوافذ ومستويات الضوضاء على أداء النموذج، مما وفر تحليلاً عميقاً حول النموذج الأكثر وعداً للكشف عن هجوم اليوم الصفري. اقترح الباحث[I] نظاماً للكشف عن الاقتحام في الزمن الحقيقي يراقب ويحلل استدعاءات النظام القائمة لنظام لينكس –Linux kernel لحاوية قيد التشغيل. اعتمد مصنف آلة المتجهات الداعمة أحادية الفئة (OC–SVM) للكشف عن الشذوذ. استخدمت مجموعة بيانات هجوم مخصصة. أظهرت النتائج التجريبية أنه يمكن تحقيق معدل إنذار كاذب FPR منخفض، مع أسوأ حالة (0.12) ، و TPR بقيمة (1) لمعظم الهجمات، عند اعتماد طريقة استخراج الميزات القائمة على التردد واختيار طول التجزئة 30000. اقترح الباحث[J] إطار عمل موزع (DCIDS) يتكون من خمس طبقات لمعالجة التحديات الأمنية في بيئات الحاويات في طبيعتها الديناميكية، مما يعيق فعالية الأساليب التقليدية لكشف التسللات. يعتمد على تقنيات تعلم الآلة لتحليل استدعاءات النظام باستخدام أدوات مفتوحة المصدر. يتميز الإطار بالتوسعية، وقدرته على توليد مجموعات بيانات حديثة، مع تفوق أدائه على الأعمال السابقة في هذا المجال. لا يقوم الإطار بأي استجابة تلقائية عند حدوث إنذارات، ولا يحدد مصدر السلوك الشاذ بدقة، ويستلزم تدريب نموذج خاص لكل تطبيق. قدّم الباحث [K] نظاماً للكشف عن التطفل قائماً على المضيف للبيئات الحاوية باستخدام تقنيات دوكر ولينكس. يهدف نظامهم إلى جمع وتحليل استدعاءات النظام باستخدام خوارزميتي تضمين التأخير الزمني التسلسلي (STIDE) وحقيبة استدعاءات النظام (BoSC). في مرحلة لاحقة، يُدرب

مُصنفاSTIDE وBoSC، بحيث يتراوح حجم نافذة كلٍّ منهما بين 3 و6 استدعاءات نظام. تُظهر النتائج حالة تعلم مستقرة لـ STIDE مع أحجام نوافذ تتراوح بين 3 و4، وتتراوح بين 3 و6 لـ BoSC.

4- طرائق وأدوات البحث:

يتضمن تحليل البيانات الاستكشافي Exploratory Data Analysis (EDA) لتتبعات استدعاءات النظام الموجودة ضمن مجموعة البياناتLID–DS.فحص هياكل ومحتويات التسلسلات، وتقييم جودة البيانات، وفهم توزيعات الفئات المختلفة، واختيار أهم الميزات. تتعلق المرحلة التالية، المعالجة المسبقة، بتنظيف البيانات، استخدم جزءٌ من التتبعات الموجودة ضمن مجموعة البيانات السابقة (40 تتبع) يقصد بالتتبع هنا قائمة استدعاءات النظام الصادرة عن عملية واحدة من بداية تنفيذها حتى نهايتها ، تمثل هذه التتبعات سلوك الطبيعي وسلوك غير طبيعي يتكون من هجمات مختلفة.

استخدمت تقنية One Hot Encoder وهي تقنية تُستخدم لتحويل البيانات الفئوية إلى بيانات رقمية يمكن استخدامها في نماذج التعلم الآلي. هذه التقنية مفيدة بشكل خاص عندما يكون لديك فئات متعددة ولا يوجد ترتيب طبيعي بينها. استخدم نهج النافذة المنزلقة[L]، يشير حجم النافذة المنزلقة إلى عدد استدعاءات النظام التي تؤخذ بالحسبان في كل مرة للتحليل، حيث تُستخدم نافذة بحجم w لمسح البيانات. بهذه الطريقة، سيقُيم تتبع واحد للاستدعاءات إلى تبعات أصغر قد تكون مناسبة لاكتشاف الشذوذ في البيانات. حاول العديد من الباحثين تحديد الحجم المناسب لنافذة الكشف ولكن حجم النافذة يختلف حسب البيئة وحجم البيانات، اختبرت سبعة أحجام مختلفة للنافذة (W=2,4,6,8,10,12,14) لفهم تأثير حجم النافذة على النتائج، باستخدام ثلاث خوارزميات تعلم الآلة، وهي خوارزمية آلة المتجهات الداعمة Support Vector Machine (SVM) والغابة العشوائية Random Forest (RF) وخوارزمية الشبكة العصبية Artificial Neural Network(ANN)، مع المعلمات المبينة في الجدول(1). تم اختيارهذه الخوارزميات لاختبار تأثير اختلاف حجم تسلسل استدعاءات النظام على الأداء، بما يغطي خوارزميات تعتمد على التجميع(RF) ، الفاصل الحدّي(SVM) ، ونماذج تعلّم عميق مرنة(ANN) ، مما يسمح بمقارنة شاملة ومتوازنة للأداء في سياقات مختلف. بالنسبة لمرحلة التدريب، قسمت مجموعة البيانات إلى 70 للتدريب /30 للاختبار. يبين الشكل(1) مخطط نظام الكشف المقترح. شُغل نظام كشف الاقتحام على مستوى المضيف، خارج الحاوية وهذا يحمي نظام كشف الاقتحام من الحاوية المخترقة.



