**الجامعة السورية الخاصة**

**كلية هندسة الذكاء الاصطناعي**

**قسم أمن النظم و الشبكات الحاسوبية**

**اكتشاف عناوين URLs الضارة باستخدام خوارزميات التصنيف في التعلم الآلي والتعلم العميق**

تقرير مشروع (تخرج2) - قدم إستكمالا لمتطلبات الحصول على درجة البكالوريوس

في هندسة الذكاء الاصطناعي – هندسة أمن النظم والشبكات الحاسوبية

**اعداد:**

**رند حموي**

**سدرة ستوت**

**اشراف:**

**د.وسيم جنيدي**

**2026-2025**

**ملخص**

Contents

**No table of contents entries found.**

**1.1** مقدمة**:**

في ظل التوسع الرقمي المتسارع, أصبحت حماية عناوين URLs)) بمثابة العصب الأساسي لضمان أمن البيانات والخدمات المنقولة عبر شبكة الأنترنت.

حيث أصبح تصفح الأنترنت نشاطاً أساسياً للأفراد والشركات والحكومات على حد سواء, ومع تزايد حجم البيانات المنقولة عبر الأنترنت وحساسيتها, أصبح ضمان أمن اتصالات الأنترنت أمراً بالغ الأهمية.[1]

تهدف هجمات التصيد الاحتيالي، التي تُنفذ عادةً عبر عناوين URL، إلى خداع المستخدمين ودفعهم إلى الكشف عن معلومات حساسة، مما يؤدي إلى خسائر مالية أو سرقة هوية أو تعريض بيانات حساسة للخطر. ولا تزال تُشكل تهديدًا كبيرًا للأفراد والمؤسسات، مما يتطلب حماية فعالة لعناوين URLs)) 2]].

تعمل الروابط الضارة كحجر الزاوية في الهجمات الإلكترونية الحديثة، وخاصة هجمات الهندسة الاجتماعية. تكمن أهميتها المركزية في كونها البوابة الرئيسية للهجمات الإلكترونية فبدلاً من استهداف الثغرات التقنية، تركز هذه الهجمات على العنصر البشري الذي يعد أضعف حلقات الأمن، مما يمكنها من تجاوز حتى أقوى الدفاعات التقنية مثل جدران الحماية والتشفير.

أكثر أنواع هذه الهجمات شيوعاً هو التصيد الاحتيالي (Phishing)، والذي يعتمد بالدرجة الأولى على إقناع المستخدم بالنقر على رابط ضار يتم تصميم هذه الروابط ضمن رسائل أو مواقع مزيفة تظهر وكأنها صادرة عن مصادر موثوقة مثل البنوك أو جهات اتصال عمل معروفة، بهدف خداع الضحية لكشف معلومات حساسة أو تحويل أموال.

تكمن خطورة هذه الروابط في تطورها المستمر. فقد ساهمت تقنيات الذكاء الاصطناعي في زيادة تعقيدها، حيث أصبح من الممكن إنشاء مواقع ويب احتيالية مقلدة بدرجة عالية من الدقة، ورسائل تصيد أكثر تخصيصاً وإقناعاً. هذا التطور يجعل اكتشافها أكثر صعوبة على المستخدم العادي. الخطر الأكبر يتجلى في العواقب الملموسة. فالهجمات التي تستخدم هذه الروابط، مثل احتيال اختراق البريد الإلكتروني للشركات يمكن أن تؤدي إلى خسائر مالية فادحة للمؤسسات والأفراد على حد سواء. غالباً ما تنجح هذه الهجمات من خلال استغلال الثقة داخل الشركة حيث ينخدع الموظفون بطلبات تبدو وكأنها صادرة عن مدرائهم.[3]

**1.2المشكلة العلمية:**

يستغل المحتالون سرقة المعلومات الشخصية مثل كلمات المرور والحسابات المصرفية وما إلى ذلك. ولذلك أصبح اكتشاف عناوين URLs الضارة مصدر قلق بالغ الأهمية بسبب العدد المتزايد من هجمات التصيد الإحتيالي التي تسعى إلى خداع المستخدمين لكسب ثقتهم من خلال انتحال شخصية أشخاص أو شركة أو خدمة, لتحقيق أهدافهم.[4]

لذا, ثمة الحاجة لمنع استنزاف البيانات من خلال تكرار مواقع الويب الموثوقة, وزيادة الوعي العام بهذه الممارسات. ولهذا الغرض, استخدمت العديد من نماذج التعلم الآلي والتعلم العميق للكشف عن هجمات التصيد الإحتيالي ومنعها, ولكن نظراً للطبيعة المتطورة باستمرار لهذه الهجمات, تفشل العديد من الأنظمة بتقديم نتائج دقيقة, ومن أبرز المناهج التي تم استخدامها هو نهج القائمة السوداء التي تحمي متصفحات google من هجمات التصيد الاحتيالي عن طريق تحديث قائمة المواقع المدرجة فيها.[5]

**1.3هدف البحث:**

في المرجع [4] تم العمل على المشكلة العلمية وتم الوصول الى النتائج التالية تم تحليل مشكلة الأمن السيبراني المتمثلة في الكشف عن عناوين URL الاحتيالية من وجهات نظر مختلفة، باستخدام التعلم الآلي بنسختيه التقليدية والكمية. حيث أن الهدف الرئيسي هو تحقيق أقل عدد من النتائج الإيجابية الكاذبة، ولكن دون زيادة كبيرة في عدد النتائج السلبية الكاذب.

وكانت أحد أهداف البحثية المتروكة هي كيفية دمج الخوارزميات في تحليل مشاكل الأمن السيبراني المختلفة، وبالتالي دمج إمكانيات جديدة للكشف المبكر عن الإجراءات الاحتيالية.

في المرجع [5] تم العمل على المشكلة العلمية وتم الوصول الى النتائج التالية طُوّر في هذه الدراسة نموذج للكشف عن التصيد الاحتيالي والمواقع الشرعية. لهذا الغرض، صممنا موقعًا إلكترونيًا يُمكّن المستخدمين من إدخال عنوان URL للتمييز بين عناوين URL المزيفة والشرعية.

وكانت أحد أهداف البحثية المتروكة هي كشف التصيد الاحتيالي القائم على المحتوى، حيث سيتم فحص صفحة الويب بالكامل للكشف عن روابط التصيد الاحتيالي الموجودة عليها وتحديدها، والتي قد تُعيد توجيه العملاء بعيدًا عن الموقع إلى صفحة تصيد احتيالي أخرى.

**الدراسة المراجعية: 1.4**

**"A hybrid deep learning technique for spoofing website URL detection in real‑time applications"[6]**

قام كل من Bridget C. Ujah‑Ogbuagu, Oluwatobi Noah Akande and Emeka Ogbuju عام 2024 باستخدام نموذج هجين يجمع بين CNN,LSTM, يمكننا الاستفادة منه في مشروعنا لاستخدامه خوارزميات تصنيف مفيدة جدا في التعلم العميق.

تعتمد المنهجية على:

* مجموعتي بيانات: UCI Spoofing Websites (11,055 عينة) و PhishTank (58,645 عينة).
* معالجة البيانات الأولية (Data Preprocessing):

التوحيد (Normalization): توحيد السمات الرقمية.

التجزئة على مستوى الأحرف (Character-Level Tokenization): تحويل كل حرف في الـ URL

إلى رقم باستخدام أداة Tokenizer في مكتبة Keras.

الحشو (Padding): تم تحديد طول ثابت لجميع الـ URLs هو 60 حرفًا (يُقطع الأطول ويُحشى الأقصر بالأصفار).

* ومن بعدها نقوم ببناء النموذج على عدة طبقات:

الطبقة الأولى: Embedding Layer لتحويل التسلسل الرقمي إلى متجهات.

الطبقة الثانية: Conv1D Layer (بـ 64 مرشح filter بحجم 5) لاستخراج السمات المحلية من التسلسل.

الطبقة الثالثة: MaxPooling1D Layer (بحجم نافذة 2) لاختزال الأبعاد.

الطبقة الرابعة: LSTM Layer (بُعد إخراج 64) لفهم التبعيات طويلة المدى في التسلسل.

الطبقة الخامسة: Dense Layer (طبقة متصلة بالكامل) مع دالة Sigmoid للتصنيف النهائي.

* ومن بعدها نبدأ بتدريب النموذج.

الهدف من هذه المنهجية: الجمع بين قدرة CNN على استخراج السمات المحلية وقدرة LSTM على التعلم من التسلسلات الطويلة.

إيجابيات هذه المنهجية: الميزات: يجمع بين مزايا CNN وLSTM، أداء عالٍ في الكشف، ومناسب للتطبيقات الزمنية الحقيقية.

الدقة: على مجموعة UCL: 98.9%, على مجموعة PhishTank: 96.8%.

**"An integrated CSPPC and BiLSTM framework for malicious URL detection"[7]**

قام كل من Jinyang Zhou, Kun Zhang, Anas Bilal, Yu Zhou, Yukang Fan, Wenting Pan, Xin Xie & Qi Peng عام 2025, باستخدام نموذج CSPPC-BiLSTM الذي يجمع بين:

:Bidirectional Long Short-Term Memory تجمع بين اثنين من نماذج LSTM, واحد يعالج التسلسل من البداية إلى النهاية (الاتجاه الأمامي)، والآخر يعالجه من النهاية إلى البداية (الاتجاه العكسي)

CBAM (وحدة الانتباه القنوية والمكانية): CBAM هي اختصار لـ Convolutional Block Attention Module, تركيز انتقائي على الأجزاء المهمة في البيانات, الانتباه القنوي (Channel Attention): تحديد "ماذا" هو المهم (أي القنوات/السمات الأكثر أهمية), الانتباه المكاني (Spatial Attention): تحديد "أين" هو المهم (أي المواقع في التسلسل).

SPP (التجميع الهرمي المكاني): SPP هي اختصار لـ Spatial Pyramid Pooling تجميع متعدد المقاييس لالتقاط الأنماط بمستويات مختلفة من الدقة.

ومنه يكون التكامل في نموذج CSPPC-BiLSTM:

BiLSTM ← يفهم السياق الزمني ثنائي الاتجاه

CBAM ← يركز على السمات والمواقع الأكثر أهمية

SPP ← يلتقط الأنماط بمقاييس متعددة

والذي يمكننا الاستفادة منه وبقوة.

تعتمد هذه المنهجية على:

* مجموعتي بيانات: Grambedding (متوازنة) وMendeley (غير متوازنة).

الهدف من هذه المنهجية هو تحسين دقة الكشف من خلال تحويل سمات التسلسل إلى تمثيل مكاني متعدد الأبعاد باستخدام آليات الانتباه والتجميع متعدد المستويات.

إيجابيات هذه المنهجية: الميزات: دقة فائقة، مرونة في التعامل مع البيانات المتوازنة وغير المتوازنة، واستخدام آليات انتباه متقدمة.

الدقة: على مجموعة Grambedding (متوازنة): 99.66%

على مجموعة Mendeley (غير متوازنة): 98.89%

**"Optimizing Malicious Website Detection with the XGBoost Machine Learning Approach"[8]**

قام كل من Fazal Malik, Muhammad Suliman, Muhammad Qasim Khan, Noor Rahman, Khairullah Khan, Muhammad Khan عام 2024 باستخدام خوارزمية XGBoost مع تحسين المعاملات باستخدام Particle Swarm Optimization - PSO.

حيث يمكننا الاستفادة منها في مشروعنا لاستخدام خوارزمية ذكاء صنعي فيها.

تعتمد هذه الخوارزمية على:

* مصدر البيانات: تم استخدام مجموعة بيانات جاهزة من Kaggle تسمى "Malicious Webpages Dataset" (قدمها Singh and Kumar, 2020).
* مواصفات البيانات: تحتوي على 1,781 حالة (instance).

تشمل 21 خاصية (feature) لكل موقع، مقسمة إلى:

خصائص لغوية (Lexical): مثل طول الرابط (URL\_LENGTH)، عدد الأحرف الخاصة (NUMBER\_SPECIAL\_CHARACTERS).

خصائص شبكية (Network): مثل عدد حزم البيانات (APP\_PACKETS)، عدد استعلامات DNS (DNS\_QUERY\_TIMES).

خصائص WHOIS: مثل معلومات التسجيل.

* أدوات التنفيذ: تم استخدام لغة Python في بيئة Jupyter Notebook.
* المكتبات المستخدمة: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, XGBoost.

حيث أنه تم تقسيم البيانات إلى قسمين وهما: مجموعة تدريب (Training Set): لبناء النموذج, و مجموعة اختبار (Testing Set): لتقييم أداء النموذج.

حيث أن الهدف منها كان الهدف الرئيسي هو تطوير نظام قوي ودقيق لاكتشاف المواقع الإلكترونية الضارة (Malicious Websites) باستخدام خوارزمية التعلم الآلي XGBoost، وذلك للتغلب على قيود الطرق التقليدية (مثل القواعد الثابتة أو خوارزميات مثل Random Forest و SVM) التي تعاني غالبًا من:

صعوبة تحقيق توازن بين الدقة (Precision) و الاستدعاء (Recall).

ارتفاع معدلات الإنذارات الكاذبة (False Positives).

عدم كفاية الحماية ضد التهديدات الإلكترونية المستجدة والمتطورة.

إيجابيات هذه المنهجية: توازن متفوق في الأداء, موثوقسة عالية, قابلية للتوسع, إطار عمل واضح وقابل للتكرار.

الدقة: (Accuracy): 86.6% , ,(Precision): 88.89%الاستدعاء (Recall): 82.64%.

**"Detecting Phishing URLs Based on a Deep Learning Approach to Prevent Cyber-Attacks"[5]**

قام كل من Qazi Emad ul Haq, Muhammad Hamza Faheem, Iftikhar Ahmad عام 2024 باستخدام منهجية D Convolutional Neural Network (1D-CNN) والتي تقوم بمعالجة الروابط كسلسلة نصوص مع تحقيق دقة فائقة, يمكننا الاستفادة منها في مشروعنا لاستخدامه منهجية مفيدة وجيدة.

تعتمد هذه المنهجية على: مجموعات بيانات من PhishTank وUNB وAlexa (400 ألف رابط).

حيث أن الهدف منها مو تطوير نظام كشف سريع ودقيق باستخدام بنية شبكة عصبية تلافيفية أحادية البعد.

إيجابيات هذه المنهجية: دقة عالية, وسرعة بالتنفيذ, بنية خفيفة, ومناسبة للتطبيقات الواقعية.

الدقة: دقة التدريب: 99.7%, دقة الاختبار: 99.3%, الاستدعاء: 99.5%, الدقة التنبؤية: 99.6% .

**"Detection of malicious URLs using machine learning"[4]**

في2024 قام كل من Nuria Reyes-Dorta, Pino Caballero-Gil, Carlos Rosa-Remedios بمقارنة شاملة بين خوارزميات التعلم الآلي الكلاسيكية: الانحدار اللوجستي، شجرة القرار، SVM بأنواعه (RBF, Poly, Sigmoid)، والشبكات العصبية و خوارزميات التعلم الآلي الكمي: تطبيق نموذج "مصنف الكم المتغير" ويمكن الاستفادة منه في مشروعنا من خلال استخدام المقارنة بين خوارزميات التعلم الآلي.

اعتمدت هذه المنهجية على

* مجموعة البيانات: Hidden Fraudulent URLs Dataset
* دورة PDCA: اتباع دورة "Plan-Do-Check-Act" للتطوير التكراري.
* **معالجة البيانات:** حذف البيانات المفقودة، معالجة المتغيرات الفئوية، إضافة ميزات جديدة من الـ URL نفسه (مثل طول الرابط، عدد الشرطات)، ومكافحة عدم التوازن في البيانات.
* **البيئة:** Python مع مكتبات Scikit-learn و TensorFlow للتعلم الكلاسيكي، و Qiskit للمحاكاة الكمية.

