الجمهورية اليمنية

وزارة التربية والتعليم والبحث العلمي

جامعة آزال للتنمية البشرية

كلية الهندسة وتكنولوجيا المعلومات

جامعة آزال للتنمية البشرية
Azal University for Human Development

Republic of Yemen

Ministry of Higher Education

Azal University for Human Development

Faculty of Engineering &IT

نظام ذكي لكشف روابط التصيد الاحتيالي باستخدام تقنيات التعلم العميق

Smart System for Detecting Phishing URLs Using Deep Learning Techniques

مهند عبدالجليل الشميري	2022010882
ايمن عادل مجلي	2022010838
احمد ياسر العامري	2022010979
بشار احمد طشان	2022011045
محمد عادل طه	2022010798
احمد شاجع الجوفي	2022010994

إشراف الدكتـــــــــــــــور:

عبدالســـــلام خاقــو

البحث مقدم الى كلية الهندسة وتكنولوجيا المعلومات – جامعة آزال للتنمية البشرية لاستكمال متطلبات الحصول على درجة البكالوريوس في الأمن السيبراني

2025 - 2026م

الملخص

يهدف هذا البحث إلى تطوير نموذج ذكاء اصطناعي متقدم قادر على كشف هجمات التصيد الاحتيالي عبر الروابط(URL Phishing Attacks) ، وذلك في ظل التزايد المستمر لهذه الهجمات واعتمادها على أساليب متطورة يصعب اكتشافها باستخدام الأنظمة التقليدية. وتُعد هجمات التصيد من أخطر التهديدات في مجال الأمن السيبراني، حيث تستهدف المستخدمين من خلال روابط مزيفة تحاكي المواقع الحقيقية بهدف سرقة البيانات الحساسة مثل كلمات المرور والمعلومات المالية.

تنطلق فكرة المشروع من الحاجة إلى إيجاد حل ذكي وفعّال يعتمد على تقنيات التعلم العميق (Deep Learning) لتحليل بنية الروابط النصية واكتشاف الأنماط الاحتيالية الكامنة فيها، دون الحاجة إلى فحص محتوى الصفحة المرتبطة بالرابط. ويعتمد النموذج المقترح على بنية هجينة تجمع بين الشبكات العصبية الالتفافية (CNN) والشبكات العصبية المتكررة (LSTM)، حيث تساهم طبقات (CNN) في استخراج الأنماط المحلية داخل الرابط، بينما تعمل طبقات (LSTM) على تحليل العلاقات التسلسلية بين مكونات الرابط، مما يعزز من قدرة النموذج على التمييز بين الروابط الآمنة والروابط الاحتيالية بدقة عالية.

تم في هذا البحث تطوير نموذج هجين متقدم قائم على تقنيات التعلم العميق، يجمع بين الشبكات العصبية الالتفافية (CNN)والشبكات العصبية المتكررة(LSTM) ، بهدف تحسين كفاءة كشف هجمات التصيد الاحتيالي عبر الروابط الإلكترونية. ويعتمد النموذج على تحليل البنية النصية للرابط واستخراج الأنماط الاحتيالية تلقائيا دون الحاجة إلى الاعتماد على الخصائص اليدوية التقليدية.

وقد أظهرت النتائج التجريبية أن النموذج المقترح حقق دقة بلغت 98.46%، مع أداء متفوق مقارنة بعدد من الأساليب التقليدية والدراسات السابقة، مما يؤكد فعاليته العالية واستقراره في بيئات العمل الواقعية.

كما يركز هذا البحث على معالجة عدد من التحديات المرتبطة ببيانات الروابط، مثل عدم التوازن بين الفئات، واختلاف أطوال الروابط، وتنوع الأنماط المستخدمة في الهجمات الحديثة. ولتحقيق ذلك، تم استخدام مجموعة بيانات كبيرة ومتنوعة، مع تطبيق تقنيات معالجة مسبقة متقدمة، مثل تحويل الروابط إلى تمثيل عددي على مستوى الأحرف (Character-Level Representation)، مما يسمح للنموذج بفهم أدق لتركيب الرابط.

وقد أظهرت النتائج التجريبية أن النموذج المقترح حقق أداءً متميزًا من حيث الدقة والاستدعاء، مما يعكس كفاءته في كشف روابط التصيد وتقليل الأخطاء، وبالتالي يساهم في تعزيز حماية المستخدمين من التهديدات الإلكترونية. كما تم اختبار النموذج على بيانات جديدة غير مستخدمة في التدريب، وأثبت قدرته على التعميم والعمل بكفاءة في بيئات واقعية.