الشكل رقم (1): مخطط نظام الكشف المقترح

استخدمت لغة البرمجة بايثون في عملية تحليل ومعالجة البيانات وتصميم النموذج واختبار دقة الكشف للمصنف المستخدم، المكتبات المستخدمة NumPy، pandas، sklearn ، بيئة عمل البايثون PyCharm.

الجدول رقم (1): معلمات الخوارزميات المستخدمة.

Algorithms	Parameters
RF	(n_estimators=100, random_state=40)
SVM	(C=1.0, kernel='linear', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=– 1, decision_function_shape='ovr', break_ties=False, random_state=None)
ANN	(alpha=1e–05, hidden_layer_sizes=(5,2),

5- مقاييس الأداء:

قُيم أداء كل خوارزمية باستخدام أربعة مقاييس: Precision، Recall، F1–Score، Accuracy. المقياس Precision هي النسبة بين الاكتشافات المتوقعة بشكل صحيح وجميع الاكتشافات التي تحدث، حيث تمثل القيمة الأعلى معدلًا إيجابياً كاذباً أقل. ومن ناحية أخرى، يمثل معدل الاستدعاء Recall نسبة الاكتشافات المحددة من بين جميع الاكتشافات المحتملة. تجمع F1–score بين قيم Precision و Recall في درجة واحدة للإشارة إلى الجودة الشاملة للنموذج.

1. دقة الكشف Accuracy: يتعلق بنسبة جميع البيانات المصنفة بشكل صحيح بواسطة النموذج على إجمالي البيانات.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \qquad (1)$$

2. الدقة Precision: هي نسبة البيانات المصنفة بشكل صحيح على جميع البيانات المتوقعة كهجمات.

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (2)$$

3. الاستدعاءrecall : يمثل نسبة الاكتشافات المحددة من بين جميع الاكتشافات المحتملة.

$$\text{Recall} = \frac{TP}{TP+FB} \qquad (3)$$

4. تجمع F1–score بين قيم الدقة المثالية Precision والاستدعاء في درجة واحدة للإشارة إلى الجودة الشاملة للنموذج.

$$\text{F1–Score} = \frac{2* Precision* Recall}{Precision+ Recall} \qquad (4)$$

حيث أن: TP: True Positive يعني أن النموذج تنبأ بشكل صحيح بوجود الحالة الإيجابية(الحالة الإيجابية تُمثل الفئة التي يُركز عليها النموذج في حالتنا اكتشاف الهجوم)، TN: True Negative يعني أن النموذج تنبأ بشكل صحيح بعدم وجود الحالة الإيجابية، FP: False Positive يعني أن النموذج تنبأ بشكل خاطئ بوجود الحالة الإيجابية، False FN: Negative يعني أن النموذج فشل في التعرف على الحالة الإيجابية وتنبأ بشكل خاطئ بأنها سلبية.

6- سيناريو التجارب:

تُعد التجارب المصممة بعناية أداة أساسية لفهم الظواهر المعقدة وتقييم الأداء في بيئات متنوعة. يهدف هذا البحث إلى دراسة العلاقة بين أحجام تسلسلات استدعاءات النظام المختلفة باستخدام نهج النافذة المنزلقة وكفاءة كشف الاقتحام في أنظمة الحوسبة الحديثة باستخدام مجموعة من الخوارزميات لتحقيق هذا الهدف، صُمم سيناريو تجارب مكون من مرحلتين متكاملتين. في المرحلة الأولى، قَيم أداء نظام كشف الاقتحام باستخدام جميع استدعاءات النظام الأصلية الموجودة في مجموعة البيانات LID–DS ، واستبعاد استدعاءات النظام ذو التهديد المنخفض (هناك مجموعة من الاستدعاءات التي تعتبر ذو تهديد منخفض ويمكن الاستغناء عنها صممن الى اربع مستويات من التهديد[M]، يشير الجدول(2) الى استدعاءات النظام التي

تم تصنيفها على أنها منخفضة التهديد الموجودة، وبالتالي إهمال استدعاءات النظام ذو التهديد المنخفض سيؤدي إلى أن التدريب والقرار الذي تتخذه المصنفات سيكون أسرع، وبالتالي سيكشف الهجوم بشكل أسرع) وفق أحجام نافذة مختلفة باستخدام خوارزمية الغابة العشوائية، آلة المتجهات الداعمة والشبكة العصبية الاصطناعية. في المرحلة الثانية من التجارب سيتم التركيز على تقييم الأداء الأساسي لخوارزميات التصنيف ذو الفئة واحدة المصممة لكشف الشذوذ وهما خوارزمية آلة المتجهات الداعمة ذات الفئة الواحدة OC–SVM [N] وخوارزمية عامل الشذوذ المحلي– Local Outlier Factor) [O] (LOF، لمعالجة مشكلة التحيز نحو الفئة المهيمنة (السلوك الطبيعي) في البيانات غير المتوازنة، مما يقلل من قدرة النموذج على اكتشاف الحالات الشاذة (الفئة الأقل). ليشمل التتبع الكامل من خلال زيادة حجم النافذة تدريجيًا بنسبة 5% حتى الوصول إلى 100% من حجم التتبع، بهدف استكشاف العلاقة بين أحجام التتبع الكبيرة وأداء كشف الهجمات في البيانات غير المتوازنة، سيتم هنا أخذ مجموعة فرعية غير متوازنة من مجموعة البيانات LID–DS. في سياق البيانات غير المتوازنة، تم اختيار 189,251 استدعاء نظام يمثل سلوكاً طبيعياً (99.25%) و1,435 استدعاء نظام يمثل سلوكاً شاذاً (0.75%)، في الواقع، السلوك الشاذ (Anomalous Behavior) يحدث عادةً بنسب صغيرة جداً مقارنةً بالسلوك الطبيعي. في بيانات الشبكات، غالباً ما تكون الحركة الطبيعية (Normal Traffic) تشكل النسبة الأكبر (قد تصل إلى 99% أو أكثر)، بينما تمثل الأحداث الشاذة مثل الهجمات أو الاختراقات نسبة ضئيلة (1% أو أقل).