حيث أن الهدف من هذه المنهجية هو تحليل وتطبيق التعلم الآلي الكمي لاكتشاف الروابط الضارة ومقارنة أدائه مع أفضل النماذج الكلاسيكة, مما يمثل خطوة استباقية نحو الجيل القادم من أدوات الأمن السيبراني.

**حيث كانت هذه المنهجية مفيدة جدًا:** أثبتت النماذج الكلاسيكية(شجرة القرار،(SVM-RBF فاعلية عالية

> 93%)دقة( النماذج الكمية أظهرت نتائج "مشجعة" (دقة حتى 97% على مجموعة بيانات مصغرة) رغم حداثة المجال، مما يفتح الباب لأبحاث مستقبلية واعدة.

**تعتمد هذه المنهجية على**: جودة وتمثيل البيانات، المعالجة الصحيحة للبيانات خاصة للبيانات غير المتوازنة.

إيجابيات المنهجية: دراسة شاملة، مقارنة مباشرة بين الكلاسيكي والكمي، معالجة جيدة لعدم توازن البيانات، استباقية في استكشاف تقنيات مستقبلية.

الدقة:  دقة عالية للنماذج الكلاسيكية (تصل إلى ~96%)، ودقة واعدة للنماذج الكمية على بيانات مصغرة.

معدل الوثوقية: مرتفع للنماذج الكلاسيكية بسبب نضج الخوارزميات والنتائج المتماسكة.

متوسط إلى منخفض للنماذج الكمية حاليًا، بسبب حداثة المجال، قيود المحاكاة، والحاجة إلى مزيد من البحث والاختبار على أجهزة حقيقية وبيانات أكبر. ولكنها موثوقة كدليل على المفهوم.

**"Identifying Malicious URLs Using Deep Learning based VGG16 Architecture with Transfer Learning"[9]**

قام كل من Ajay Indian, Gaurav Meena, Krishna Kumar Mohbey, Siddharth Singh Kushwaha عام(2025). الجامعة المركزية في راجاستان، الهند, باستخدام منهجية نقل المعرفة مع VGG16 تم اقتراح تحويل مشكلة اكتشاف الـ URL إلى مشكلة تصنيف صور. يتم تمثيل كل رابط كصورة مصفوفة 24x24 عن طريق ترميز كل حرف في الرابط إلى قيمته الرقمية في ,Unicode ثم يتم استخدام نموذج VGG16 (المدرب مسبقًا على تصنيف الصور) مع تعديل طبقة التصنيف الأخيرة للتعرف على هذه الصور, حيث لا يمكننا الاستفادة من هذه المنهجية في مشروعنا لعدم استخدامه خوارزميات التصنيف للتعلم الآلي أو التعلم العميق.

اعتمدت هذه المنهجية على:

* مجموعة البيانات: Malicious URLs Dataset من Kaggle (651,191 رابط).
* البيئة: Google Colab مع TensorFlow.
* الترميز: ترميز Unicode للحروف لإنشاء التمثيل البصري.

حيث أن الهدف من هذه المنهجية هو التغلب على قيود طرق التعلم الآلي التقليدية التي تعتمد على استخراج الميزات يدوياً, واستخدام قوة التعلم العميق لاكتشاف الميزات المعقدة والخفية تلقائيًا من بنية الرابط النصية، مع الاستفادة من المعرفة المسبقة في النماذج المدربة على الصور.

حيث كانت هذه المنهجية مفيدة وفعالة و أظهرت دقة عالية جدًا دون الحاجة إلى هندسة الميزات يدويًا، مما يجعلها أقل اعتمادًا على الخبرة البشرية وأكثر قدرة على تعميم الأنماط.

إيجابيات المنهجية: تجنب التعقيد في استخراج الميزات يدويًا، أداء متفوق مقارنة بطرق أخرى (مثل XGBoost) على نفس مجموعة البيانات، القدرة على التعامل مع الروابط كبيانات غير مهيكلة.

الدقة: دقة عالية جدًا (95.87%)، مما يجعلها من أفضل الطرق الموثقة.

معدل الوثوقية: مرتفع جدًا، نظرًا للأداء المتميز، واستخدام مجموعة بيانات كبيرة ومتنوعة، ومقارنة النتائج مع طرق أخرى مما يعزز مصداقيتها.

**"Hyperparameter Optimization for Malicious URL Detection"[10]**

قام كل من Miloud Khaldi, Zohra Alilat, Hana Bendoubba, Nadir Mahammed عام (2025) في هذه الورقة باتباع منهجية تحسين المعاملات الفائقة وأهمية ضبط معاملات النماذج بدقة باستخدام نموذج BERT مع Optuna الذي يستخدم خوارزمية .Bayesian نموذج LightGBM مع Random Search استخدام خوارزمية LightGBM مع ضبط معاملاتها باستخدام البحث العشوائي حيث لا يمكننا الاستفادة من هذه المنهجية في مشروعنا لعدم استخدامه خوارزميات التصنيف للتعلم الآلي أو التعلم العميق.

تعتمد هذه المنهجية على:

* مجموعة البيانات Malicious URLs Dataset :من Kaggle.
* إطار التحسين Optuna :للـ BERT، و RandomizedSearchCV لـ LightGBM.
* ميزات استخدام BERT تمثيل النص مباشرة.
* ميزات استخدام LightGBM تم استخدام متجهات.TF-IDF

حيث أن هدف هذه المنهجية إثبات أن تحسين المعاملات الفائقة بشكل منهجي يمكن أن يعزز أداء النماذج بشكل كبير، سواء كانت نماذج تعلم عميق متقدمة (مثل BERT) أو نماذج تعلم آلي تقليدية فعالة (مثل LightGBM)، للوصول إلى أعلى دقة ممكنة.

وتبين ان المنهجية مفيدة بشكل استثنائي: الدراسة تظهر أن ضبط المعاملات ليس رفاهية بل ضرورة، حيث قفز أداء النماذج بشكل ملحوظ.

تعتمد على: وجود موارد حاسوبية قوية (خاصة لتدريب BERT)، اختيار نطاقات المعاملات المنطقية للبحث فيها، وفهم سلوك النموذج.

إيجابيات المنهجية: تحقيق أعلى دقة مسجلة في الأوراق المذكورة، تحليل مقارنة رائع بين نموذج تعلم عميق وآخر تقليدي، إبراز دور تحسين المعاملات.

الدقة: أعلى دقة على الإطلاق (98.84%)، مما يضع معيارًا جديدًا للأداء.

مرتفع للغاية، بسبب النتائج القياسية، والمنهجية الصارمة في التحسين، والتقييم على مجموعة اختبار محجوزة.

**"Phishing Website URL Detection Using a Hybrid Machine Learning Approach"[11]**

قام كل من Muhammad Usman Javeed, Shafqat Maria Aslam, Hafiza Ayesha Sadiqa, Ali Raza, Muhammad Munawar Iqbal, Misbah Akram

عام (2025) باستخدام منهجية مقارنة بين خوارزميات التعلم الآلي التقليدية حيث تم تطبيق وتقييم أربع خوارزميات (شجرة القرار, الغابة العشوائية, SVM, AdaBoost), حيث يمكننا الاستفادة من هذه المنهجية في مشروعنا لاستخدامه خوارزميات للتعلم الآلي.

تعتمد هذه المنهجية على:

* مجموعة البيانات: مجموعة بيانات Phishing Websites من مستودع UCI.
* البيئة: Jupyter Notebook مع Scikit-learn.
* التقنيات: التحقق المتقاطع، تقسيم التدريب/الاختبار.

الهدف من استخدام هذه المنهجية هو تقديم تقييم شامل وموثوق لأداء خوارزميات التعلم الآلي "التقليدية" والقوية في مهمة اكتشاف الروابط التصيدية، مع التأكيد على أفضل الممارسات مثل ضبط المعاملات والتحقق المتقاطع.

إيجابيات المنهجية: بساطة ووضوح المنهجية، تطبيق صارم لأساليب تقييم النموذج، تحديد أفضل نموذج (Random Forest) بشكل واضح.

الدقة: دقة عالية (97.7%)، خاصة considering استخدام خوارزميات "تقليدية".

معدل الوثوقية: مرتفع، بسبب المنهجية السليمة، والنتائج المتسقة، واستخدام تقنيات التحقق المتقاطع التي تعزز ثقة النتائج.

وفيما يلي ملخص ماسبق:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **المجلة والمؤلف** | **التاريخ** | **الموضوع** | **النتائج** | **العلاقة مع المشروع** |
| \*Journal of Electrical Systems and Information Technology\*  Bridget C. Ujah‑Ogbuagu, Oluwatobi Noah Akande and Emeka Ogbuju | 2024 | A hybrid deep learning technique for spoofing website URL detection in real‑time applications | حقق النموذج الهجين (CNN-LSTM) دقة 98.9% على مجموعة UCL و 96.8% على مجموعة PhishTank. | نعم, خوارزميات CNN,LSTM |
| \*Scientific Reports\*  Jinyang Zhou, Kun Zhang, Anas Bilal, Yu Zhou, Yukang Fan, Wenting Pan, Xin Xie & Qi Peng | 2025 | Optimizing Malicious Website Detection with the XGBoost Machine Learning Approach | حقق النموذج دقة 99.66% على مجموعة البيانات المتوازنة و 98.89% على مجموعة البيانات غير المتوازنة. | نعم, خوارزميات BiLSTM، CBAM، SPP |
| \*Journal of Computing &Biomedical Informatics\*  Fazal Malik, Muhammad Suliman, Muhammad Qasim Khan, Noor Rahman, Khairullah Khan, Muhammad Khan | 2024 | Optimizing Malicious Website Detection with the XGBoost Machine Learning Approach | حقق النموذج دقة 86.6%، ودقة تنبؤية 88.89%. | نعم, تعلم آلي و XGBoost |
| Applied Sciences\*\*  Qazi Emad ul Haq, Muhammad Hamza Faheem, Iftikhar Ahmad | 2024 | Detecting Phishing URLs Based on a Deep Learning Approach to Prevent Cyber-Attacks | حقق النموذج دقة 99.7% باستخدام شبكة CNN أحادية البعد. | نعم, 1D-CNN |
| \* Wireless Networks\*  Nuria Reyes‑Dorta,  Pino Caballero‑Gil,  Carlos Rosa‑Remedios | 2024 | Detection of malicious URLs using machine learning | حقق أفضل نموذج دقة 96.35%  وحقق أفضل نموذج كمي (VQC): دقة 97% على بيانات مبسطة. | نعم, التعلم الآلي والتعلم العميق والتعلم الكمي |
| \*Procedia Computer Science\*  Ajay Indian، Gaurav Meena، Krishna Kumar Mohbey، Siddharth Singh Kushwaha | 2025 | Identifying Malicious URLs Using Deep Learning based VGG16 Architecture with Transfer Learning | دقة 95.87% في تصنيف الروابط الضارة إلى 4 فئات. | نعم, تعلم عميق VGG16 مع نقل التعلم |
| Informatica\*\*  Miloud Khaldi, Zohra Alilat, Hana Bendoubba, Nadir Mahammed | 2025 | Hyperparameter Optimization for Malicious URL Detection | BERT&Optuna=98.84%  LightGBM&RandomSearch=80.46% | نعم، التعلم الآلي (LightGBM) والتعلم العميق (BERT) |
| \*Journal of Computing & Biomedical Informatics\*  Muhammad Usman Javeed, Shafqat Maria Aslam, Hafiza Ayesha Sadiqa, Ali Raza, Muhammad Munawar Iqbal, Misbah Akram | 2025 | Phishing Website URL Detection Using a Hybrid Machine Learning Approach | أفضل نموذج (Random Forest): دقة 97.7% | نعم, التعلم الآلي |

**التحديات:**

1. التكيف مع الهجمات سريعة التطور (Zero-Day Attacks): المهاجمون يطورون باستمرار تقنيات جديدة لتجنب الكشف، مما يجعل النماذج التقليدية غير قادرة على التعرف على الهجمات الجديدة.[12]

2. عدم توازن البيانات (Class Imbalance): datasets تحتوي على عدد كبير من الـ URLs الآمنة مقارنة بالضارة، مما يؤدي إلى تحيز النماذج نحو الفئة الغالبة.[7]

3. التمويه والتشويش (Obfuscation and Evasion Techniques): يستخدم المهاجمون تقنيات مثل اختصار الروابط، إضافة أحرف عشوائية، أو محاكاة الروابط الشرعية لتضليل أنظمة الكشف.[13]

4. الحاجة إلى الكشف في الزمن الحقيقي (Real-Time Detection Requirements): التطبيقات العملية تتطلب كشفًا فوريًا دون التأثير على أداء النظام، مما يشكل تحديًا تقنيًا كبيرًا.[6]

5. التعميم على مجموعات بيانات متنوعة (Generalization Across Diverse Datasets): النماذج المدربة على dataset معين قد تفشل في التعميم على بيانات من مصادر أخرى[14].

6. التكلفة الحسابية والتعقيد (Computational Cost and Complexity): النماذج العميقة المتقدمة تتطلب موارد حسابية كبيرة، مما يحد من إمكانية نشرها على نطاق واسع.[5]

7. الهجمات التنافسية (Adversarial Attacks): مهاجمون يستخدمون تقنيات مثل GANs لإنشاء أمثلة خادعة تخدع النماذج المتعلقة.[13]

8. تفسير النتائج وشفافية النموذج (Model Interpretability and Transparency): صعوبة فهم سبب تصنيف النموذج لرابط معين كضار، مما يحد من قبول هذه الأنظمة في البيئات الحساسة.[15]

**التطبيقات العملية:**

1. متصفحات الويب ووظائف الحماية المدمجة (Web Browsers & Built-in Protection): دمج أنظمة الكشف في المتصفحات مثل Google Chrome و Firefox لتحذير المستخدمين عند زيارة مواقع ضارة.[16]

2. أنظمة كشف البريد الإلكتروني الضار (Email Security Systems): تحليل الروابط في البريد الإلكتروني للكشف عن محاولات التصيد (Phishing) وهجمات البرمجيات الخبيثة.[17]

3. تطبيقات الجوال والأمن النقال (Mobile Applications & Mobile Security): حماية مستخدمي الهواتف الذكية من الروابط الضارة في التطبيقات والرسائل النصية.[18]

4. أنظمة حماية الشبكات المؤسسية (Enterprise Network Protection): مراقبة وتحليل نشاط الشبكة في المؤسسات الكبيرة للكشف عن الاتصالات الضارة.[19]

5. أنظمة الأمن السحابي (Cloud Security Systems): حماية التطبيقات والخدمات السحابية من الهجمات التي تستهدف البنية التحتية السحابية [20].

6. أنظمة الأمن الصناعي وإنترنت الأشياء (Industrial IoT Security): حماية الأنظمة الصناعية وأجهزة IoT من الهجمات الإلكترونية التي تستهدف البنية التحتية الحرجة.[21]

7. أنظمة الأمن المالي والمصرفي (Financial & Banking Security): حماية العملاء والمؤسسات المالية من عمليات الاحتيال الإلكتروني والتصيد.[22]

8. منصات التجارة الإلكترونية (E-commerce Platforms): حماية المتسوقين عبر الإنترنت من مواقع التصيد التي تحاول سرقة معلومات الدفع.[23]

9. أنظمة الأمن السيبراني للأجهزة الطبية (Medical Device Cybersecurity): حماية الأجهزة الطبية المتصلة بالإنترنت من الهجمات الإلكترونية.[24]

**الفصل الاول:**

**محدد مواقع الموارد Uniform Resource Locator (URL)**

* 1. **مقدمة:**

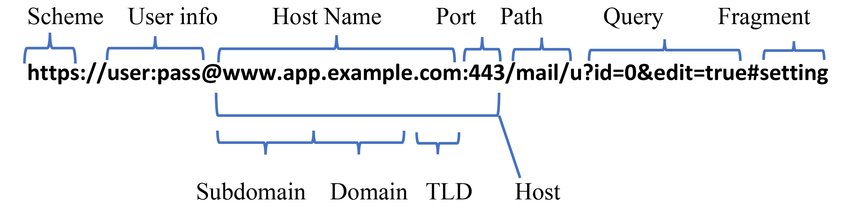
يشار إلى محدد مواقع الموارد بشكل شائع باسم عنوان ويب أو موقع أو عنوان موقع ويب وبواسطته يتم تحديد مواقع وموارد الإنترنت. وهو ذلك العنوان الذي تكتبه في شريط العنوان للانتقال إلى مواقع صفحات الانترنت ويستخدم أيضاً لنقل الملفات (FTP) والبريد الالكتروني والوصول إلى قاعدة البيانات والعديد من التطبيقات الاخرى على الانترنت [25].