يساهم هذا المشروع في تقديم إضافة مهمة في مجال الأمن السيبراني، حيث يوفر أداة ذكية تساعد في تقليل احتمالية الوقوع ضحية لهجمات التصيد الاحتيالي، كما يدعم المؤسسات والأفراد بحلول عملية للكشف المبكر عن الروابط المشبوهة. بالإضافة إلى ذلك، يقدم البحث نموذجًا أوليًا يمكن تطويره مستقبلًا ليصبح نظامًا متكاملًا يُستخدم في التطبيقات الواقعية أو يتم دمجه مع المتصفحات وأنظمة الحماية المختلفة.

ويغطي هذا البحث مجموعة من المحاور الأساسية، تشمل تحليل المشكلة، واستعراض الدراسات السابقة، وبناء النموذج المقترح، وتنفيذ التجارب العملية، وتقييم الأداء باستخدام مقاييس معيارية، بالإضافة إلى مناقشة النتائج واستخلاص الاستنتاجات والتوصيات. كما يستعرض الفصل الأول الخلفية النظرية للمشكلة، ويعرض مشكلة البحث وأهدافه ومساهماته، إلى جانب تحديد نطاق الدراسة وحدودها، مما يشكل أساسًا علميًا متينًا لبقية فصول البحث.

Abstract

The aim of this research is to develop an advanced artificial intelligence model capable of detecting URL phishing attacks in light of the continuous rise of such attacks and their increasing reliance on sophisticated techniques that are difficult to detect using traditional security systems. Phishing attacks are considered among the most dangerous threats in the field of cybersecurity, as they target users through fraudulent links that imitate legitimate websites in order to steal sensitive information such as passwords and financial data.

The project is based on the need to establish an intelligent and effective solution that utilizes Deep Learning techniques to analyze the textual structure of URLs and identify hidden malicious patterns without requiring inspection of the external webpage content. The proposed model relies on a hybrid architecture that combines Convolutional Neural Networks (CNN) and Long Short-Term Memory networks (LSTM). CNN layers contribute to extracting local patterns within URLs, while LSTM layers analyze sequential relationships between URL components, thereby enhancing the model's ability to accurately distinguish between legitimate and phishing links.

This research also addresses several challenges associated with URL datasets, including class imbalance, variations in URL lengths, and the diversity of patterns used in modern phishing attacks. To overcome these challenges, a large and diverse dataset was utilized, along with advanced preprocessing techniques such as Character-Level Representation, which allows the model to achieve a deeper understanding of URL structures.

Experimental results demonstrated that the proposed model achieved outstanding performance in terms of accuracy and recall, reflecting its high efficiency in phishing detection and error reduction. This significantly contributes to improving user protection against cyber threats. Furthermore, the model was tested on new unseen data outside the training process and proved its ability to generalize effectively and operate efficiently in real-world environments.

This project provides a significant contribution to the field of cybersecurity by offering an intelligent tool that reduces the likelihood of users becoming victims of phishing attacks, while also supporting individuals and organizations with practical solutions for the early detection of suspicious links. In addition, the research presents a prototype model that can be further developed into a fully integrated system for practical applications or integrated into web browsers and security protection systems.

The research covers several essential aspects, including problem analysis, literature review, development of the proposed model, practical implementation, performance evaluation using standard metrics, as well as discussion of results and extraction of conclusions and future recommendations. The first chapter presents the theoretical background of the problem, including the research problem, objectives, contributions, scope, and limitations, thereby establishing a strong scientific foundation for the remaining chapters of the study.

بسم الله الرحمن الرحيم

{ فَتَعَالَى اللَّهُ الْمَلِكُ الْحَقُّ ۗ وَلَا تَعْجَلْ بِالْقُرْآنِ مِن قَبْلِ أَن يُقْضَىٰ إِلَيْكَ وَحْيُهُ وَقُل رَّبِّ زِدْنِي عِلْمًا } صدقَ اللّٰهُ العظيمُ.

سورة طه الآية (114)

إهـــدَاء

إلى من كانوا نبض هذا الطريق، وسرّ قوتنا في كل لحظة تعب أو ضعف،

إلى أهلنا الأعزاء، أنتم الدافع الذي جعلنا نواصل رغم كل الصعوبات،

بدعواتكم، بصبركم، وبابتسامتكم وقت العثرات، اكتمل هذا الحلم.