الجدول رقم (2): استدعاءات النظام المصنفة على أنها غير ضارة [M].

1	oldstat, oldfstat, access, sync, pipe, ustat, oldstat, readlink, readdir, statfs, fstatfs, stat, getpmsg, lstat, fstat, oldname, bdflush, sysfs, getdents, fdatasync.
2	getpid, getppid, getuid, getgid, geteuid, getegid, acct, getpgrp, sgetmask, getrlimit, getrusage, getgroups, getpriority, sched get priority min, getpid, getsid, getcwdm getresgid, getresuid
3	get kernel syms, create module, query module
4	times, time, gettimeofday, getitimer
5	sysinfo, uname
6	idle
7	break, ftime, prof, ulimit, gtty, lock, profil

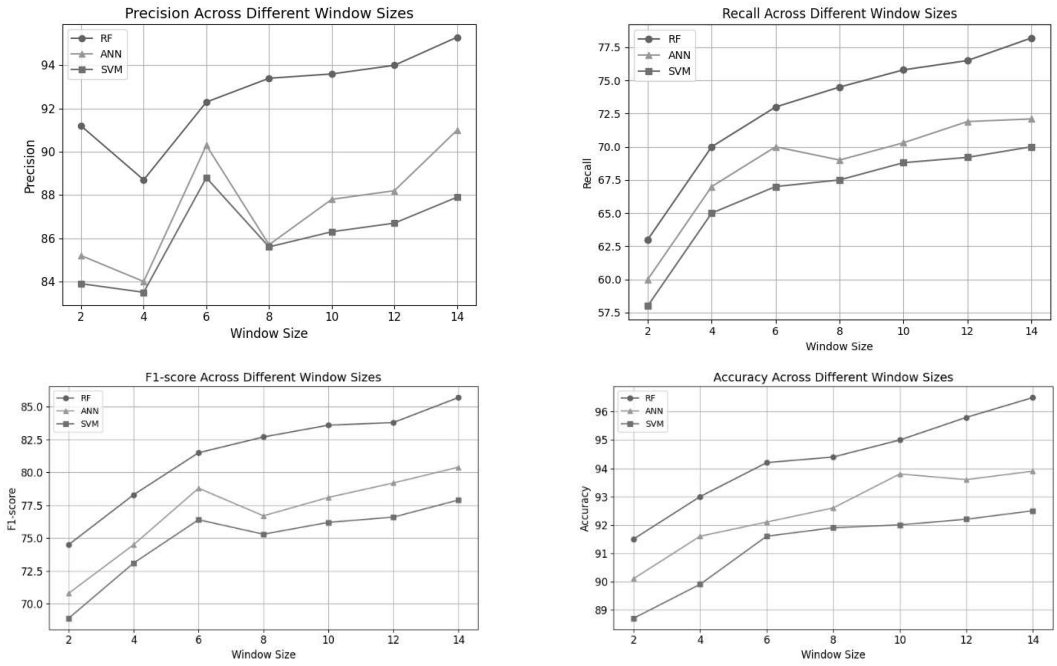
ترتكز العديد من الدراسات السابقة على استخدام التتبع كاملاً أثناء عملية اكتشاف الاقتحام، والتي تعتمد غالباً على الاكتشاف من خلال الكشف في الزمن غير الحقيقي، وهذا غير فعال في الزمن الحقيقي لأن عملية التحليل لن تتم إلا بعد إنتهاء تنفيذ الهجوم وبالتالي سيؤثر على عمل نظام كشف الاقتحام كما هو متوقع في الزمن الحقيقي، وهنا تكمن أهمية نهج النافذة المنزلقة (الاكتشاف في الزمن الحقيقي). تشير الدراسات السابقة إلى اختلافات في الحجم الأمثل للنافذة بناءً على البيئة. فعلى سبيل المثال، أشارت دراسات [Q][P] إلى أن أحجام النوافذ من 6 إلى 7 هي الأمثل لكشف الهجمات في أنظمة Unix، بينما أوصت دراسات [S][R] بأحجام تتراوح بين 10 و12 للأجهزة الافتراضية. في هذا البحث، أظهرت التجارب أن أحجام النوافذ من 12 إلى 14 كانت الأكثر كفاءة ، مما يعكس تأثير خصائص البيانات والبيئة على اختيار الحجم الأمثل، ونظراً لأن مجموعة البيانات تتألف من تتبعات بأحجام مختلفة، فإن حجم النافذة الثابت مطلوب لكل من مرحلتي التدريب والاختبار. يحدد حجم النافذة هذا بناءً على أصغر حجم تتبع متاح في مجموعة البيانات. على سبيل المثال، عند تحليل استدعاءات النظام التي تمثل سلوك ضار، فإن 100% من التتبعات تتوافق مع حجم 1347.