* 1. **عنوان URL:**

هو توصيف منظم لموقع المورد الرقمي [25]، حيث تعمل محددات مواقع الموارد (URLs) كوسيلة أساسية يمكن من خلالها لنسبة كبيرة من مستخدمي الإنترنت الوصول إلى المعلومات الموجودة على الشبكة العالمية. ويعد استخدام الروابط (Links) أسلوب شائع يستخدم لتوجيه المستخدمين نحو محتوى معين، سواء كان ذلك صفحات ويب أو رسائل دردشة أو وسائل التواصل الاجتماعي أو حتى رسائل البريد الإلكتروني [26].

**1.3 صيغة محدد موقع الموارد:**

يتكون محدد موقع الموارد (URL) من الأقسام التالية، وهي المخطط (Scheme)، والمجال الفرعي (Subdomain)، ومجال المستوى الأعلى (Top-level domain)، ومجال المستوى الثاني (Second-level domain)، وسلسلة الاستعلام (Query String)، والمعلمة (Parameter)، والمنفذ (Port)، والمسار (Path)، والجزء (Fragment). كما هو موضح في الشكل (1) ولكن يجب الإشارة هنا إلى أن عنوان URL ليس ملزمًا بتضمين كل هذه المكونات في وقت واحد.



الشكل 1 الصيغة العامة لمحدد موقع الموارد (URL) [25]

* 1. **أقسام محدد موقع الموارد:**

**1.4.1** **المخطط (Scheme):**

يقوم المخطط، الذي يشار إليه عادةً باسم البروتوكول، بإرشاد خوادم الويب حول البروتوكول المناسب لاستخدامه للوصول إلى صفحة ويب على موقع الويب الخاص بك .[url9 ] في الوقت الحاضر، البروتوكول الأكثر استخداماً من قبل المستخدمين هو بروتوكول نقل النص التشعبي الآمن (HTTPS). حيث يوجه هذا البروتوكول متصفح الويب الخاص بك لتشفير جميع البيانات التي تدخلها على صفحة الويب، مثل كلمات المرور أو تفاصيل بطاقة الائتمان، لمنع الوصول غير المصرح به من قبل مجرمي الإنترنت.

**1.4.2** **المجال الفرعي (Subdomain):**

يمتلك نظام اسم النطاق (DNS) بنية هرمية، تتميز بتنظيمها الشبيه بالشجرة، حيث تمثل العقد الموجودة على الشجرة أسماء النطاقات.

النطاق الفرعي، كونه جزءًا أساسيًا من نطاق أكبر، يعتبر نطاقًا في حدد ذاته، ومن المهم أن نأخذ في الاعتبار أن اسم النطاق الكامل يجب ألا يتجاوز الطول الإجمالي لـ 253 حرف ASCII عند تمثيله نصًا. من الناحية العملية، يتم استخدام النطاقات الفرعية بشكل متكرر من قبل مقدمي خدمات الإنترنت (ISPs) الذين يقدمون خدمات الويب.

حيث يخصص مزودو خدمة الإنترنت هؤلاء نطاقًا فرعيًا واحدًا أو أكثر للعملاء الذين لا يمتلكون اسم المجال الخاص بهم. ومن خلال القيام بذلك، يكون العملاء قادرين على ممارسة سيطرة وإدارة مستقلة على النطاقات الفرعية الخاصة بهم.

وهناك العديد من النطاقات الفرعية الأساسية والمستخدمة بكثرة، مثل (WWW,FTP). أي أنه قد يحتوي نطاق FTP الفرعي على سجلات بالإضافة إلى أدلة لصفحات الويب، في حين أن النطاق الفرعي WWW مخصص للأدلة المتعلقة بصفحات الويب نفسها. بالإضافة إلى ذلك، يتم تنفيذ المصادقة المستقلة لكل مجال لتنظيم التحكم في الوصول عبر المستويات المختلفة داخل المجال [27].

**1.4.3 نطاق المستوى الثاني (Second-Level Domain)**:

في التسلسل الهرمي لنظام اسم المجال (DNS)، يتم وضع نطاق المستوى الثاني (SLD) مباشرةً أسفل نطاق المستوى الأعلى (TLD)، على سبيل المثال في النطاق Google.com))، يشير المصطلح "Google" إلى نطاق المستوى الثاني ضمن نطاق المستوى الأعلى "com".

وتتعلق نطاقات المستوى الثاني عمومًا بالمؤسسة التي سجلت اسم النطاق من خلال مسجل اسم النطاق [28].

**1.4.4 نطاق المستوى الاعلى (Top -Level Domain)**:

هو أحد النطاقات الموجودة في المستوى الأقصى ضمن نظام اسم النطاق الهرمي للإنترنت. يتم إنشاء أسماء نطاقات المستوى الأعلى في منطقة الجذر [[1]](#footnote-1)لمساحة الاسم. وفيما يتعلق بجميع النطاقات في المستويات الأدنى، فهي تمثل الجزء الأخير من اسم النطاق، وعلى وجه التحديد، التسمية الأخيرة غير الفارغة لاسم النطاق المؤهل بالكامل. على سبيل التوضيح، في اسم النطاق (www.google.com)، نطاق المستوى الأعلى هو ."com" إسناد مسؤولية إدارة معظم نطاقات المستوى الأعلى إلى منظمات معينة من قبل (ICANN[[2]](#footnote-2))، وهو مجتمع أصحاب المصلحة المتعددين على الإنترنت، والذي تحكمه هيئة الإنترنت للأرقام المخصصة (IANA[[3]](#footnote-3)) [29].

**1.4.5 المنفذ (Port):**

المنفذ أو رقم المنفذ هو قيمة رقمية يتم تعيينها لتمييز نقطة نهاية الاتصال بشكل فريد وتوجيه البيانات نحو خدمة معينة. على مستوى البرنامج، داخل نظام التشغيل، يعد المنفذ كيانًا مفاهيميًا يعمل على تحديد عملية معينة أو نوع معين من خدمة الشبكة. يتم تخصيص منفذ على مستوى البرنامج لكل مجموعة من بروتوكول النقل والعنوان، ويتم تحديده من خلال رقم المنفذ المخصص له. بروتوكولا النقل الأكثر استخدامًا اللذين يستخدمان أرقام المنافذ هما بروتوكول التحكم في الإرسال (TCP) وبروتوكول مخطط بيانات المستخدم(UDP)، ومن الضروري أن يتم ربط رقم المنفذ دائمًا بعنوان الشبكة الخاص بالمضيف، مثل عنوان IP ونوع بروتوكول النقل المستخدم للاتصال.

حيث يتم حجز أرقام منافذ معينة خصيصًا لتحديد خدمات معينة، وبالتالي تمكين إعادة توجيه الحزمة الواردة بسهولة إلى تطبيق قيد التشغيل. ولتحقيق هذا الغرض، يتم استخدام أرقام المنافذ الأقل من 1024 لتحديد الخدمات الأكثر استخدامًا تاريخيًا، ويشار إليها بأرقام المنافذ المعروفة. وعلى العكس من ذلك، يتم توفير المنافذ ذات الأرقام الأعلى للاستخدام العام بواسطة التطبيقات ويشار إليها بالمنافذ المؤقتة.

ويتم تقسيم أرقام المنافذ إلى ثلاثة أقسام: المنافذ المعروفة كما في الجدول (1)، والمنافذ المسجلة، والمنافذ الديناميكية أو الخاصة، وتكون المنافذ المعروفة (منافذ النظام) هي تلك المرقمة من 0 إلى 1023 [31] [30].

جدول 1 المنافذ الأكثر استخداماً

|  |  |
| --- | --- |
| **الرقم** | **الوظيفة** |
| 20 | بروتوكول نقل الملفات (FTP) نقل البيانات |
| 21 | التحكم في أوامر بروتوكول نقل الملفات (FTP). |
| 22 | تسجيل الدخول الآمن لـ Secure Shell (SSH). |
| 23 | خدمة تسجيل الدخول عن بعد Telnet، رسائل نصية غير مشفرة |
| 25 | تسليم البريد الإلكتروني عبر بروتوكول نقل البريد البسيط (SMTP). |
| 53 | خدمة نظام أسماء النطاقات (DNS). |
| 67،68 | بروتوكول التكوين الديناميكي للمضيف (DHCP) |
| 80 | بروتوكول نقل النص التشعبي (HTTP) المستخدم في شبكة الويب العالمية |
| 110 | بروتوكول مكتب البريد (POP3) |
| 119 | بروتوكول نقل أخبار الشبكة (NNTP) |
| 123 | بروتوكول وقت الشبكة (NTP) |
| 143 | بروتوكول الوصول إلى رسائل الإنترنت (IMAP) إدارة البريد الرقمي |
| 161 | بروتوكول إدارة الشبكة البسيط (SNMP) |
| 194 | الدردشة عبر الإنترنت (IRC) |
| 443 | HTTP الآمن (HTTPS) HTTP عبر TLS/SSL |
| 564،547 | إصدار DHCPv6 IPv6 من DHCP |

* + 1. **المسار (Path):**

يشير المسار إلى الموقع الدقيق لصفحة الوب أو المورد أو الملف أو أي أصل آخر، وغالبًا ما يكون مشابهًا لبنية الملف الأساسي لموقع الويب ويكتب المسار بعد اسم المضيف ويتم فصله بـ "/" (شرطة مائلة للأمام) [url9]

**1.4.7** **سلسلة الاستعلام (Query String) والمعلمات (Parameters):**

سلسلة الاستعلام هي عنصر من محدد موقع الموارد الموحد (URL) وتأتي بعد علامة الاستفهام (؟) وتقوم بتخصيص القيم لمعلمات معينة ويتم فصل كل معلمة باستخدام علامة (&).

تشتمل سلسلة الاستعلام بشكل عام على المكونات الملحقة بعنوان URL الأساسي بواسطة مستعرض ويب أو تطبيق عميل آخر على سبيل المثال كجزء من مستند (Html) أو لتحديد التخطيط المرئي للصفحة أو للانتقال إلى أقسام معينة داخل محتوى الوسائط المتعددة [32].

**1.4.8 الأجزاء (Fragments):**

الأجزاء، التي توجد عادةً في نهاية عناوين URL ويتم تحديدها بعلامة (#)، تعمل كعنصر اختياري، حيث تعمل هذه الأجزاء على تحديد مكان معين على صفحة الويب، مثل المعرف أو سمة الاسم. كما تستخدم للقيام بإعادة التوجيه إلى موارد إضافية، على سبيل المثال، التذييل أو الشريط الجانبي [33].

**1.5 الخاتمة:**

قدم هذا الفصل نبذة تعريفية عن محددات مواقع الموارد (URLs) حيث تم القاء النظر على تعريف محدد مواقع الموارد، وتوضيح البنية العامة له كما تطرقنا إلى شرح كل قسم من أقسام العنوان وتوضيح أهمية هذه الأقسام واستخداماتها.

**الفصل الثاني:**

**بيئة العمل**

بيئات العمل المتاحة:

في كل من الأوراق البحثية [36],[11],[10],[9],[5] تم استخدام بيئتي Google Colab و Anaconda.

|  |  |  |
| --- | --- | --- |
| **بيئة العمل** | **مزايا** | **مساوئ** |
| **Google Colab** | - مجانية بالكامل للاستخدام الأساسي.  - توفر موارد حاسوبية مجانية (GPU وTPU) لتسريع التدريب.  - التكامل مع Google Drive. | - محدودة الموارد في النسخة المجانية (دورة حياة الجلسة، ذاكرة).  - بطيئة نسبيًا مقارنة بالخوادم المحلية القوية.  - تتطلب اتصال إنترنت. |
| **Anaconda** | - عدم الحاجة لاتصال إنترنت دائم بعد التثبيت.  - خصوصية كاملة للبيانات والنماذج.  - مرونة غير محدودة في تثبيت المكتبات وتعديلها. | - مكلفة إذا كنت تحتاج لشراء أجهزة قوية (GPU).  - مشاكل في التوافق بين المكتبات قد تستهلك وقتًا.  - المسؤولية عن تحديث البيئة وصيانتها. |

بناءً على المقارنة أعلاه والأهداف المذكورة في الأوراق (خاصة الأوراق التي ركزت على تطوير نماذج قابلة للتطبيق)، فإن الخيار الأمثل هو استخدام Google Colab.

الأسباب:

ملاءمة للأبحاث والتطوير الأولي: جميع الأوراق تقريبًا استخدمت Colab أو أوصت به للتجارب، مما يجعله البيئة القياسية للأكاديميين والمطورين.

تبسيط عملية الإعداد: الهدف هو توفير بيئة يمكن لأي شخص إعدادها بسرعة. Colab يحل مشكلة التعقيد في الإعداد المحلي.

توفير الموارد الحاسوبية: تدريب نماذج التعلم العميق (مثل VGG16، BERT، 1D-CNN) يتطلب قوة معالجة. يوفر Colab هذه الموارد مجانًا، مما يلغي الحاجة لامتلاك جهاز قوي.

التكرارية وسهولة الاختبار: يسمح Colab بتشغيل الكود وتعديله وتجربة نماذج مختلفة بسرعة وسهولة.

دعم المكتبات: جميع المكتبات المذكورة في الأوراق (TensorFlow, Keras, PyTorch, Scikit-learn, Pandas, Numpy) مثبتة مسبقًا وجاهزة للعمل.

لذلك، سيتم اعتماد Google Colab كبيئة العمل المعيارية.

توصيف لكيفية تنصيب البيئة وربطها:

هذا الدليل المفصل سيسمح لأي شخص بتشغيل البيئة خطوة بخطوة.

الخطوة 1: الحصول على حساب Google: إذا لم يكن لديك حساب Gmail، نقوم بإنشاء حساب من خلال موقع Gmail.

الخطوة 2: الدخول إلى Google Colab: نقوم بتسجيل الدخول باستخدام حساب Google الخاص بك.

الخطوة 3: إنشاء دفتر ملاحظات (Notebook) جديد: من الصفحة الرئيسية، ننقر على "New notebook":سيتم فتح نافذة جديدة تحتوي على خلية كود قابلة للتعديل.