نهديكم ثمرة جهدٍ وسهرٍ وأملٍ طال انتظاره،

فأنتم من يستحق الفخر أولاً.

وإلى أصدقائنا الذين شاركونا درب التحدّي،

ضحكنا سوياً، تعبنا سوياً، وها نحن نصل معاً إلى خط النهاية،

نهديكم هذا العمل وفاءً لرحلةٍ مليئةٍ بالذكريات التي لن تُنسى.

شكر وتقدير

نتقدّم بخالص الشكر والتقدير لكل من قدّم لنا الدعم والمساندة خلال إعداد هذا العمل.

نتوجّه بامتنانٍ عميق إلى المشرف الدكتور الفاضل/ عبدالسلام خاقو على توجيهاته القيّمة وملاحظاته البنّاءة التي ساهمت في إثراء محتوى البحث وتطويره.

كما نعبّر عن شكرنا لـ أعضاء هيئة التدريس في قسم الأمن السيبراني على ما قدّموه لنا من علمٍ ومعرفة كانت الأساس في إنجاز هذا المشروع.

إقرار المشرف

أقر أنا الدكتور/ .....

بأن مشروع التخرج بعنوان:

كشف هجمـات التصيـد الاحتيـالي عبر الروابط باستخـدام الذكـاء الاصطناعـي

Detecting Phishing Attacks Using Artificial Intelligence

والمقدم من الطلاب:

1- .....

2- .....

3- .....

4- .....

5- .....

6.....

قد تم إنجازه تحت إشرافي في قسم .....، كلية .....، كجزء من متطلبات الحصول على درجة البكالوريوس في تخصص .....

وأشهد بأن العمل مستوفٍ للمتطلبات الأكاديمية.

اسم المشرف: .....

التوقيع: .....

التاريخ: ..../..../....

لجنة المناقشة

تم مناقشة المشروع المقدم من طلاب كلية الهندسة وتكنلوجيا المعلومات قسم الامن السيبراني في مشروع

كشف هجمات التصيد الاحتيالي عبر الروابط باستخدام الذكاء الاصطناعي

وبعد مناقشة الطلاب في محتويات المشروع وفيما له علاقة به تم قبول المشروع، وهو جزء من متطلبات الحصول على درجة البكالوريوس في كلية الهندسة وتكنولوجيا المعلومات.