من خلال هذا التصميم التجريبي ثنائي المراحل، يسعى البحث إلى تقديم رؤى شاملة حول كيفية تحسين أنظمة كشف الاقتحام في الزمن الحقيقي بالنسبة إلى الدقة في التصنيف.

7- النتائج والمناقشة:

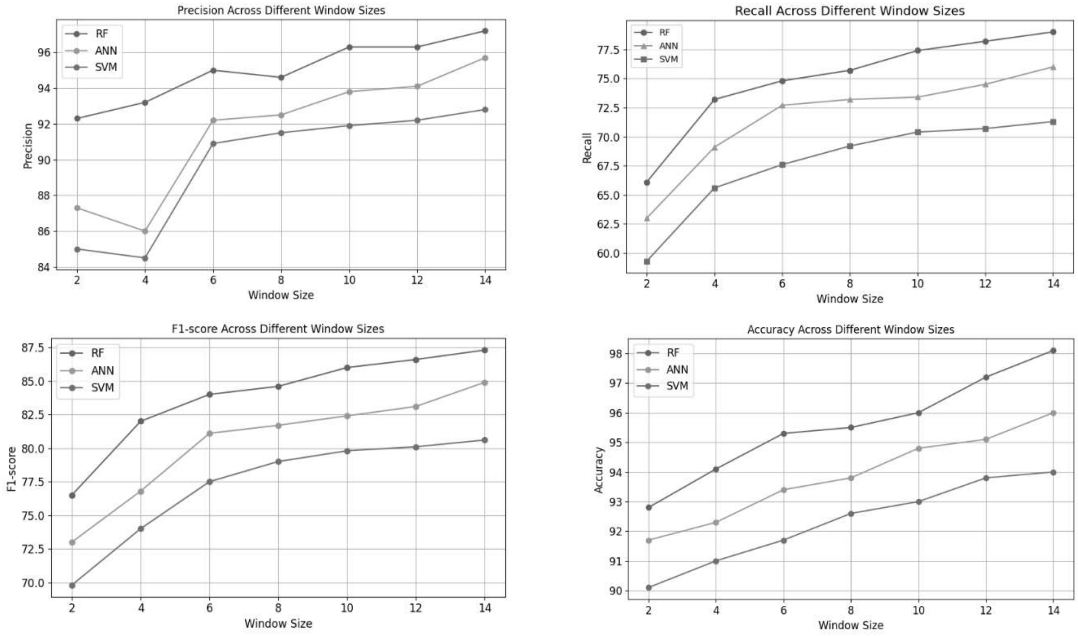
في المرحلة الأولى من التجارب قُيم أداء نظام كشف الاقتحام باستخدام جميع استدعاءات النظام الأصلية الموجودة في مجموعة البيانات LID–DS نطلق عليها Raw، واستخدام جميع استدعاءات النظام بعد إزالة الاستدعاءات ذات التهديد المنخفض نطلق عليها Filter. يعرض الشكل(2) النتائج التي تم الحصول عليها من خلال متوسط خمسة عمليات تنفيذ. تظهر النتائج تفوق خوارزمية الغابة العشوائيةRF بالنسبة إلى جميع المقاييس، حيث حققت أعلى دقة accuracy (96.5%) و التذكر Recall (78%) والدقة الإجمالية (95.2%) عند حجم نافذة 14، مع زيادة مستمرة بزيادة حجم النافذة. تظهر الخوارزمية ANN أداءً متوسطاً مع تقلبات ملحوظة في الدقة الاجمالية precision (84%–91%) وRecall (65%–72.5%)، بينما يحقق SVM أداءً مستقراً نسبياً لكنه الأدنى، خاصة في التذكر Recall (60%–70%). يعود الانخفاض الملحوظ في الأداء عبر المقاييس (Precision, Recall, F1, Accuracy) عند أحجام النوافذ الصغيرة (w=2 6) إلى قلة السياق الزمني (Contextual Information) المتاح لتحليل سلوك العملية. لا تمثل التسلسلات القصيرة أنماط السلوك الطبيعي أو الهجومي بشكل كافٍ، مما يسبب صعوبة في التمييز بين السلوك الطبيعي والشاذ، وبالتالي ارتفاع معدلات الإيجابيات الخاطئة.(False Positives) وهذا كان واضحاً في جميع الخوارزميات، لكنه كان أكثر تأثيراً لدى SVMو ANN مقارنة بـRF، يعود السبب في ذلك الى اعتماد هذه الخوارزميات على تمييز حدود دقيقة بين الفئات والتي تتطلب تمثيلاً دقيقاً للسلوك. تحتوي النوافذ الصغيرة على عدد محدود من استدعاءات النظام، مما يجعلها غير كافية لتمثيل الأنماط المعقدة للهجمات أو السلوك الطبيعي في بيئة الحاويات. على سبيل المثال، قد يتشابه تسلسل قصير مكون من استدعاءات طبيعية مع جزء من هجوم، مما يؤدي إلى زيادة الإيجابيات الخاطئة (FP) وانخفاض Precision مثل انخفاض Precisionللمصنف ANN إلى 86% وانخفاض Precision للمصنف SVM الى84.3%عند w=4 ، تكون الخوارزميات في النوافذ الصغيرة (خاصة ANN وSVM) أكثر عرضة للضوضاء بسبب افتقارها إلى بيانات كافية للتعلم. على سبيل المثال، ANN، التي تستخدم هيكلية (5، 2) للطبقات، قد تعاني من الإفراط في التكيف (Overfitting) أو ضعف التعلّم (Underfitting)عند w=2 وw=4، مما يؤدي إلى تقلبات في (58%–63%) Recall كما هو موضح في الشكل(2)، تواجه خوارزمية SVM باستخدام النواة الخطية صعوبة في التعامل مع الأنماط غير الخطية في التسلسلات القصيرة، مما يقلل من قدرتها على الفصل بين الفئتين (طبيعي مقابل شاذ)، وهو ما يفسر انخفاض قيمة Recall عند w=2 و w=4، الغابة العشوائية أقل عرضة للإفراط في التكيف بفضل التجميع والعشوائية، مما يجعلها مناسبة لمجموعات بيانات صغيرة أو متوسطة الحجم مثل LID–DS (40 تتبعًا في الدراسة) .