الخطوة 4: التحقق من الإعدادات الأساسية وربطها بـ GPU (اختياري لكن موصى به)

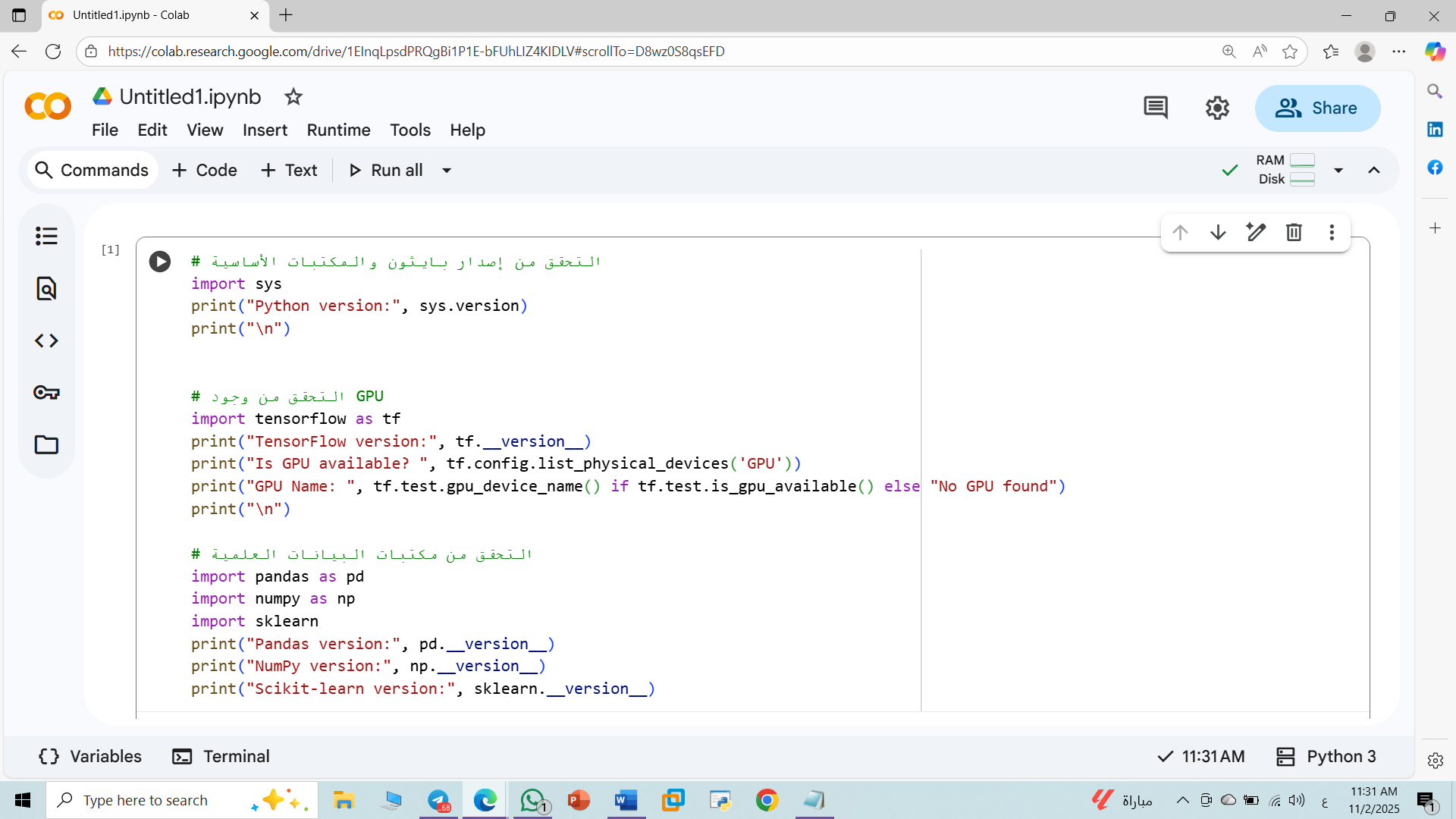
من القائمة العلوية، ننقر على "Runtime".

من القائمة المنسدلة، نختر "Change runtime type".

في النافذة المنبثقة، بجوار "Hardware accelerator"، نختر "GPU" من القائمة المنسدلة.

ننقر على "Save".

الخطوة 5: التحقق من تثبيت المكتبات والتكوين: للتأكد من أن كل شيء يعمل بشكل صحيح، نقوم بتنفيذ كود التحقق التالي في الخلية الأولى ثم نضغط على Ctrl+Enter أو ننقر على زر التشغيل.



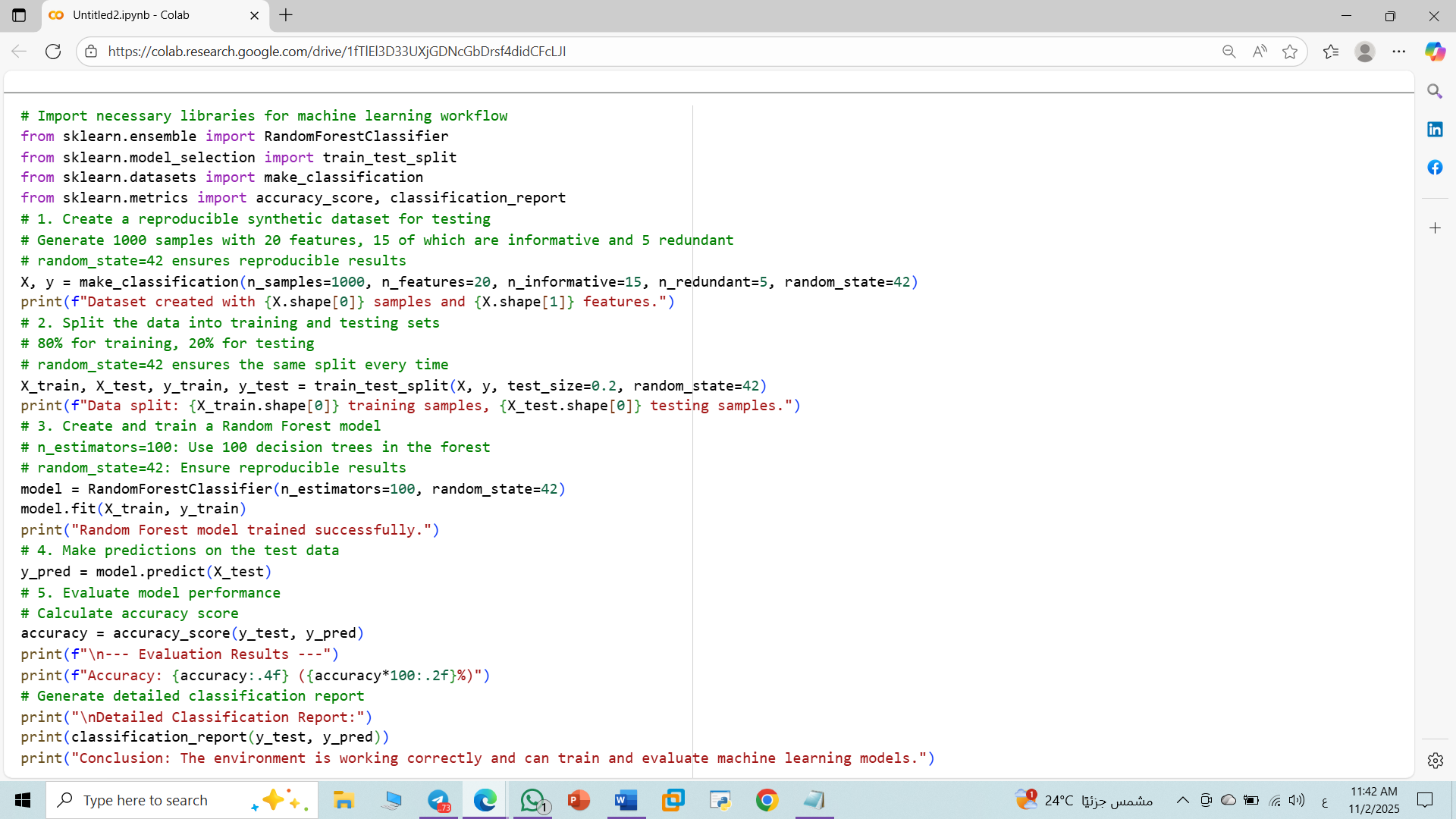
النتيجة المتوقعة من كود تجهيز المكاتب:



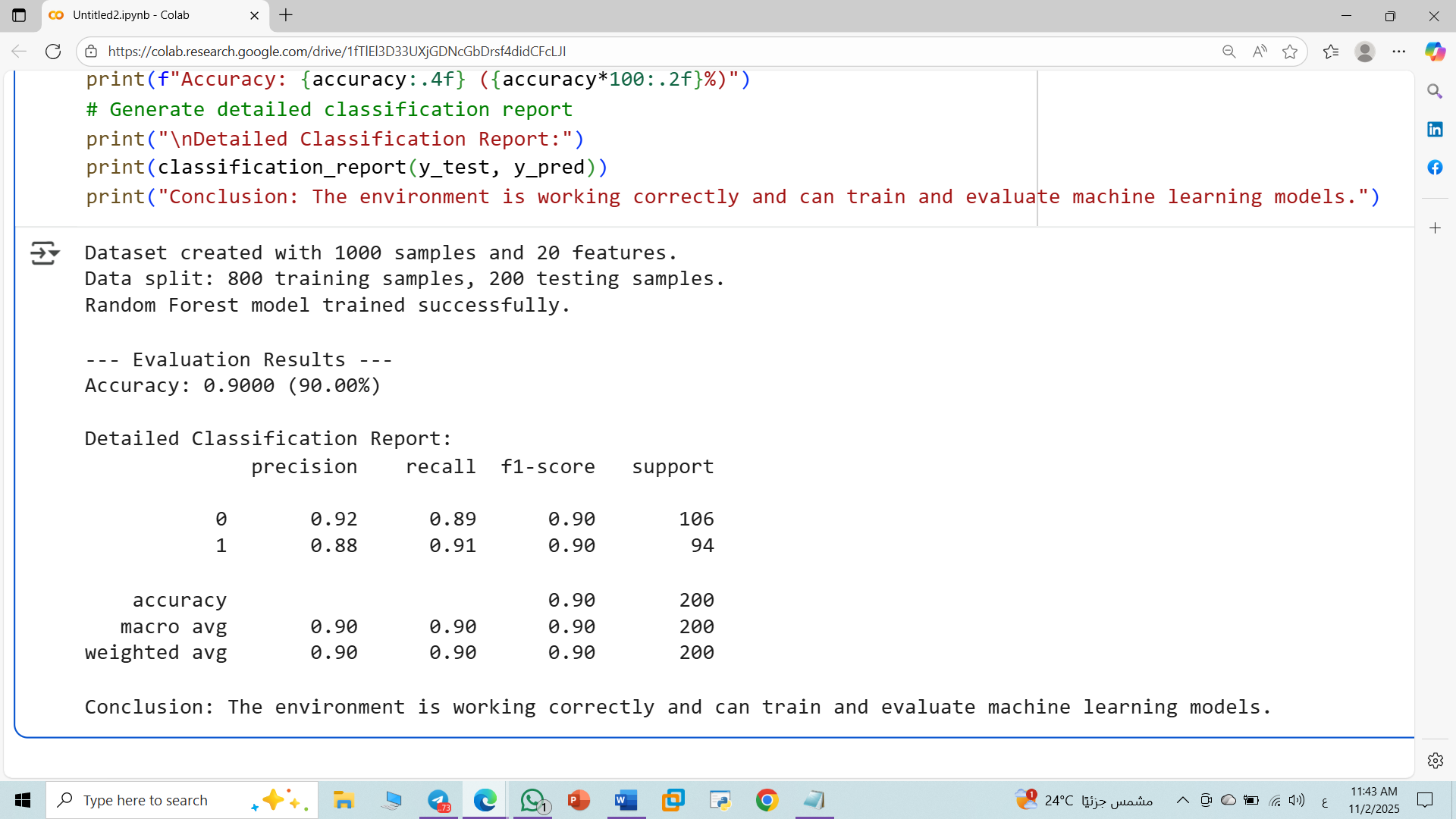
تطبيق معياري على البيئة:

لإثبات أن البيئة تعمل بشكل صحيح وقادرة على تنفيذ مهام التعلم الآلي الأساسية، سنقوم بتنفيذ نموذج تصنيف بسيط باستخدام مجموعة بيانات "أرقام" الشهيرة (MNIST-like لكننا سنستخدم make\_classification من sklearn للبساطة).

الهدف: تدريب مصنف غابة عشوائية (Random Forest) للتمييز بين فئتين.



النتيجة المتوقعة من تدريب هذا النموذج التجريبي:



تفسير النتيجة:

تشير الدقة العالية (~90%) إلى أن النموذج قد تعلم الأنماط في البيانات الاصطناعية.

يشير تقرير التصنيف المتوازن بين الفئتين (0 و 1) إلى أن النموذج يعمل بشكل جيد دون تحيز.

نجاح جميع خطوات الكود (استيراد المكتبات، إنشاء البيانات، التدريب، التقييم) هو دليل قاطع على أن بيئة Google Colab مثبتة ومكونة بشكل صحيح وجاهزة لتطوير نماذج أكثر تعقيدًا لاكتشاف الروابط الخبيثة.

**الفصل الثالث:**

**الحل المقترح**

**3. 3.1 المقدمة:**

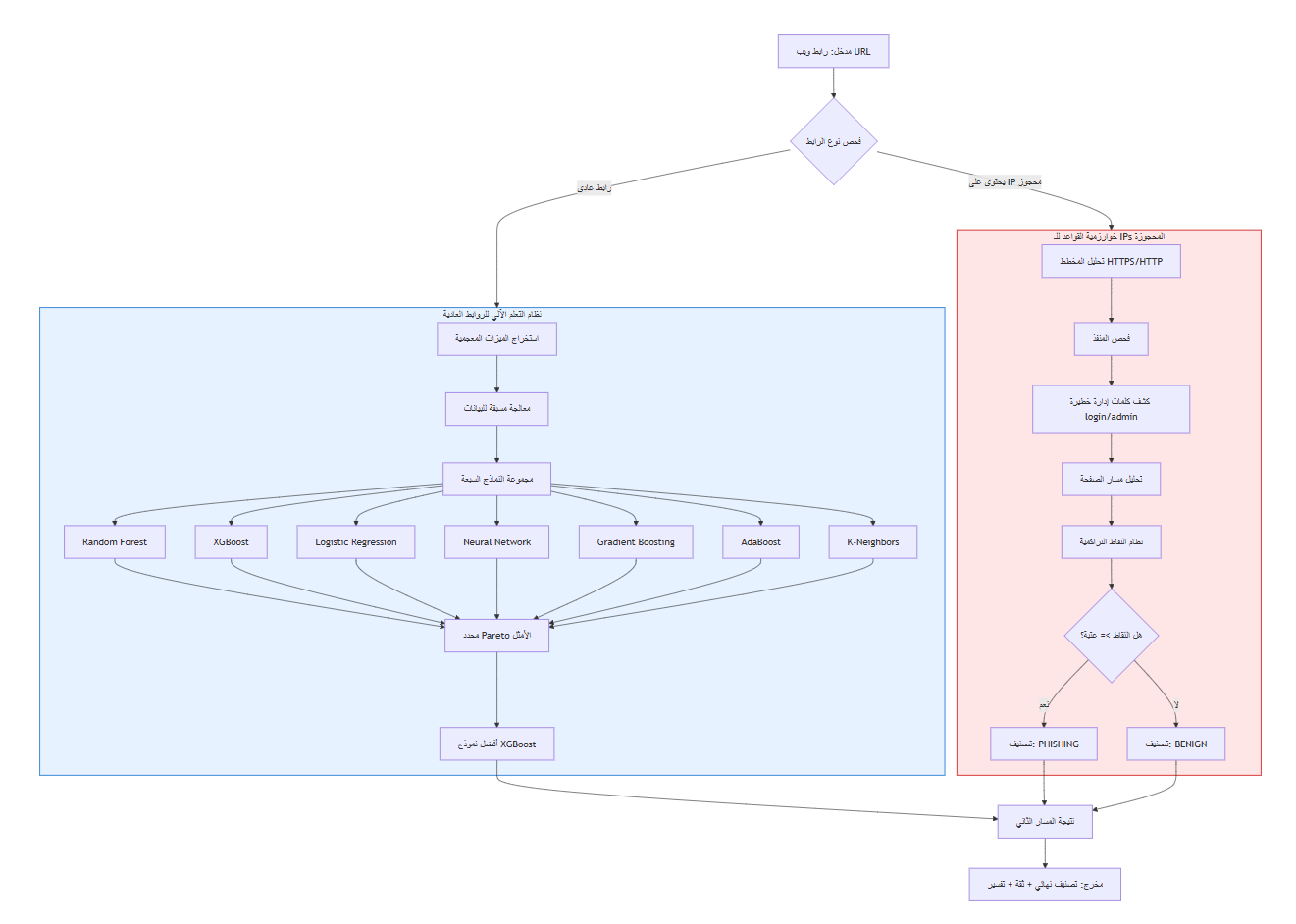
في عصر التكنولوجيا الحديثة، أصبحت هجمات التصيد الإلكتروني واحدة من أخطر التهديدات الأمنية التي تواجه الأفراد والمؤسسات على حد سواء. تتطور هذه الهجمات باستمرار وتصبح أكثر تطوراً وتعقيداً، مما يستدعي تطوير أنظمة كشف متطورة قادرة على مواكبة هذا التطور. يقدم هذا الفصل الحل المقترح لنظام متكامل لاكتشاف هجمات التصيد باستخدام روابط الويب، مع تركيز خاص على نوع خطير من هذه الهجمات التي تستخدم عناوين IP محجوزة (Reserved IPs) لخداع المستخدمين.