م	الاسم	الصفة	التوقيع
1			
2			
3			

عميد كلية الهندسة وتكنلوجيا المعلومات

د/ مختار غيلان

المحتويات

الملخص I

Abstract..... II

الآية III

الإهداء IV

الشكر والتقدير V

إقرار المشرف..... VI

لجنة المناقشة..... VII

قائمة الاشكال..... VIII

قائمة الجداول IX

جدول الاختصارات XIII

الفصل الأول 1

1.1 المقدمـــة 2

2.1 مشكلة البحث: 3

3.1 أهداف المشروع: 3

4.1 مساهمة المشروع: 4

5.1 حدود المشروع:..... 4

6.1 خلاصة الفصل:..... 4

الفصل الثاني..... 5

1.2 مقدمـــــة..... 6

2.2 انواع التصيد 6

3.2 التصيد عبر الروابط 8

1.3.2 كيفية عمل الهجوم 8

2.3.2 ما يميزه عن الأنماط الأخرى..... 8

3.3.2 أسباب خطورته..... 8

4.2 الأدبيات والدراسات السابقة..... 9

1.4.2 الطرق التقليدية..... 9

2.4.2 أساليب الذكاء الاصطناعي..... 10

5.2 تحليل URL 14

6.2 استخراج الميزات 16

1.6.2 الميزات المعجمية (Lexical Features) 16

2.6.2 ميزات معتمدة على النطاق(Domain-Based Features) 17

3.6.2 ميزات معتمدة على الشبكة (Network-Based Features)..... 17

7.2 دور الذكاء الاصطناعي وتعلم الآلة في كشف التصيد عبر الروابط 18
1.7.2 الخوارزميات الشائعة المستخدمة 18
2.7.2 أنواع الميزات المستخدمة 20
3.7.2 استخراج البيانات وتدريب النماذج 21
4.7.2 آليات التقييم 21
5.7.2 مزايا التعلّم الآلي مقابل الطّرُق التقليدية 23
6.7.2 التحديات المحتملة 24
الفصل الثالث 25
1.3 مقدمة عن منهجية Agile 26
2.3 مراحل منهجية إعداد النموذج: 26
3.3 دراسة الجدوى 29
4.3 المخطط الزمني للمشروع 29
5.3 أدوات وتقنيات المشروع 30
6.3 إجراءات التنفيذ 30
7.3 مخطط بنية النظام 32
8.3 مراحل تدريب النموذج 34
9.3 آلية التقييم 34
10.3 الواجهة المقترحة 35
11.3 ملخص الفصل 35
الفصل الرابع 36
1.4 مقدمة الفصل 37
2.4 إعدادات التجربة وبيئة العمل 37
1.2.4 وصف مجموعة البيانات (Dataset) 37
2.2.4 معالجة عدم توازن البيانات 38
3.2.4 إعداد البيانات والمعالجة المسبقة 38
4.2.4 بنية النموذج 38
3.4 بيئة تطوير النظام 40
4.4 الواجهة الرسومية للنظام 41
5.4 خلاصة الفصل 43
الفصل الخامس 44
1.5 مقدمة الفصل 45
2.5 مناقشة النتائج 45
1.2.5 أداء النموذج المقترح 45
2.2.5 تحليل الأخطاء 45

3.2.5 تحليل أداء النموذج على روابط جديدة 46
4.2.5 تأثير عدد دورات التدريب (Epochs)..... 46
5.2.5 تأثير معالجة عدم توازن البيانات 46
3.5 الإجابة على أسئلة البحث..... 47
4.5 الاستنتاجات 47
5.5 خلاصة الفصل..... 49
الفصل السادس 50
1.6 مقدمة 51
2.6 عرض النتائج..... 51
3.6 تحليل النتائج..... 51
4.6 تقييم أداء النموذج 51
5.6 مناقشة النتائج..... 52
6.6 التحديات التي واجهت المشروع 52
7.6 التوصيات والأعمال المستقبلية..... 52
1.7.6 تحسينات مقترحة 52
2.7.6 تحسينات متوسطة المدى..... 52
3.7.6 توصيات بحثية مستقبلية 53
8.6 خلاصة الفصل 53
الخاتمة.....54
المراجع 55

قائمة الاشكال

الشكل 1:2 توضيح آلية عمل هجمات التصيد الاحتيالي.....6

الشكل 2:2 أبرز أنواع هجمات التصيد الاحتيالي التي تستهدف المستخدمين.....7

الشكل 3:2 دورة تنفيذ هجوم التصيد الاحتيالي.....7

الشكل 4:2 تفكيك الـ URL إلى مكوناته الأساسية.....14

الشكل 5:2 مخطط خوارزمية تقييم أداء النموذج.....22

الشكل 1:3 رسم توضيحي للمخطط الزمني.....29

الشكل 2:3 المخطط العام للمشروع.....30

الشكل 3:3 مخطط بنية النظام.....32

الشكل 4:3 مخطط بنية النموذج.....33

الشكل 5:3 الواجهة المقترحة.....35

الشكل 1:4 الواجهة الرئيسية للنظام.....41

الشكل 2:4 واجهة عرض نتيجة فحص الرابط (آمن) 41

الشكل 3:4 واجهة عرض نتيجة فحص الرابط (غير آمن).....42

الشكل 4:4 فحص الروابط المختصرة.....42

الشكل 5:4 فحص ملف كامل يحتوي على الروابط.....43

الشكل 1:5 يوضح منحنى دقة التحقق تحسن أداء النموذج واستقراره عبر مراحل التدريب.....48

الشكل 2:5 يوضح مصفوفة الارتباك كفاءة النموذج في تصنيف الروابط الآمنة والاحتيالية بدقة عالية.....48

الشكل 3:5 يوضح منحنى خسارة التدريب والتحقق استقرار عملية التعلم وتحسن أداء النموذج.....49

الشكل 4:5 يوضح منحنى ROC القدرة العالية للنموذج على التمييز بين الروابط الآمنة وروابط التصيد.....49

قائمة الجداول

جدول1 الاختصارات XIII
جدول 1:2 خلاصة الدراسة السابقة 13
جدول 1:4 توزيع عينات مجموعة البيانات 37
جدول 2:4 طبقات النموذج المصممة 39
جدول 3:4 معلمات تدريب نموذج CNN-BiLSTM 40