تبدأ النوافذ عند أحجام (14 w=8) بالتقاط سياق زمني أوسع، يتضمن تسلسل أطول لاستدعاءات النظام، مما يساعد النماذج في بناء رؤية أوضح عن النمط السلوكي للعملية. يُشير التحسن في مقاييس الأداء (خصوصًا Recall وF1–score) إلى قدرة النماذج على التقاط المزيد من الحالات الشاذة مما يقلل من الإيجابيات الخاطئة ويزيد من الإيجابيات الحقيقية . نلاحظ تفوق الغابة العشوائية بسبب قدرتها على التعامل مع الأنماط غير الخطية والتفاعلات المعقدة بين السمات دون الإفراط في التكيف و هذا يفسر الارتفاع المستمر في قيم مقاييس الأداء بالنسبة للغابة العشوائية. يشير الحصول على أفضل أداء لجميع المقاييس (Accuracy, Precision, Recall, F1–score) عند حجم نافذة 14 إلى أن قيمة 14 تمثل الحجم الامثلي للنافذة التي تحقق التوازن بين الغنى السياقي المطلوب لتحليل السلوك وبين الكفاءة الزمنية المطلوبة في بيئات الزمن الحقيقي، يكفي هذا الحجم لالتقاط النمط الكامل للهجوم دون إدخال ضجيج زائد أو تحميل زائد على النظام.



الشكل رقم (2): أداء الخوارزميات وفق مجموعة البيانات الأصلية مع أحجام نوافذ مختلفة

أظهرت النتائج في الشكل(3) تحسن في قيم مقاييس الأداء بالنسبة لجميع الخوارزميات مع استمرار تفوق خوارزمية الغابة العشوائية حيث سجلت أعلى(87.2%) F1–score ، الدقة الاجمالية (97.2%)، والتذكر (79%) والدقة (98.1%) عند حجم نافذة 14،. يُظهر ANN أداءً متوسطاً مع تقلبات، حيث يتراوح F1–score حول 85%، الدقة الاجمالية بين 86%–94%، والتذكر بين 65%–72.5%. يحقق SVM أداءً مستقراً لكنه الأدنى، بـ F1–score حول 80.5%، الدقة الاجمالية بين 84%–93%، والتذكر بين 60%–71%. بشكل عام، تتحسن الخوارزميات مع زيادة حجم النافذة، لكن RF يبرز بأداء متميز وثابت عبر جميع المقاييس. حقق حجم النافذة w=14 أفضل النتائج، مما يشير إلى أن هذا الحجم يوفر السياق الأمثل لتحليل تسلسلات استدعاءات النظام، مما يعزز دقة الكشف عن الهجمات. قدمت الحالتان كلتاهما قيم تذكر (Recall)ودقة إجمالية (Precision) عالية باستخدام RF مع أطوال تسلسل مناسبة، ارتفاع قيم Recall يعني تقليل عدد الهجمات غير المكتشفة، و ارتفاع قيم Precision يعني تقليل عدد الإنذارات الكاذبة.



الشكل رقم (3): أداء الخوارزميات بعد إهمال استدعاءات النظام ذو التهديد المنخفض مع أحجام نوافذ مختلفة