يهدف النظام المطور إلى توفير حل عملي وفعال يجمع بين دقة خوارزميات تعلم الآلة المتقدمة ومرونة النظم القاعدية (Rule-based)، مع التركيز على تحقيق التوازن الأمثل بين الدقة العالية وسرعة المعالجة. تم تصميم النظام ليكون شاملاً وقابلاً للتطوير، بحيث يمكن استخدامه في مختلف السيناريوهات العملية بدءاً من البحث الأكاديمي وصولاً إلى التطبيقات الأمنية في المؤسسات.

**3.2 نظرة عامة على النظام المقترح:**

تم تطوير نظام متكامل ومتعدد الطبقات لاكتشاف هجمات التصيد (Phishing) باستخدام روابط الويب، مع تركيز خاص على الروابط التي تحتوي على عناوين IP محجوزة (Reserved IPs) يعتمد النظام على دمج تقنيات تعلم الآلة التقليدية مع خوارزميات قاعدية (Rule-based) متخصصة لتحقيق دقة عالية وسرعة في الكشف.

**3.3 مخطط العمل:**

**مخطط تدفق النظام الهجين المقترح للكشف عن التصيد.**

يتكون الحل المقترح من المكونات الرئيسية التالية:

وحدة التوجيه الذكي (Intelligent Router): هذه هي الخطوة الأولى في النظام. تقوم بتحليل الرابط الوارد وفحصه باستخدام أنماط regex محددة مسبقًا للتعرف على ما إذا كان يحتوي على عنوان IP ينتمي إلى النطاقات المحجوزة المعروفة (مثل 10.x.x.x، 192.168.x.x، 172.16.x.x، 127.x.x.x، 169.254.x.x). بناءً على نتيجة هذا الفحص، يتم توجيه الرابط إلى أحد المسارين التاليين:

المسار الثاني - خوارزمية القواعد المتخصصة (Specialized Rule-Based Algorithm): يتم توجيه جميع الروابط التي تحتوي على IPs محجوزة إلى هذه الخوارزمية. تعتمد هذه الخوارزمية على مجموعة موسعة ومحسنة من القواعد الاستدلالية التي تفحص:

مخطط البروتوكول: غياب HTTPS يضيف نقاط خطر كبيرة.

المنفذ: المنافذ غير القياسية (غير 80 أو 443) تزيد من درجة الشك.

المسار والاستعلام: البحث عن كلمات مفتاحية عالية الخطورة في مسار الرابط أو معاملاته (مثل login، admin، password، bank، verify، authenticate، payment). وجود أكثر من كلمة خطيرة، خاصة على IP محجوز، يرفع العتبة ويشير بقوة إلى التصيد.

نوع الـ IP: بعض النطاقات (مثل 169.254.x.x - Link-local أو 127.x.x.x - Loopback) تعتبر أكثر خطورة عند الوصول إليها من خارج نطاقها المحدد.

سياق الصفحة: يتم تقييم إذا كان الرابط يشير إلى صفحة "إدارة" أو "بوابة" شائعة قد تكون شرعية في الشبكات الداخلية، مما قد يخفف من حكم الخوارزمية قليلاً.

تقوم الخوارزمية بحساب نقاط خطر تراكمية ومقارنتها ب عتبة ديناميكية (تتغير بناءً على وجود كلمات إدارة خطيرة). إذا تجاوزت النقاط العتبة، يصنف الرابط على أنه PHISHING، وإلا يصنف كـ BENIGN. تُرجع الخوارزمية أيضًا درجة ثقة وسبب القرار.

المسار الاول - نظام نماذج التعلم الآلي للروابط العادية (ML Model System for Normal URLs): يتم توجيه جميع الروابط العادية (التي تحتوي على نطاقات اسمية) إلى هذا المسار. يتكون هذا النظام من:

مستخرج الميزات المحسن (Optimized Feature Extractor): يستخرج مجموعة غنية من السمات المعجمية والهيكلية (Lexical & Structural Features) من الرابط، مثل الطول، عدد النقاط، وجود @، عدد المعلمات، نوع نطاق المستوى الأعلى (TLD)، وغيرها.

مجموعة النماذج السبعة (The Top-7 Models Ensemble): يتم تدريب وتقييم سبعة من خوارزميات التعلم الآلي والتعلّم العميق الأكثر تنوعًا وفعالية على مجموعة البيانات الرئيسية للروابط العادية. تضم هذه المجموعة:

Random Forest (نموذج تجميعي قوي).

XGBoost (نموذج تعزيز متقدم).

Logistic Regression (نموذج خطي كلاسيكي).

Neural Network (MLP) (نموذج تعلم عميق).

Gradient Boosting (نموذج تعزيز).

AdaBoost (نموذج تعزيز تكيفي).

K-Neighbors (نموذج قائم على المسافة).

محدد Pareto الأمثل (Pareto Optimal Selector): بدلاً من اختيار النموذج ذي الدقة الأعلى فقط، يستخدم هذا المحدد معايير متعددة الأهداف (Multi-Objective) مستوحاة من أمامية Pareto (Pareto Front). يبحث عن النماذج التي تحقق توازنًا متفوقًا بين المقاييس الحاسمة في مجال الأمن السيبراني، وخاصة:

الاستدعاء العالي (High Recall): لضمان التقاط أكبر عدد ممكن من هجمات التصيد الحقيقية (تقليل الإنذارات الكاذبة السلبية False Negatives).

الدقة المعقولة (Reasonable Precision): للحفاظ على جودة التنبؤات الإيجابية.

منطقة تحت منحنى ROC عالية (High AUC-ROC): كمقياس شامل للأداء عبر جميع العتبات.

درجة F1 جيدة (Good F1-Score): كمتوسط توافقي بين الدقة والاستدعاء.

النموذج الذي يقع على "أمامية Pareto" ويحقق أفضل موازنة بين هذه المعايير (غالبًا ما يكون XGBoost حسب نتائج التجارب) يتم اختياره كأفضل نموذج لهذا المسار.

مقارنة وتقييم متخصص (Specialized Comparison & Evaluation): بالإضافة إلى النظام الهجين الرئيسي، يقترح البحث إطار عمل لمقارنة أداء نهجين متخصصين لمعالجة روابط الـ IPs المحجوزة:

نموذج XGBoost المتخصص (Specialized XGBoost Model): يتم تدريب نموذج XGBoost من الصفر باستخدام مجموعة ميزات مصممة خصيصًا لخصائص عناوين IP (مثل أجزاء الـ IP، كونه خاصًا، loopback، إلخ) وذلك على مجموعة بيانات تحتوي فقط على روابط ذات IPs محجوزة.

مقارنة مع خوارزمية القواعد (Vs. Rule-Based Algorithm): يتم إجراء مقارنة منهجية بين نموذج XGBoost المتخصص وخوارزمية القواعد المحسنة على مجموعة اختبار من روابط الـ IPs المحجوزة، مع مراعاة مقاييس الدقة، الثقة، سرعة التنبؤ، وقابلية التفسير.

**3.4 الأدوات والتقنيات المستخدمة:**

**3.4.1 لغات البرمجة والمكتبات الأساسية:**

* **لغة البرمجة** : **Python** اللغة الرئيسية لتطوير النظام لمرونتها وتوفر المكتبات المتخصصة.
* **مكتبات معالجة البيانات**:
  + :pandas معالجة وتنظيف البيانات
  + :numpyالعمليات الحسابية والمصفوفات
  + re: التعامل مع النصوص باستخدام التعبيرات النمطية
* **مكتبات التعلم الآلي**:
  + scikit-learn: لتطبيق خوارزميات التعلم الآلي التقليدية.
  + :XGBoost خوارزمية Gradient Boosting المتقدمة
  + :CatBoostخوارزمية Boosting أخرى
  + :TensorFlow/Kerasللتعلم العميق والشبكات العصبية
* **مكتبات إضافية**:
  + matplotlib, seaborn: التصور البياني والرسوم البيانية
  + :urllib.parse تحليل روابط الويب واستخراج مكوناتها
  + :Ipaddressمعالجة وتصنيف عناوين IP
  + :threadingمعالجة متوازية لتحسين الأداء

**3.4.2 بيئة التنفيذ:**

* **Google Colab** - : البيئة السحابية المجانية المستخدمة لتطوير وتنفيذ النظام
* **ميزات بيئة Google Colab المستغلة:**

o معالجة سحابية مجانية

o دعم GPU لتسريع تدريب النماذج

o تخزين سحابي للملفات الناتجة

o إمكانية المشاركة والتعاون

**3.5 مكونات النظام الرئيسية:**

**3.5.1 مستخرج الميزات المحسن :(OptimizedURLFeatureExtractor)**

* **الوظيفة**: استخراج 33 ميزة متنوعة من روابط الويب
* **الميزات المستخرجة**:
  + **الميزات البنيوية**: طول الرابط، طول النطاق، عدد النقاط، الشرطات, الخطوط السفلية
  + **الميزات الأمنية**: وجود HTTPS ، عدد المعلمات، وجود منفذ, وجود رمز
  + **ميزات المحتوى**: عدد الكلمات المشبوهة، عدد الكلمات الآمنة, توازن الكلمات
  + **ميزات :IP** نوع العنوان (خاص/عام/محجوز)، فئة IP, تصنيفات RFC
  + **ميزات TLD**
* **تحسينات الأداء المتقدمة:**

o **ذاكرة التخزين المؤقت (Caching):** تخزين نتائج تحليل الروابط المتكررة

o **المعالجة المتوازية (Parallel Processing):** استخدام ThreadPoolExecutor

لتحليل 4 روابط في نفس الوقت

o **المعالجة الدفعية (Batch Processing):** تحليل مجموعات كبيرة من الروابط بكفاءة

**3.5.2 معالج البيانات المحسن (OptimizedDataProcessor):**

**• الوظيفة:** معالجة وتحضير البيانات بكفاءة عالية

* **العمليات الرئيسية:**

o استخراج الميزات بشكل دفعي.

o دمج البيانات المستخرجة مع البيانات الأصلية.

o إزالة التكرارات والسجلات غير الصالحة.

* **معالجة متقدمة:**

o معالجة عمود TLD (نطاق المستوى الأعلى) وتحويله إلى قيم رقمية.

o تنظيف قيم NaN و infinity واستبدالها بالقيم المناسبة.

o إعداد البيانات النهائية للتدريب والاختبار.

**3.5.3 مصنف أفضل 7 نماذج :(Top7ModelsClassifier)**

* **النماذج المستخدمة**:

:Random Forest.1 نموذج تجميعي يعتمد على أشجار القرار.

: XGBoost.2 خوارزمية Gradient Boosting متقدمة.

:Logistic Regression.3 نموذج خطي كلاسيكي للتصنيف.

:Neural Network (MLPClassifier).4 شبكة عصبية متعددة الطبقات.

:Gradient Boosting.5 خوارزمية Boosting تقليدية.

AdaBoost.6 : تعزيز تكيفي.

:K-Neighbors.7 نموذج قائم على المسافات والنقاط المجاورة.

**•** **آلية اختيار النموذج الأمثل:**

**Pareto Optimal Front** o : تقنية متقدمة لاختيار النموذج الذي يوازن بين مقاييس متعددة (الدقة،

الاستدعاء، AUC)

o  **آلية الاختيار**: **Pareto Optimal Front**  مع شروط (الدقة ≥ 70%، الاستدعاء ≥ 80%)

**o نظام ترجيح متعدد المعايير:** وزن مختلف المقاييس حسب الأهمية.

**3.5.4 خوارزمية :(ReservedIPPhishingDetector)**

* **الهدف**: اكتشاف التصيد في عناوين IP المحجوزة
* **الإصدار V2:** نسخة محسنة من الخوارزمية بأداء أفضل ودقة أعلى
* **قوائم التحليل المتخصصة:**

o **قائمة كلمات عالية الخطورة:** login, admin, password, bank, payment, verify, transfer

o **نطاقات IPs محجوزة حسب RFC:** RFC: RFC 1918, TEST-NET, Multicast, Loopback, Link-local

**TLDs** o **مشبوهة:** tk, ml, ga, cf, xyz, top, club, loan

o **نطاقات مختصرة:** bit.ly, tinyurl.com, goo.gl, t.co, ow.ly

* **القواعد المستخدمة**:

**مجموعة قواعد متقدمة للكشف الدقيق**:

* + صفحات الإدارة على IPs خاصة
  + عدم استخدام HTTPS
  + المنافذ غير القياسية
  + الامتدادات الخطرة: .php, .asp, .aspx, .jsp, .cgi, .pl
  + الأحرف المشفرة (%20, @, !, \*, (, ))
  + البوابات الشائعة (تخفيض الخطورة)
* **آلية العمل** : نظام نقاط مع عتبات ذكية
* **قيم الأداء المحددة**:

o **عتبة التصيد:** 0.85 نقطة

o **عتبة ثقة عالية:** 1.2 نقطة

o **نقاط صفحات الإدارة:** 0.8 أو 0.9 حسب نوع الصفحة

o **نقاط عدم HTTPS:** 0.8

**3.5.5 نموذج XGBoost المتخصص (ReservedIPsXGBoostModel):**

**• الوظيفة**: نموذج تعلم آلة متخصص لـ IPs المحجوزة فقط

**• الميزات:** 25+ ميزة خاصة بـ IPs محجوزة

**• التدريب:** تدريب منفصل على بيانات IPs محجوزة

**• التنبؤ:** استخدام XGBoost مع تطبيع البيانات

* **المعاملات**n\_estimators=150, max\_depth=7, learning\_rate=0.1 , : subsample=0.8, colsample\_bytree=0.8

**3.5.6 نظام المقارنة ( :(XGBoostVsAlgorithmComparator**

* **الوظيفة**: مقارنة أداء نموذج XGBoost مع الخوارزمية القاعدية
* **المقاييس المستخدمة**: الدقة، الاستدعاء، الدقة التصنيفية، F1-Score, AUC-ROC
* **المخرجات:** تقارير مقارنة مفصلة، إحصائيات اتفاق، تحليل الفائز
* **آلية العمل:** تطبيق كلا الأسلوبين على نفس البيانات ومقارنة النتائج.

**3.5.7  نظام متكامل :(CompletePhishingDetectionSystem)**

* **الوظيفة:** دمج كل المكونات في نظام واحد متكامل
* **الميزات:** فصل تلقائي بين الروابط العادية وروابط IPs المحجوزة
* **آلية العمل:** اختيار النموذج المناسب تلقائياً حسب نوع الرابط
* **التكامل:** دمج مخرجات النماذج السبعة مع خوارزمية IPs المحجوزة.