جدول الاختصارات

جدول1 الاختصارات

م	المعنى بالعربية	الاسم الكامل بالإنجليزية	الاختصار
1	الذكاء الاصطناعي	Artificial Intelligence	**AI**
2	رقم النظام المستقل	Autonomous System Number	**ASN**
3	المساحة تحت المنحنى	Area Under Curve	**AUC**
4	سلطة الشهادات	Certificate Authority	**CA**
5	شبكة توصيل المحتوى	Content Delivery Network	**CDN**
6	الشبكة العصبية الالتفافية	Convolutional Neural Network	**CNN**
7	قيم مفصولة بفواصل	Comma-Separated Values	**CSV**
8	نظام أسماء النطاقات	Domain Name System	**DNS**
9	نموذج كائن المستند	Document Object Model	**DOM**
10	الشبكة الالتفافية المحسنة لإزالة الضوضاء	Enhanced Denoising Convolutional Neural Network	**EDCNN**
11	الذاكرة الطويلة قصيرة المدى	Earth Mover's Distance	**EMD**
12	سلبي كاذب	False Negative	**FN**
13	إيجابي كاذب	False Positive	**FP**
14	شجرة القرار المعززة بالتدرج	Gradient Boosting Decision Tree	**GBDT**
15	الشبكة العصبية البيانية	Graph Neural Network	**GNN**
16	الوحدة المتكررة ذات البوابات	Gated Recurrent Unit	**GRU**
17	لغة ترميز النص التشعبي	HyperText Markup Language	**HTML**
18	بروتوكول نقل النص التشعبي	HyperText Transfer Protocol	**HTTP**

م	المعنى بالعربية	الاسم الكامل بالإنجليزية	الاختصار
19	بروتوكول نقل النص التشعبي الآمن	HyperText Transfer Protocol Secure	**HTTPS**
20	بروتوكول الإنترنت	Internet Protocol	**IP**
21	تدوين كائنات جافا سكريبت	JavaScript Object Notation	**JSON**
22	خوارزمية الجيران الأقرب	K-Nearest Neighbors	**KNN**
23	آلة تعزيز التدرج الخفيفة	Light Gradient Boosting Machine	**LightGBM**
24	الذاكرة الطويلة قصيرة المدى	Long Short-Term Memory	**LSTM**
25	التعلم الآلي	Machine Learning	**ML**
26	المتعصب متعدد الطبقات	Multilayer Perceptron	**MLP**
27	معالجة اللغة الطبيعية	Natural Language Processing	**NLP**
28	متعدد النواة المتوازي	Parallel Multi-Core	**PMC**
29	خاصية تشغيل المستقبل	Receiver Operating Characteristic	**ROC**
30	الشبكة العصبية المتكررة	Recurrent Neural Network	**RNN**
31	طبقة المقابس الآمنة	Secure Sockets Layer	**SSL**
32	آلة الدعم الشعاعي	Support Vector Machine	**SVM**
33	تكرار المصطلح - التردد العكسي للوثيقة	Term Frequency-Inverse Document Frequency	**TF-IDF**
34	سلبي حقيقي	True Negative	**TN**
35	إيجابي حقيقي	True Positive	**TP**
36	جامعة كاليفورنيا في إيرفين	University of California, Irvine	**UCI**
37	مؤشر مواقع الإنترنت	Uniform Resource Locator	**URL**
38	تعزيز التدرج المتطرف	Extreme Gradient Boosting	**XGBoost**

الفصل الأول
خلفية البحث وأهميته

1.1 المقدمـــة

شهد العالم في السنوات الأخيرة زيادة ملحوظة في الهجمات الإلكترونية، والتي أصبحت تهدد الأفراد والمؤسسات على حد سواء.

من بين هذه الهجمات، يعتبر **التصيد الاحتيالي (Phishing)** أحد أكثر التهديدات شيوعًا وخطورة، حيث يستهدف المهاجمون المعلومات الحساسة للمستخدمين مثل كلمات المرور، بيانات البطاقات البنكية، الحسابات الإلكترونية، أو أي معلومات يمكن استغلالها ماليًا أو معلوماتيًا. تعتمد هجمات التصيد على الخداع النفسي والإقناع، إذ يتم تصميم رسائل أو مواقع مزيفة تشبه الرسائل الرسمية للبنوك أو المؤسسات، مما يجعل من الصعب على المستخدم العادي التمييز بين الأصلية والمزيفة.