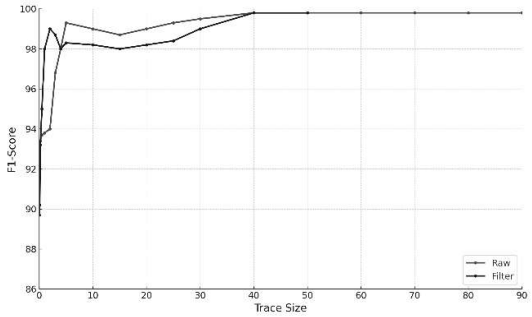
يعرض الشكل (4) والشكل(5) نتائج خوارزمية LOF ونتائج خوارزمية آلة المتجهات الداعمة OC–SVM، بالنسبة الى الشكل(4)، في المرحلة الأولى (من 0% إلى 10%) نجد أداء منخفضاً نسبياً في البداية بسبب محدودية السياق الزمني المتوفر لتحليل السلوك، تذبذب واضح في أداء "Filter" مقارنة بـ"Raw" ، مما يشير إلى أن التصفية قد تؤدي إلى فقدان إشارات مهمة تحتاجها خوارزمية LOF في هذه المرحلة المبكرة. النوافذ الصغيرة 2 الى 6 فعالة للكشف السريع، لكنها تُعاني من زيادة الإيجابيات الخاطئة نظراً لانخفاض السياق. في المرحلة الثانية (من 10% إلى 30%) يتحسن F1–Score بشكل ملحوظ، خصوصاً بالنسبة إلى بيانات"Raw" ، مما يدل على أن خوارزمية LOF تحتاج إلى حجم تتبع متوسط لتشكيل تصور أوضح حول التوزيع المحلي للنقاط.، هذا يتوافق مع ما أشير إليه في النتائج السابقة للدراسات بأن النافذة المتوسطة 8 الى 14 تعزز دقة الكشف في السيناريوهات المعقدة. من نقطة 40% فصاعداً، نلاحظ تشبع في الأداء، حيث تصل F1–Scoreإلى قرابة 99.5% وتبقى ثابتة تقريباً هذا يعكس أن التحليل الكامل للتتبع ليس ضرورياً للوصول لأداء عالٍ. يشير إلى أن نظام كشف الاقتحام يمكنه العمل بكفاءة في الزمن الحقيقي دون الحاجة إلى انتظار اكتمال الهجوم.

بالنسبة إلى الشكل(5)، حجم التتبع (0 20) ، في حالة البيانات الكاملة: تبدأ عند99%، وتنخفض قليلاً إلى 97% عند حجم تتبع 10، وتتعافى إلى 98.4% عند حجم تتبع 20. في حالة البيانات المفلترة: يبدأ عند 92.4%، و يرتفع الى 99% وينخفض بشكل حاد إلى 95% عند حجم تتبع 5، ويتعافى إلى 98% عند حجم تتبع 20. عند أحجام التتبع الصغيرة (التي تعادل أحجام النوافذ الصغيرة 2–6)، تبالغ الخوارزمية بداية في تقدير الأداء (92.4%) بالنسبة للبيانات المفلترة، ربما بسبب محدودية السياق التي تؤدي إلى تقليل الإيجابيات الخاطئة ولكنها تفوت الأنماط طويلة الأمد، مما يؤدي إلى انخفاض حاد. هذا يتماشى مع أن النوافذ الصغيرة يمكن أن تزيد من الإيجابيات الخاطئة بسبب محدودية السياق. بالنسبة إلى مجموعة البيات الكاملة نجد أن OC–SVM يتفيد من المزيد من البيانات لنمذجة السلوك الطبيعي بشكل أفضل.

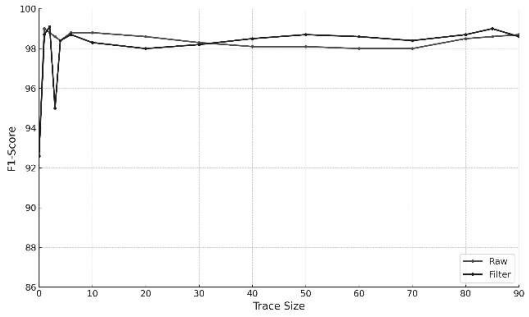
حجم التتبع (20 40)، يتقارب كلا المخططين حول 98% عند حجم تتبع 40، مع تفوق طفيف للبيانات المفلترة حيث يستمر في الارتفاع. مع زيادة حجم التتبع (الذي يتوافق مع أحجام النوافذ المتوسطة 8–14)، يتم الاستفادة من سياق أوسع. نلاحظ ارتفاع قيم البيانات المفلترة "Filter" من الانخفاض الأولي، مما يشير إلى أن التصفية تقلل من الضوضاء أو

الإيجابيات الخاطئة مع توفر المزيد من البيانات، وهو ما يتماشى مع ملاحظة أن النوافذ المتوسطة والكبيرة تعزز كشف الهجمات المعقدة من خلال توفير سياق أكبر. بينما يبقى مخطط البيانات الكاملة مستقراً، مما يظهر أن OC–SVM يمكن أن تنمذج السلوك الطبيعي بفعالية مع بيانات كافية.

حجم التتبع (40 90)، يستقر الخطان كلاهما حول 98% بعد حجم تتبع 40. حجم تتبع 40% يتوافق مع ~538 استدعاء نظام (40% من 1347)، مما يشير إلى أن حجم نافذة يبلغ حوالي 12–14 كافٍ للكشف الفعال، وهو ما يتماشى مع النتيجة السابقة بأن 12–14. التحسن الطفيف في مخطط الفلترة "Filter" عند أحجام التتبع الكبيرة يشير إلى أن التصفية تصبح أكثر فعالية مع المزيد من البيانات، ربما من خلال تمييز الحالات الشاذة بشكل أفضل في مجموعة بيانات غير متوازنة. الانخفاض في مخطط البيانات الكاملة "Raw" يشير إلى أن OC–SVM بدون تصفية قد يفرط في التكيف أو يعاني من الضوضاء مع زيادة حجم التتبع أكثر من حجم النافذة المثالي.