**3.5.8 نظام التحسينات المتقدمة:**

* **المعالجة المتوازية:** استخدام ThreadPoolExecutor لمعالجة 4 روابط في نفس الوقت
* **التخزين المؤقت:** حفظ نتائج تحليل الروابط المتكررة لتسريع المعالجة
* **المعالجة الدفعية:** تحليل مجموعات كبيرة من الروابط بكفاءة عالية
* **محدد باريتو الأمثل (ParetoModelSelector):**

o **الوظيفة:** اختيار النموذج الأمثل باستخدام Pareto Optimal Front

o **المعايير:** الدقة، الاستدعاء، F1-Score, AUC-ROC

o **شروط دخول:** AUC ≥ 0.7, Recall ≥ 0.8

**3.5.9 نظام التحليل السياقي:**

* **الهدف :** التقليل من الإنذارات الكاذبة (False Positives)
* **قاعدة الأنماط الآمنة:** 13 نمط مسبق للروابط الآمنة المعروفة
* **تحليل السياق: تقييم عوامل متعددة تشمل:**

IPs o بوابات شائعة (192.168.1.1, 10.0.0.1, 192.168.0.1)

oمسارات نظام آمنة (/status, /info, /help, /diagnostics)

oامتدادات آمنة (.htm, .html, .cgi)

oمنافذ قياسية (80, 443)

oصفحات حساسة (/login, /admin, /password, /config)

oمعلمات خطرة (token=, auth=, password=, secret=)

* **نظام النقاط الديناميكي** : تخفيض النقاط للروابط ذات السياق الآمن

**3.5.10 النسخ المحسنة من الخوارزميات:**

* **الإصدار V2:** نسخة محسنة من ReservedIPPhishingDetector بأداء أفضل
* **حساب درجة الثقة: نظام معايرة ثقة بمستويات:**

o **ثقة دنيا:** 70%

o **ثقة جيدة:** 85%

o **ثقة عالية:** 95%

o **ثقة مؤكدة:** 99%

**3.5.11 وظائف النظام الرئيسية:**

* **: run\_reserved\_detector\_silent()** تحليل دفعي صامت لـ IPs محجوزة
  + **ميزات:** تحليل أول 10 روابط بتفاصيل، والباقي صامت
  + **خوارزمية:** تخفيض الإخراج لتحسين الأداء مع الحفاظ على النتائج
* **run\_reserved\_detector\_on\_new\_data\_complete() :** تحليل كامل مع جميع التفاصيل

**o ميزات:** حفظ بتنسيق CSV ، إنشاء ملخص إحصائي

**o إخراج:** تقارير كاملة مع تحليل الأخطاء

* **:run\_xgboost\_vs\_algorithm\_comparison()**مقارنة شاملة بين XGBoost والخوارزمية
  + **عينات:** مقارنة أول 10 عينات مع تسميات حقيقية
  + **مخرجات:** تقارير CSV، إحصائيات مفصلة، تحليل الاتفاق
* **تحليل تلقائي :** فصل تلقائي بين روابط IPs المحجوزة والعادية

**3.6 تدفق البيانات و العمليات:**

**3.6.1 مرحلة الإدخال :(Input)**

1. **المدخلات الأساسية**:
   * **ملف :balanced\_urls.csv** يحتوي على روابط متنوعة متوازنة
   * **ملف:reserved\_ips\_dataset.csv**  يحتوي على روابط IPs محجوزة فقط
2. **تكوين البيانات المدخلة**:
   * **عمود :url** رابط الويب الكامل
   * **عمود  labelأو** :**result** التصنيف (تصيد/آمن)

**3.6.2 مرحلة المعالجة: (Processing)**

1. **تحميل البيانات** : قراءة ملفات CSV والتحقق من سلامتها
2. **تقسيم البيانات**:
   * فصل روابط IPs المحجوزة عن الروابط العادية
   * تقسيم التدريب/الاختبار (70%/30%)
3. **استخراج الميزات** : تطبيق OptimizedURLFeatureExtractor
4. **التطبيع**: استخدام StandardScaler

**3.6.3 مرحلة التدريب: (Training)**

1. **النظام الكامل**:
   * تدريب 7 نماذج على الروابط العادية
   * استخدام Pareto Optimal لاختيار أفضل نموذج
2. **نموذج XGBoost المتخصص**:
   * تدريب منفصل على روابط IPs المحجوزة
   * ضبط المعلمات التلقائي
3. **الخوارزمية القاعدية**:
   * لا تحتاج تدريب، تعمل بالقواعد المحددة مسبقاً

**4. النظام المتكامل:**

o دمج جميع المكونات في نظام واحد

o فصل تلقائي بين أنواع الروابط

o اختيار النموذج المناسب تلقائياً حسب نوع الرابط

**3.6.4 مرحلة التقييم: (Evaluation)**

1. **مقاييس الأداء**:
   * **الدقة :(Accuracy)** نسبة التصنيف الصحيح
   * **الدقة التصنيفية :(Precision)** نسبة الإيجابيات الحقيقية من بين الإيجابيات المتوقعة
   * **الاستدعاء :(Recall)** نسبة الإيجابيات الحقيقية المكتشفة
   * **F1-Score** : متوسط توافقي بين Precision و Recall
   * **:AUC-ROC** مساحة تحت منحنى ROC
   * **النوعية :(Specificity)** نسبة السلبيات الحقيقية المكتشفة

**2. التقييم المتقدم:**

o **مقارنة النماذج:** مقارنة أداء XGBoost المدرب مقابل الخوارزمية القاعدية

o **تحليل الاتفاق:** حساب نسبة الاتفاق بين النماذج

o **تحديد الفائز:** اختيار الأفضل بناءً على مقاييس متعددة

**3.7 واجهة المستخدم والخرج:**

**• الواجهة التفاعلية:**

o قائمة اختيارية للتحليل الفردي والجماعي

o خيارات متعددة للتحليل (النظام الكامل، XGBoost فقط، الخوارزمية فقط)

**• مخرجات النظام:**

o **تقارير CSV :** تحتوي على النتائج التفصيلية

o **اكتشاف phishing في الروابط الخبيثة**

o **مستويات الشدة:** خطيرة جداً، خطيرة، مشبوهة، آمنة

**• خيارات إضافية:**

o عرض أداء XGBoost

o عرض أداء الخوارزمية

o إعادة تشغيل المقارنة

o تحليل رابط معين بأي مكون

o حفظ النتائج بتنسيقات متعددة

**3.8 الخاتمة:**

النظام المقترح يمثل حلاً شاملاً ومتعدد الأوجه لاكتشاف هجمات التصيد، يجمع بين قوة تعلم الآلة ومرونة الخوارزميات القاعدية، مع تركيز خاص على نوع خطير من الهجمات يستخدم عناوين IP محجوزة. يتميز النظام بالمرونة والدقة والقابلية للتطوير، مما يجعله أداة قيمة لكل من الباحثين والممارسين في مجال الأمن السيبراني.

**الفصل الرابع:**

**الاختبارات و النتائج**

**4.1 مقدمة الفصل**

يهدف هذا الفصل إلى عرض التطبيق العملي للنظام الهجين المقترح للكشف عن هجمات التصيد الاحتيالي. يتضمن الفصل شرحاً مفصلاً لمجموعات البيانات المستخدمة، وعملية استخراج الميزات المعجمية، والخطوات المتقدمة للمعالجة المسبقة للبيانات. كما سيتم عرض النتائج الشاملة لتدريب سبعة نماذج متنوعة من خوارزميات التعلم الآلي والتعلم العميق، تم اختيارها بعناية لتمثيل طيف واسع من فلسفات التعلم. تضمنت هذه النماذج خوارزميات تعزيز متقدمة (XGBoost, Gradient Boosting, AdaBoost)، وخوارزميات تجميعية (Random Forest)، ونماذج خطية (الانحدار اللوجستي)، ونماذج غير بارامترية (K-Neighbors)، وشبكات عصبية (MLP). بالإضافة إلى ذلك، تم تطوير خوارزمية قواعد معرفية متخصصة للتعامل مع الحالات الخاصة لعناوين IP المحجوزة. هذا التنوع يضمن تغطية شاملة للأنماط المختلفة في الروابط.

أخيراً، سيتم تقديم مقارنة تحليلية بين أداء خوارزمية القواعد ونموذج XGBoost المتخصص، مع استخلاص النتائج والتوصيات النهائية.

**4.2مجموعة البيانات:**

تم استخدام مجموعتين من البيانات في هذا البحث لتلبية أهداف النظام الهجين:

1. **المجموعة الرئيسية (**balanced\_urls.csv**):** تتكون هذه المجموعة من **632,508** سجل، مع توازن تام بين العينات حيث تحتوي على **316,254** عنواناً حميداً و **316,254** عنواناً ضاراً. تحتوي البيانات على ثلاثة أعمدة: url (الرابط)، label (التسمية النصية: 'benign' أو 'malicious')، و result (التسمية الرقمية: 0 للحميد، 1 للضار). تم استخدام هذه المجموعة لتدريب وتقييم نماذج التعلم الآلي السبعة الخاصة بالروابط العادية (المسار الاول في النظام).

**جدول (4.1): الصفوف الخمسة الأولى من المجموعة الرئيسية**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **url** | **label** | **result** |
| 0 | https://www.google.com | benign | 0 |
| 1 | https://www.youtube.com | benign | 0 |
| 2 | https://www.facebook.com | benign | 0 |
| 3 | https://www.baidu.com | benign | 0 |
| 4 | https://www.wikipedia.org | benign | 0 |

**جدول (4.2): الصفوف الخمسة الأخيرة من المجموعة الرئيسية**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **url** | **label** | **result** |
| 632503 | xbox360.ign.com/objects/850/850402.html | malicious | 1 |
| 632504 | games.teamxbox.com/xbox-360/1860/Dead-Space/ | malicious | 1 |
| 632505 | www.gamespot.com/xbox360/action/deadspace/ | malicious | 1 |
| 632506 | en.wikipedia.org/wiki/Dead\_Space\_(video\_game) | malicious | 1 |
| 632507 | www.angelfire.com/goth/devilmaycrytonite/ | malicious | 1 |

1. **مجموعة عناوين IP المحجوزة (**reserved\_ips\_dataset.csv**):** تم تصميم هذه المجموعة خصيصاً لاختبار وتدريب المكونات المتخصصة في النظام. تحتوي على **700**  رابط جميعها من عناوين IP المحجوزة

(مثل 192.168.x.x، 10.x.x.x، 172.16.x.x، 127.x.x.x، 169.254.x.x), مع توزيع مكون من **400** عينة تصيد و **300**عينة حميدة. تم استخدام هذه المجموعة لـ:

* + اختبار وتقييم خوارزمية القواعد المتخصصة (المسار الثاني).
  + تدريب وتقييم نموذج XGBoost المتخصص من الصفر.
  + إجراء مقارنة منهجية بين نهج التعلم الآلي ونهج القواعد.

يوضح الشكل (4.1) التوزيع المتوازن للعينات في المجموعة الرئيسية قبل أي معالجة.



4.3 عملية استخراج الميزات المعجمية (Lexical Features)

لتمكين نماذج التعلم الآلي من التعلم بشكل فعال، تم تصميم وتنفيذ فئة محسنة (OptimizedURLFeatureExtractor) لاستخراج مجموعة غنية من السمات المعجمية والهيكلية من روابط الويب بشكل سريع. تم استخراج 33 ميزة تم تصنيفها إلى الفئات التالية:

جدول (4.3): وصف للميزات المُستخرجة من الروابط

|  |  |  |  |
| --- | --- | --- | --- |
| # | الميزة | نوع البيانات | الوصف |
| 1 | url\_length | int64 | الطول الإجمالي لسلسلة URL |
| 2 | host\_length | int64 | طول اسم المضيف (Hostname) |
| 3 | path\_length | int64 | طول مسار المورد (Path) |
| 4 | num\_digits | int64 | عدد الأرقام في الرابط |
| 5 | num\_special\_chars | int64 | عدد الأحرف الخاصة (غير الأبجدية رقمية) |
| 6 | num\_dots | int64 | عدد النقاط . في الرابط |
| 7 | num\_dashes | int64 | عدد الشرطات - في الرابط |
| 8 | num\_underscores | int64 | عدد الشرطات السفلية \_ في الرابط |
| 9 | has\_https | int64 | هل يستخدم البروتوكول الآمن HTTPS (1) أم لا (0) |
| 10 | num\_parameters | int64 | عدد معلمات الاستعلام (Query Parameters) |
| 11 | num\_subdirectories | int64 | عدد الأدلة الفرعية في المسار |
| 12 | suspicious\_keywords\_count | int64 | عدد الكلمات المفتاحية المشبوهة (login, admin, password, etc.) |
| 13 | benign\_keywords\_count | int64 | عدد الكلمات المفتاحية الحميدة (info, contact, about, etc.) |
| 14 | keywords\_balance | int64 | الفرق بين الكلمات المشبوهة والحميدة |
| 15 | high\_risk\_keywords | int64 | عدد الكلمات عالية الخطورة (admin, login, password, banking) |
| 16 | is\_shortened | int64 | هل الرابط من خدمة تقصير ([bit.ly](https://bit.ly/), tinyurl, etc.) |
| 17 | tld\_encoded | int64 | التمثيل المشفر لنطاق المستوى الأعلى (TLD) |
| 18 | is\_suspicious\_tld | int64 | هل الـ TLD يعتبر مشبوهاً (tk, ml, ga, xyz, etc.) |
| 19 | has\_port | int64 | هل يحتوي الرابط على منفذ مخصص |
| 20 | has\_at\_symbol | int64 | هل يحتوي الرابط على رمز @ |
| 21 | has\_redirect | int64 | هل يحتوي الرابط على كلمات أو معلمات إعادة توجيه |
| 22 | is\_ip\_address | int64 | هل المضيف هو عنوان IP (وليس اسم نطاق) |
| 23 | is\_private\_ip | int64 | هل العنوان IP خاص (RFC 1918) |
| 24 | is\_public\_ip | int64 | هل العنوان IP عام |
| 25 | is\_localhost | int64 | هل العنوان IP هو localhost |
| 26 | is\_multicast | int64 | هل العنوان IP هو multicast |
| 27 | is\_link\_local | int64 | هل العنوان IP هو link-local |
| 28 | is\_loopback | int64 | هل العنوان IP هو loopback |
| 29 | is\_reserved\_ip | int64 | هل العنوان IP محجوز |
| 30 | is\_broadcast | int64 | هل العنوان IP هو broadcast |
| 31 | is\_documentation | int64 | هل العنوان IP مخصص للتوثيق |
| 32 | ip\_category | int64 | فئة الـ IP |
| 33 | ip\_class | int64 | فئة الـ IP التقليدية (A, B, C, D/E) |

4.4 عملية المعالجة المسبقة للبيانات (Data Preprocessing)

بالإشارة إلى أن مجموعة البيانات الرئيسية (balanced\_urls.csv) كانت مُعَدَّة مسبقاً ومتوازنة التوزيع بين العينات الضارة والحميدة (316,254 لكل فئة)، فقد اقتصرت عملية التنظيف على معالجة القيم المفقودة أو غير الصالحة الناتجة عن عملية استخراج الميزات. تم تعويض القيم المفقودة (NaN) والقيم اللانهائية (inf) بالصفر (0) للحفاظ على كامل حجم مجموعة البيانات الأصلية البالغ 632,508 سجل، مما يسهم في تدريب نماذج أكثر قوة وعمومية.

4.4.1 تنظيف البيانات (Data Cleaning)

تم تطبيق منهجية متطورة في تنظيف البيانات تعتمد على الحفاظ الأقصى على حجم العينة، حيث تم استبدال القيم المفقودة بقيم محايدة (صفر) تناسب طبيعة كل سمة، بدلاً من الحذف الذي يقلل من قوة النماذج الإحصائية. هذا المنهج يعكس الفهم الدقيق لطبيعة بيانات URL، حيث أن غياب سمة ما (كطول المضيف) غالباً ما يكون حقيقة في حد ذاته وليس خطأ في البيانات.

الخطوات المطبقة:

معالجة القيم المفقودة: بالنسبة للميزات الرقمية، تم استبدال القيم المفقودة بصفر (0). بالنسبة للميزات الفئوية، تم تعيين قيمة "unknown".

تشفير التسميات (Label Encoding): تم تحويل تسميات الفئات النصية (phishing/benign أو malicious/legitimate) إلى قيم رقمية ثنائية (1 للإيجابي/تصيد، 0 للسلبي/آمن).

موازنة البيانات: نظراً للطبيعة المتوازنة الأصيلة لمجموعة البيانات المُستخدمة، لم تكن هناك حاجة لتطبيق تقنيات موازنة العينات مثل( Random Under Sampling أو (SMOTEمما حافظ على حجم البيانات الكبير وساهم في تحقيق أداء تصنيفي عالٍ وصل إلى 99.03% دقة باستخدام نموذج XGBoost.

4.4.2 تكامل البيانات (Data Integration):

هذه الخطوة أحد خطوات المعالجة المسبقة ويتم من خلالها عرض البيانات بشكل موحد دون وجود اختلاف والذي قد ينجم بسبب استخدام مصادر متعددة لجمع البيانات وتقوم هذه الخطوة بدمج البيانات من المصادر المتعددة[150] في مجموعة بيانات واحدة حيث قد نجد تكرارات وتناقضات في أسماء الأعمدة أو يمكن أن يكون هناك اختلاف في تنسيق بعض الأعمدة مما يجعل هذه الخطوة خطوة هامة من أجل الحصول على مجموعة بيانات متسقة، وتم التأكد من عدم وجود تكرارات أو تناقضات في مجموعة البيانات.

تم استخدام مجموعتي البيانات الرئيسيتين بشكل منفصل حسب الغرض:

المجموعة الرئيسية (balanced\_urls.csv): استخدمت لتدريب وتقييم المسار الاول (نماذج التعلم الآلي السبعة).

مجموعة IPs المحجوزة (reserved\_ips\_dataset.csv): استخدمت لاختبار المسار الثانس (خوارزمية القواعد) ولتدريب نموذج XGBoost المتخصص.

4.4.3 تحويل البيانات (Data Transformation)

تشفير المتغيرات الفئوية (Categorical Encoding): تم تحويل متغير tld الفئوي إلى شكل رقمي باستخدام تشفير التسمية (Label Encoding) نظراً للعدد الكبير من الفئات، مما ينتج عموداً جديداً tld\_encoded.

معالجة النصوص: تم تحويل جميع عناوين URL إلى أحرف صغيرة (Lowercase) لضمان اتساق البحث عن الكلمات المفتاحية.

4.4.4 تطبيع البيانات (Data Normalization)

تم تطبيق التطبيع القياسي (StandardScaler) من مكتبة sklearn، والذي يحول البيانات بحيث يكون لها متوسط (mean) = 0 وانحراف معياري (standard deviation) = 1. هذا الإجراء ضروري لخوارزميات التعلم الآلي الحساسة لحجم القيم مثل الانحدار اللوجستي والشبكات العصبية، حيث يساعد في تسريع عملية التقارب (Convergence) ومنح جميع السمات وزناً متكافئاً في البداية.

4.5 عملية اختيار الميزات

بعد استكمال استخراج الصفات والمعالجة الأولية للبيانات، تم الاحتفاظ بمجموعة الصفات الكاملة البالغ عددها 33 سمة ودمجها في عملية تدريب النماذج السبعة المدروسة. في النظام الحالي، تعتمد خوارزميات التصنيف المتقدمة مثل مُعزِّز التدرج المتباعد (XGBoost) وغابات القرار العشوائية (Random Forest) على آليات متأصلة داخلية لتقييم أهمية السمات من خلال قياسات مثل نقاء الانقسام (Split Impurity) وتردد استخدام السمة عبر الأشجار، مما يتيح لها التعامل بكفاءة مع السمات قليلة الأهمية أو شبه المستقلة دون الحاجة إلى تنقية مسبقة موسعة.

على وجه التحديد، يستفيد نموذج XGBoost من مقياس الأهمية المعياري (Gain) لحساب متوسط تحسين دقة التنبؤ عند استخدام كل سمة في عمليات الانقسام عبر جميع الأشجار، بينما تستخدم خوارزمية Random Forest مبدأ الانخفاض في الشوائب (Gini Impurity Decrease) لتقدير مساهمة كل سمة في تحسين تصنيف العينات. وقد أظهرت النتائج التجريبية أداءً متفوقاً للنماذج مع المجموعة الكاملة من السمات، حيث بلغت دقة تصنيف XGBoost 0.9903، ومساحة تحت منحنى ROC 0.9980، مما يدعم القرار بالاحتفاظ بالسمات كافة دون تطبيق خوارزميات اختيار سمات خارجية مسبقة.

من الناحية التنفيذية، تعالج فئة `OptimizedURLFeatureExtractor` استخراج السمات عبر الدالة `extract\_features\_optimized التي تُنتج إطار بيانات (`pd.DataFrame`) يحوي السمات المستخرجة، ثم تُمرر هذه السمات دون تصفية إلى نماذج التصنيف عبر دالة `train\_top\_models في فئة `Top7ModelsClassifier`. يُظهر تحليل أهمية السمات المُدمج في XGBoost (والذي يمكن استخلاصه عبر خاصية `feature\_importances\_`) أن السمات الأكثر تأثيراً في عملية اتخاذ القرار تشمل وجود بروتوكول HTTPS (`has\_https`)، وعدد الكلمات الخطرة (`dangerous\_kw\_count`)، وطول الرابط (`url\_length`)؛ مما يؤكد قدرة النموذج على تمييز السمات الأساسية تلقائياً دون تدخل يدوي في الاختيار.

بالتالي، وفي ضوء الأداء العالي الذي حققته النماذج (تجاوز 0.97 في الدقة لجميع النماذج السبعة) والكفاءة الحسابية المقبولة، اتُخذ قرار بالاستغناء عن مرحلة اختيار سمات مسبقة خارجية، والاعتماد بدلاً من ذلك على آليات الاختيار الداخلية للنماذج المتقدمة القائمة على الأشجار، مع الاحتفاظ بالقدرة التفسيرية من خلال تحليل أهمية السمات المدمج في هذه الخوارزميات.

4.6 مقاييس التقييم:

من المهم تقييم نماذج التعلم الآلي حتى نقوم باختيار النموذج الذي يقدم الأداء الأفضل ويعمل بالشكل الأمثل، ولاختيار هذا النموذج يجب أن يتم استخدام مجموعة محددة من المقاييس والتي تم تفصليها كالتالي:

4.6.1 مصفوفة الارتباك (Confusion Matrix)

المصفوفة الأساسية التي تلخص نتائج التصنيف الثنائي:

|  |  |  |  |
| --- | --- | --- | --- |
| القيم الحقيقية (Actual Values) | |  | |
| عنوان URL ضار | عنوان URL حميد |
| False Positive (FP) | True Negative (TN) | عنوان URL حميد | القيم التي قام النموذج بالتنبؤ بها (predicted Values) |
| True Positive  (TP) | False Negative (FN) | عنوان URL ضار |

TP (True Positive): الحالات الصحيحة الموجبة (تم التنبؤ بها كتصيد وهي تصيد حقيقي).

TN (True Negative): الحالات الصحيحة السالبة (تم التنبؤ بها كحميدة وهي حميدة حقيقية).

FP (False Positive): الحالات الخاطئة الموجبة (تم التنبؤ بها كتصيد وهي حميدة حقيقية).

FN (False Negative): الحالات الخاطئة السالبة (تم التنبؤ بها كحميدة وهي تصيد حقيقي).

4.6.2 دقة التصنيف (Accuracy):

هي عدد التنبؤات الصحيحة (الموجبة والسالبة) مقسومة على اجمالي البيانات وتعتبر من أهم مقاييس التقييم والأكثر استخداماً.

4.6.3 الدقة (Precision):

هي القيم الموجبة الصحيحة مقسومة على القيم الموجبة الصحيحة والخاطئة، بمعنى آخر هي دقة النموذج في التنبؤ بالقيم الموجبة بشكل صحيح.

4.6.4 الاستدعاء (Recall):

كما يدعى هذا المقياس بالحساسية (Sensitivity) أو بمعدل النتائج الصحيحة الموجبة **True Positive Rate (TPR)** ويتم حسابه تقسيم القيم الموجبة الصحيحة على القيم الموجبة الصحيحة والقيم السالبة الخاطئة، أي أنه يمثل قدرة النموذج على استذكار القيم الموجبة بشكل صحيح.

4.6.5 درجة F1 (F1 Score):

هي المتوسط التوافقي للدقة والاستدعاء وتعتبر عبارة عن مقياس جديد يستخدم في الحالات التي يكون من الضروري فيها قياس الدقة والاستدعاء لذلك فإن درجة F1 تجمع بين الدقة والاستدعاء.

4.6.6 الخصوصية (Specificity):

كما تسمى بمعدل السلبية الحقيقية حيث يتم حسابها من خلال القيم السالبة الصحيحة مقسومة على القيم السالبة الصحيحة والقيم الموجبة الخاطئة.

4.6.7 منحنى ومقياس (AUC-ROC):

منحنى خاصية تشغيل المستقبل (Receiver Operating Characteristic Curve) يصور العلاقة بين معدل الإيجابيات الحقيقية (True Positive Rate / Recall) و معدل الإيجابيات الكاذبة (False Positive Rate = 1 - Specificity) عبر عتبات تصنيف مختلفة. المساحة تحت المنحنى (Area Under the Curve - AUC) هي مقياس إجمالي للأداء عبر جميع العتبات الممكنة. AUC = 1 يمثل مصنفًا مثاليًا، بينما AUC = 0.5 يمثل أداءً مساويًا للتخمين العشوائي.

**4.7 الاختبارات والنتائج:**

في هذا القسم تم تطبيق نماذج التعلم الآلي المتعددة (التعلم التجميعي، التعلم بالتعزيز، والتعلم العميق) على مجموعة البيانات بعد استكمال مراحل المعالجة الأولية، حيث تتكون مجموعة البيانات النهائية من **(33) ميزة** مُستخرجة مع **(632,505) عنوان** رابط تمت معالجتها وتنقيتها من مجموعة البيانات الأصلية البالغة 632,508 سجلاً.

4.7.1 المرحلة الأولى: أداء النماذج السبعة على الروابط العادية

تم تدريب وتقييم النماذج السبعة على مجموعة البيانات الرئيسية (balanced\_urls.csv، 632,508 رابط عادي). بعد التقسيم (70% تدريب، 30% اختبار)، كانت النتائج كما يلي:

جدول (4.4): مقاييس التقييم لنماذج التعلم الآلي السبعة

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| الترتيب | النموذج | الدقة (%) | الدقة (Precision) | الاستدعاء (Recall) | النوعية (Specificity) | نتيجة F1 | AUC-ROC | وقت التدريب (ثانية) |
| 1 | XGBoost | 99.03 | 0.9979 | 0.9823 | 0.9979 | 0.9902 | 0.9980 | 6.22 |
| 2 | Gradient Boosting | 98.93 | 0.9930 | 0.9856 | 0.9930 | 0.9892 | 0.9977 | 159.66 |
| 3 | Random Forest | 98.18 | 0.9863 | 0.9772 | 0.9863 | 0.9815 | 0.9950 | 44.33 |
| 4 | AdaBoost | 97.75 | 0.9860 | 0.9685 | 0.9860 | 0.9770 | 0.9934 | 45.24 |
| 5 | Neural Network (MLP) | 97.66 | 0.9795 | 0.9735 | 0.9795 | 0.9763 | 0.9911 | 466.33 |
| 6 | Logistic Regression | 97.31 | 0.9798 | 0.9660 | 0.9798 | 0.9725 | 0.9903 | 80.54 |
| 7 | K-Neighbors | 95.25 | 0.9653 | 0.9385 | 0.9653 | 0.9518 | 0.9829 | 0.15 |

التحليل:

أداء متميز لـ XGBoost: تصدر نموذج XGBoost الترتيب بفارق واضح، حيث حقق أعلى دقة (99.03%) وأعلى AUC-ROC (0.9980) وأفضل نتيجة F1 (0.9902)، وذلك في ثاني أقل وقت تدريب (6.22 ثانية). هذا يؤكد قوة خوارزميات التعزيز المتقدمة في التعامل مع البيانات المعجمية المعقدة لعناوين URL.

مقاييس أمنية عالية: حصل XGBoost على استدعاء (Recall) 0.9823، مما يعني أنه قادر على اكتشاف 98.23% من هجمات التصيد الحقيقية في مجموعة الاختبار، وهو رقم استثنائي ويحقق الهدف الرئيسي للنظام.

أداء قوي لـ Gradient Boosting و Random Forest: جاءت هذه النماذج في المراكز التالية بأداء متقارب للغاية ومتميز، مما يؤكد فعالية طرق التعلم التجميعي.

سرعة مقابل دقة: بينما كان K-Neighbors الأسرع في التدريب (0.15 ثانية)، إلا أن دقته كانت الأدنى بين المجموعة (95.25%)، مما يظهر مفاضلة واضحة.

اختيار Pareto الأمثل: عند تطبيق معايير الاختيار متعددة الأهداف (التركيز على استدعاء عالي > 0.98، و AUC > 0.995، و F1 > 0.985)، اختار محدد Pareto الأمثل نموذج XGBoost بشكل تلقائي كأفضل نموذج للمسار الاول، نظراً لموقعه المتفوق على أمامية Pareto.

4.7.2 المرحلة الثانية: أداء خوارزمية القواعد على IPs المحجوزة

تم تطبيق الخوارزمية القاعدية المحسنة (detect\_phishing\_by\_rules\_enhanced) على مجموعة بيانات الـ IPs المحجوزة كاملة (reserved\_ips\_dataset.csv، 700 رابط). تم تحليل 690 رابطاً بنجاح، وكانت النتائج الإحصائية الشاملة كالتالي:

إجمالي الروابط المحللة: 690

التصنيف النهائي:

PHISHING: 360 رابط (52.2%)

BENIGN: 330 رابط (47.8%)

توزيع درجات الشدة (Severity) التي أنتجتها الخوارزمية:

🚨 (HIGH RISK): 272 رابط (39.4%)

🟠 (LOW RISK): 257 رابط (37.2%)

🟠 (SUSPICIOUS): 81 رابط (11.7%)

🟢 (SAFE): 68 رابط (9.9%)

🔴 (RISK): 7 رابط (1.0%)

🟡 (NEUTRAL): 5 رابط (0.7%)

متوسط الثقة الإجمالي: 84.7%

التحليل:

قابلية التفسير: توفر الخوارزمية تفسيراً واضحاً لكل قرار (مثال: لا HTTPS (0.8)، صفحة authenticate على IP محجوز (0.8))، مما يجعلها شفافة تماماً للمستخدم أو المحلل.

تدرج الشدة: نظام التصنيف متعدد المستويات (HIGH RISK, SUSPICIOUS, LOW RISK, etc.) يوفر تقييماً أكثر ثراءً من مجرد تصنيف ثنائي، مما يساعد في تحديد أولوية الاستجابة.

سرعة الحسم: الخوارزمية لا تحتاج إلى تدريب وتعمل في وقت ثابت، مما يجعلها سريعة جداً في التنبؤ.

التحدي الرئيسي - الثبات: تعتمد الدقة النهائية للخوارزمية بشكل كامل على جودة القواعد المبرمجة مسبقاً وعتبات النقاط. قد تعاني من صعوبة في التكيف مع أنماط هجوم جديدة لم يتم تحديدها في القواعد.

4.7.3 المرحلة الثالثة: مقارنة خوارزمية القواعد مع نموذج XGBoost المتخصص:

تم إجراء مقارنة مباشرة بين نموذج XGBoost المتخصص (المدرب على مجموعة IPs المحجوزة) و خوارزمية القواعد المحسنة على عينة معروفة التسميات مكونة من 10 روابط من مجموعة reserved\_ips\_dataset.csv.