الشكل رقم (4): توسع النافذة لتشمل 100% من حجم التتبع – LOF



الشكل رقم (5): توسع النافذة لتشمل 100% من حجم التتبع – OC–SVM

يبين الجدول(3)، مقارنة لدراستنا مع مجموعة من الدراسات، على الرغم من تحقيق الدراسات [U][T] نتائج اعلى من دراستنا الحالية من حيث Recall و F1–Score إلا أن الدراسات السابقة تعتمد عبى مجموعات بيانات غير موثقة تم جمعها من قبل الباحث ، بالاضافة الى اعتماد الدراسة [T] الى حجم تسلسل ثابت(نافذة ثابتة الحجم)، دون سبب توضيح اختيار هذا الحجم او دراسة تأثير احجام مختلفة، الدراسة [I] بالإضافة الى عدم موثوقية مجموعة البيانات فأن الهجمات المستخدمة قليلية وتناقش حالة قواعد البيانات على عكس مجموعة البيانات المستخدمة في بحثنا و التي تشمل هجمات متنوعة.

الجدول رقم (3): مقارنة بين الدراسة الحالية مع دراسات سابقة.

الدراسة	حجم النافذة الأمثل	تصفية بيانات	النموذج	مجموعة البيانات	Precision	Recall	F1–score
الدراسة الحالية	14	نعم	RF	LID–DS	95.2%	78.0%	86%
[T]	6	لا	STIDE	غير موثقة	68.2%	99.2%	80%
			BoSC		75.5%	99.1%	85%
[U]	تسلسل بطول 256	نعم	U–SCAD Syscall2vec+) تصنيف)	غير موثقة	91.4%	97.8%	94.4%
[V]	30	لا	KNN, RF	غير موثقة	0.996	0.991	0.994
[I]	تسلسل بطول 30,000	لا	OC–SVM, CouchDB	غير موثقة	~33.5%	100%	50.1%
			OC–SVM, MongoDB		~83.3%	100%	90.9%

8- الاستنتاجات:

في هذا البحث، تم التركيز على دراسة تأثير أحجام مختلفة لتسلسلات استدعاءات النظام اعتماداً على نهج النافذة المنزلقة لكشف الهجمات القائمة على ميزة التسلسل. في التجارب، استخدمت مجموعة البيانات LID–DS التي تحتوي على مجموعة متنوعة من الهجمات عبر استدعاءات النظام، أداء الخوارزميات: تفوقت خوارزمية الغابة العشوائية في دقة الكشف مقارنة بـ SVM وANN، حيث حققت أعلى دقة accuracy (96.5%) و التذكر Recall (78%) والدقة الإجمالية (95.2%) عند حجم نافذة 14 بالنسبة إلى البيانات الأصلية و سجلت أعلى(87.2%) F1–score ، الدقة الاجمالية (97.2%)، والتذكر (79 ,%) و الدقة (98.1%) بالنسبة إلى البيانات المفلترة مما يجعلها خياراً مفضلاً لتحليل تسلسلات استدعاءات النظام، تأثير حجم النافذة: الحجم الأمثل للنافذة (w=14) يحقق توازناً بين دقة الكشف والكفاءة الحسابية، لكن النوافذ الأصغر (10–14) يمكن أن تكون فعالة مع فروق أداء طفيفة عند استخدام أطوال قصيرة (مثل 10 أو 20 استدعاء)، تكون النماذج غير قادرة على التقاط نمط الهجمات الكامل. مع زيادة الطول إلى 40 أو 50، ترتفع الدقة بشكل كبير لأن النموذج يلتقط السياق الكامل للسلوك الخبيث.بعد نقطة معينة، تزداد تكلفة الحسابات دون تحسن جوهري في الأداء، هذا يعكس أهمية التوازن بين الأداء والكفاءة الحسابية في بيئة الحاويات التي تتطلب كشفاً سريعاً وخفيفاً للموارد.

9- التوصيات:

- ✧ اختيار المصنفات: يُوصى باستخدام خوارزمية الغابة العشوائية كخيار أساسي لتطبيقات كشف الاقتحام القائمة على تسلسل استدعاءات النظام نظراً لأدائها المتفوق. مع استخدامات خوارزميات تعتمد على التعلم العميق مستقبلا
- ✧ تحسين حجم النافذة: ينبغي إجراء تجارب مكثفة لتحديد حجم النافذة المناسب بناءً على نوع الهجمات وخصائص البيانات، مع التركيز على النوافذ المتوسطة للكشف في الوقت الفعلي.
- ✧ تصفية البيانات: يُوصى بتطبيق تصفية مسبقة لاستبعاد استدعاءات النظام غير الضارة لتقليل الضوضاء وتحسين كفاءة النظام.
- ✧ تكييف النظام: يجب تصميم أنظمة الكشف بحيث تكون مرنة للتكيف مع مجموعات بيانات متنوعة، مع مراعاة متطلبات الموارد في البيئات السحابية.

10-الأعمال المستقبلية:

يُنصح بدراسة تأثير تقنيات متقدمة مثل التعلم العميق أو تحليل السياق الديناميكي على أداء أنظمة الكشف مع النوافذ المنزلقة، مع التركيز على تحسين الكشف عن الهجمات المعقدة.