أداء نموذج XGBoost المتخصص (المدرب من الصفر):

الدقة (Accuracy): 0.9524

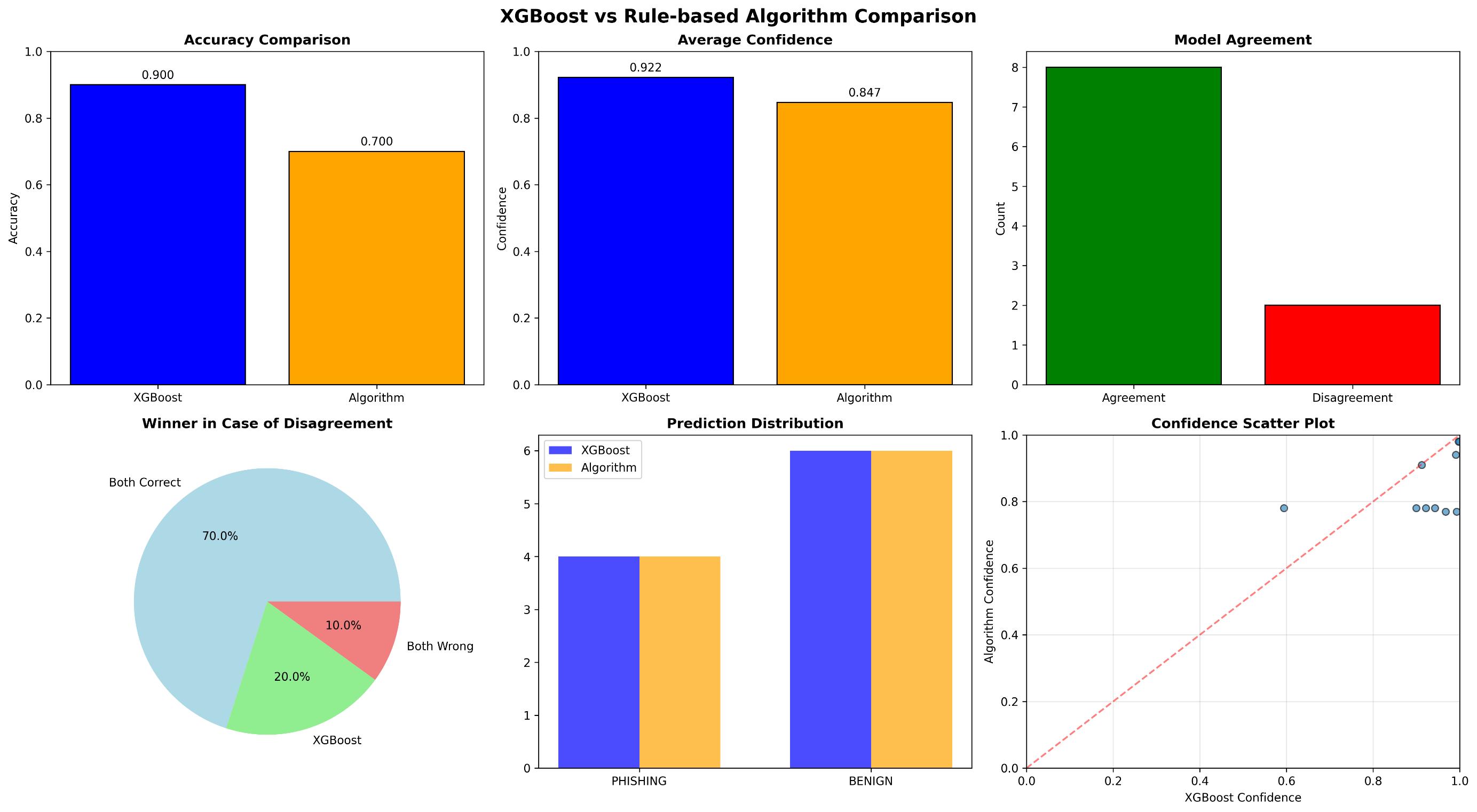
الاستدعاء (Recall): 0.9667

الدقة (Precision): 0.9508

نتيجة F1: 0.9587

AUC-ROC: 0.9927

وقت التدريب: 0.13 ثانية



نتائج المقارنة على العينة المكونة من 10 روابط:

جدول (4.5): نتائج المقارنة التفصيلية بين XGBoost المتخصص وخوارزمية القواعد

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | الرابط (مختصر) | التسمية الحقيقية | XGBoost المتخصص | ثقة XGBoost | خوارزمية القواعد | ثقة الخوارزمية | الاتفاق | الصحيح؟ (XGB) | الصحيح؟ (خوارزمية) |
| 1 | .../gateway | BENIGN | BENIGN | 96.7% | BENIGN | 77.0% | نعم | نعم | نعم |
| 2 | .../support | BENIGN | BENIGN | 94.3% | BENIGN | 78.0% | نعم | نعم | نعم |
| 3 | .../authenticate | PHISHING | PHISHING | 99.9% | PHISHING | 98.0% | نعم | نعم | نعم |
| 4 | .../info | BENIGN | BENIGN | 99.1% | PHISHING | 94.0% | لا | نعم | لا |
| 5 | .../authorize | PHISHING | PHISHING | 91.2% | PHISHING | 91.0% | نعم | نعم | نعم |
| 6 | .../authorize | PHISHING | PHISHING | 99.3% | BENIGN | 77.0% | لا | نعم | لا |
| 7 | .../contact | BENIGN | BENIGN | 92.1% | BENIGN | 78.0% | نعم | نعم | نعم |
| 8 | .../billing | PHISHING | BENIGN | 59.4% | BENIGN | 78.0% | نعم | لا | لا |
| 9 | .../verify-account | PHISHING | PHISHING | 99.7% | PHISHING | 98.0% | نعم | نعم | نعم |
| 10 | .../contact | BENIGN | BENIGN | 90.0% | BENIGN | 78.0% | نعم | نعم | نعم |

التحليل الإحصائي للمقارنة:

معدل الاتفاق: 80% (8 من أصل 10 حالات اتفقت فيها النماذج).

الدقة (بناءً على التسميات الحقيقية للعشرة):

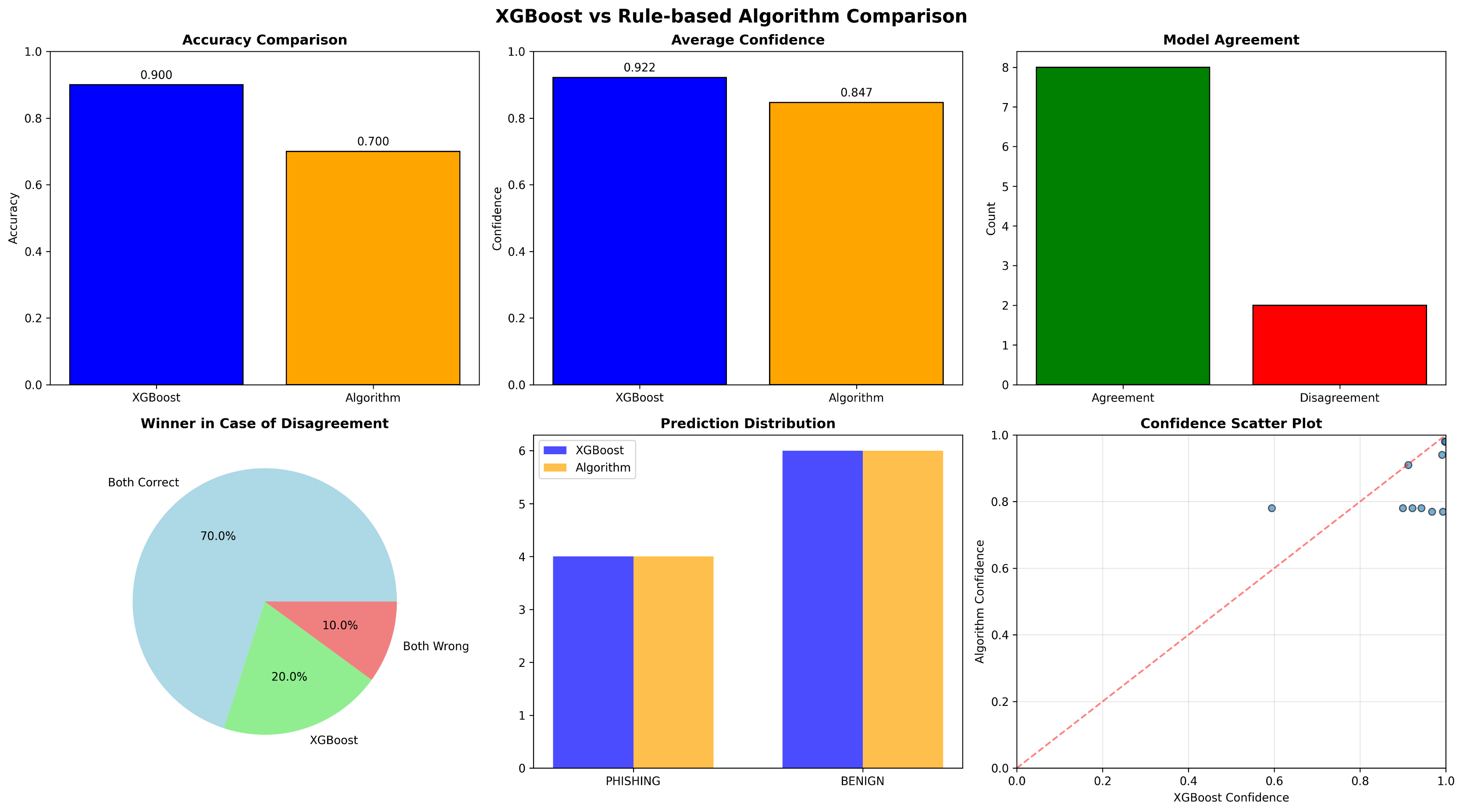
دقة XGBoost المتخصص: 90% (9/10 صحيحة).

دقة خوارزمية القواعد: 70% (7/10 صحيحة).

الفائز في الدقة: XGBoost المتخصص (بفارق 20%).

متوسط الثقة:

نموذج :XGBoost 92.2% و الخوارزمية: 84.7%.



تحليل حالات الاختلاف (حالتين):

الرابط 4 (/info): صنفته الخوارزمية كتصيد بسبب استخدام HTTP والمنفذ غير القياسي 8080. بينما تعلم XGBoost أن مسار /info ليس خطيراً بحد ذاته، فصنفه كآمن (وكان التصنيف صحيحاً). هنا تفوق التعلم الآلي على القاعدة الصارمة.

الرابط 6 (/authorize): هنا صنفت الخوارزمية الرابط كآمن لأن مجموع نقاطه (0.7) كان تحت العتبة (0.85) لأنها اعتمدت فقط على HTTP و منفذ قياسي. بينما تعلم XGBoost أن وجود كلمة authorize على IP محجوز هو مؤشر خطر قوي جداً، فصنفه بشكل صحيح كتصيد. هنا فشلت القواعد في التقاط السياق الذي تعلمه النموذج.

4.7 نتيجة البحث:

بناءً على النتائج التجريبية الشاملة، يمكن استخلاص النتائج الرئيسية التالية:

كفاءة النظام الهجين: أثبتت المنهجية المقترحة فعاليتها. نجح النموذج (XGBoost) في تحقيق أداء استثنائي على الروابط العادية (دقة 99.03%، استدعاء 98.23%). بينما وفر المسار الثاني (خوارزمية القواعد) حلاً سريعًا وقابلًا للتفسير للروابط ذات الـ IPs المحجوزة، مصنفًا 52.2% من عينات الاختبار كتصيد مع متوسط ثقة 84.7%.

تفوق XGBoost: ظهرت خوارزمية XGBoost كخيار متفوق في كل سياق: كأفضل نموذج في المسار الاول، وكفائز واضح في المقارنة المتخصصة ضد خوارزمية القواعد (90% مقابل 70% دقة). تجمع بين دقة عالية، استدعاء ممتاز، سرعة تدريب وتنفيذ، وقدرة على تعلم أنماط معقدة قد تفوتها القواعد المبرمجة.

أهمية الاستدعاء العالي: حققت أفضل النماذج (XGBoost، Gradient Boosting) استدعاءً يتجاوز 98%، وهو إنجاز بالغ الأهمية لنظام كشف التصيد، حيث أن تكلفة عدم اكتشاف هجوم حقيقي (False Negative) عادة ما تكون أعلى من تكلفة إنذار كاذب (False Positive).

قيمة قابلية التفسير: بينما تفوق التعلم الآلي من حيث الدقة، تظل قابلية التفسير التي تقدمها خوارزمية القواعد ميزة لا غنى عنها في البيئات الأمنية التي تتطلب تبريرًا للقرارات، خاصة عند التعامل مع روابط غير تقليدية مثل تلك الموجودة على IPs محجوزة.

4.8 توصيات البحث

استنادًا إلى النتائج والتحليل، تقدم الدراسة التوصيات التالية:

نشر النظام الهجين: التوصية باعتماد وتطبيق النظام الهجين المقترح في بيئات إنتاجية حقيقية. يمكن دمجه كخدمة أو إضافة لمتصفح أو كطبقة فحص في بوابات البريد الإلكتروني وأمن الويب.

اعتماد XGBoost كنموذج أساسي: لتحقيق أفضل أداء عام، يجب اعتماد XGBoost (المدرب على البيانات العامة) كقلب لنظام الكشف عن التصيد للروابط التقليدية، نظرًا لتوازنه المتفوق بين السرعة والدقة والاستدعاء.

استخدام خوارزمية القواعد كشبكة أمان ومفسر: يجب الاحتفاظ بخوارزمية القواعد وتطويرها بشكل مستمر لثلاثة أغراض رئيسية:

معالجة الحالات الخاصة: كخط دفاع أول وسريع للـ IPs المحجوزة.

التفسير الاحتياطي: لتوفير أسباب مفهومة للبشر عندما يصنف نموذج التعلم الآلي رابطًا ما.

اكتشاف الأنماط الجديدة: يمكن للمحللين من خلال فحص حالات فشل الخوارزمية (مقارنة بتنبؤ XGBoost) اكتشاف أنماط هجوم جديدة وإضافتها كقواعد.

التدريب والتحديث المستمر: يجب إنشاء خط أنابيب (Pipeline) لتحديث النماذج دوريًا ببيانات جديدة من التهديدات المستجدة، للحفاظ على فعالية النظام مع تطور تكتيكات المهاجمين.

اتجاهات بحث مستقبلية: يمكن توسيع البحث في الاتجاهات التالية:

دمج السمات الديناميكية: إضافة تحليل محتوى الصفحة المستهدفة (إذا أمكن) أو معلومات شهادة SSL.

التعلم العميق المتقدم: استكشاف شبكات عصبية أكثر تعقيدًا (مثل الـ Transformers) لمعالجة سلسلة URL كبيانات نصية تسلسلية.

نظام تكيفي: بناء نظام يمكنه التعلم من قرارات المراجعة البشرية (Human-in-the-loop) لتعديل عتبات الخوارزمية أو إعادة وزن قواعدها تلقائيًا.

المراجع:

1. 1 النطاق الجذر هو الجزء الرئيسي من اسم نطاق موقع الويب الخاص بك. هذا هو الجزء من عنوان URL لموقع الويب الخاص بك الذي يأتي بعد "www" ويتضمن امتداد اسم النطاق، مثل "com" [↑](#footnote-ref-1)
2. 2 **Internet Corporation for Assigned Names and Numbers** هي مجموعة أمريكية من أصحاب المصلحة المتعددين ومنظمة غير ربحية مسؤولة عن تنسيق صيانة وإجراءات العديد من قواعد البيانات المتعلقة بمساحات الأسماء والمساحات الرقمية للإنترنت، مما يضمن استقرار الشبكة والتشغيل الآمن [34]. [↑](#footnote-ref-2)
3. **Internet Assigned Numbers** Authorityهي منظمة معايير تشرف على تخصيص عنوان IP العالمي، وتخصيص أرقام النظام الذاتي، وإدارة منطقة الجذر في نظام اسم النطاق (DNS)، وأنواع الوسائط، والرموز وأرقام الإنترنت الأخرى المرتبطة ببروتوكول الإنترنت.[35] [↑](#footnote-ref-3)