المراجـــع

[A]. Joraviya, N., Gohil, B. N., & Rao, U. P. (2024). Ab-HIDS: An anomaly-based host intrusion detection system using frequency of N-gram system call features and ensemble learning for containerized environment. Concurrency and Computation: Practice and Experience, 36(23), e8249.

[B]. Zhang, L., Cushing, R., de Laat, C., & Grosso, P. (2021). A real-time intrusion detection system based on OC-SVM, In 2021 IEEE 24th International Conference on Computational Science and Engineering (CSE) (pp. 138-145).

[C]. Grimmer, M., Kaelble, T., Nirsberger, F., Schulze, E., Rucks, T., Hoffmann, J., & Rahm, E. (2022, September). Dataset Report: LID-DS 2021. In International Conference on Critical Information Infrastructures Security (pp. 63-73). Cham: Springer Nature Switzerland.

[D]. Shamim, N., Asim, M., Baker, T., & Awad, A. I. (2023). Efficient Approach for Anomaly Detection in IoT Using System Calls. Sensors, 23(2), 652. https://doi.org/10.3390/s23020652

[E]. Birihanu, E., & Lendák, I. (2025). Explainable correlation-based anomaly detection for Industrial Control Systems. Frontiers in Artificial Intelligence, 7, 1508821.

[F]. Vajda, D. L., Do, T. V., Bérczes, T., & Farkas, K. (2024). Machine learning-based real-time anomaly detection using data pre-processing in the telemetry of server farms. Scientific Reports, 14(1), 23288.

[G]. El Khairi, A., Caselli, M., Knierim, C., Peter, A., & Continella, A. (2022, November). Contextualizing system calls in containers for anomaly-based intrusion detection. In Proceedings of the 2022 on Cloud Computing Security Workshop (pp. 9-21).

[H]. Rossotti, A. (2022). Anomaly detection framework and deep learning techniques for zero-day attack in container based environment

[I]. Zhang, L., Cushing, R., de Laat, C., & Grosso, P. (2021, October). A real-time intrusion detection system based on OC-SVM for containerized applications. In 2021 IEEE 24th international conference on computational science and engineering (CSE) (pp. 138-145). IEEE.

[K]. Flora, J., & Antunes, N. (2019, September). Studying the applicability of intrusion detection to multi-tenant container environments. In 2019 15th European Dependable Computing Conference (EDCC) (pp. 133-136). IEEE.

[L]. Carmona-Cabezas, R., Gómez-Gómez, J., Gutiérrez de Ravé, E., & Jiménez-Hornero, F. J. (2019). A sliding window-based algorithm for faster transformation of time series into complex networks. Chaos: An Interdisciplinary Journal of Nonlinear Science, 29(10).

[M]. Bernaschi, M., Gabrielli, E., & Mancini, L. V. (2002). REMUS: A security-enhanced operating system. ACM Transactions on Information and System Security (TISSEC), 5(1), 36-61.

[N]. Mahfouz, A. M., Abuhussein, A., Venugopal, D., & Shiva, S. G. (2021). Network intrusion detection model using one-class support vector machine. In Advances in Machine Learning and Computational Intelligence: Proceedings of ICMLCI 2019 (pp. 79-86). Springer Singapore

[O]. Alghushairy, O., Alsini, R., Soule, T., & Ma, X. (2020). A Review of Local Outlier Factor Algorithms for Outlier Detection in Big Data Streams. Big Data Cogn. Compute. 2021, 5,

[P]. Srinivasan, S., Kumar, A., Mahajan, M., Sitaram, D., & Gupta, S. (2019). Probabilistic realtime intrusion detection system for docker containers. In Security in Computing and Communications: 6th International Symposium, SSCC 2018, Bangalore, India, September, Revised Selected Papers 6 (pp. 336-347). Springer Singapore.

[Q]. Byrnes, J., Hoang, T., Mehta, N. N., & Cheng, Y. (2020, October). A modern implementation of system call sequence-based host-based intrusion detection systems. In 2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA) (pp. 218-225). IEEE

[R]. Abed, A. S., Clancy, T. C., & Levy, D. S. (2015, December). Applying bag of system calls for anomalous behavior detection of applications in linux containers. In 2015 IEEE globecom workshops (GC Wkshps) (pp. 1-5). IEEE.

[S]. Liu, M., Xue, Z., Xu, X., Zhong, C., & Chen, J. (2018). Host-based intrusion detection system with system calls: Review and future trends. ACM computing surveys (CSUR), 51(5), 1-36

[T]. Flora, J. E. F. (2019). *Container-level Intrusion detection for multi-tenant environments* (Master's thesis, Universidade de Coimbra (Portugal))

[U]. Ye, J., Yan, M., Wu, S., Tan, J., & Wu, J. (2025). U-SCAD: An Unsupervised Method of System Call-Driven Anomaly Detection for Containerized Edge Clouds. Future Internet, 17(5), 218

[V]. Cavalcanti, M., Inacio, P., & Freire, M. (2021, August). Performance evaluation of container-level anomaly-based intrusion detection systems for multi-tenant applications using machine learning algorithms. In Proceedings of the 16th International Conference on Availability, Reliability and Security (pp. 1-9