تكمن خطورة التصيد الاحتيالي في بساطته وفعاليته معًا.

فحتى المستخدمين ذوي الخبرة التقنية قد يقعون ضحايا لهجمات متقنة الصنع، نظرًا لتطور أساليب المهاجمين واعتمادهم على أساليب اجتماعية متقدمة لإقناع الضحايا بالكشف عن معلوماتهم. وقد أظهرت الدراسات أن جزءًا كبيرًا من الهجمات الإلكترونية الناجحة يعود إلى قلة الوعي الأمني للمستخدمين، وعدم القدرة على التعرف على العلامات التحذيرية في الرسائل المشبوهة.

مع تطور الذكاء الاصطناعي، ظهرت إمكانيات كبيرة لمواجهة هذه التهديدات بشكل أكثر فعالية. **تقنيات الذكاء الاصطناعي والتعلم الآلي** تسمح بتحليل البيانات بشكل أوسع وأسرع، واكتشاف الأنماط غير الطبيعية في الرسائل الإلكترونية والروابط المشبوهة، مما يساهم في تقليل نسبة الضحايا وتحسين الأمن السيبراني. وقد أظهرت الأبحاث أن استخدام الذكاء الاصطناعي يمكن أن يزيد من دقة كشف هجمات التصيد مقارنة بالطرق التقليدية، التي غالبًا ما تعتمد على قواعد ثابتة أو تحديثات يدوية للبرامج.

هذا البحث يهدف إلى استكشاف **إمكانية استخدام الذكاء الاصطناعي في الكشف المبكر لهجمات التصيد الاحتيالي**، من خلال تصميم نموذج تحليلي قادر على التمييز بين الرسائل الآمنة والاحتيالية بدقة عالية. كما يسعى البحث إلى تقييم أداء النموذج ومقارنته بالحلول التقليدية، بالإضافة إلى تقديم توصيات عملية لتعزيز الوعي الأمني لدى المستخدمين.

من خلال هذا البحث، سيتمكن القارئ من فهم طبيعة هجمات التصيد الاحتيالي، أساليب المهاجمين، والتحديات التي تواجه الكشف التقليدي، إلى جانب معرفة الدور الكبير الذي يمكن أن يلعبه الذكاء الاصطناعي في حماية الأفراد والمؤسسات من هذه التهديدات المتزايدة. وبذلك، يوفر هذا البحث أساسًا متينًا لدراسة باقي فصول البحث، والتي ستتطرق إلى منهجية المشروع، أدواته، ونتائج تطبيق نموذج الذكاء الاصطناعي في بيئة عملية.

2.1 مشكلة البحث:

تُعد هجمات التصيد الاحتيالي عبر الروابط (URLs) من أبرز التهديدات التي تواجه الأمن السيبراني في الوقت الحالي، نظرًا للتطور المستمر في أساليب المهاجمين وقدرتهم على إنشاء روابط خبيثة تحاكي المواقع الحقيقية بشكل يصعب اكتشافه. وعلى الرغم من وجود أنظمة حماية تقليدية، إلا أنها لا تزال تعاني من ضعف في القدرة على تحليل هذه الروابط المتطورة ومواكبة التغيرات المستمرة في تقنيات الهجوم.

كما يعتمد العديد من المستخدمين على طرق يدوية في التحقق من الروابط، وهي طرق غير فعّالة خاصة مع الروابط المصممة باحترافية عالية، مما يزيد من احتمالية الوقوع ضحية لهجمات التصيد. بالإضافة إلى ذلك، فإن معظم الحلول الحالية تعتمد على تحليل محتوى الصفحات المرتبطة بالرابط، وهو ما قد لا يكون متاحًا دائمًا أو يتطلب وقتًا وجهدًا إضافيًا.

بناءً على ذلك، تبرز الحاجة إلى تطوير نظام ذكي قادر على تحليل الروابط بشكل مباشر دون الاعتماد على محتوى الصفحات الخارجية، وذلك بهدف تحقيق كشف سريع ودقيق للروابط الاحتيالية باستخدام تقنيات الذكاء الاصطناعي.

3.1 أهداف المشروع:

يهدف هذا المشروع إلى تطوير نظام ذكي يعتمد على تقنيات الذكاء الاصطناعي لتحليل الروابط الإلكترونية والكشف عن روابط التصيد الاحتيالي بدقة عالية. ويسعى المشروع إلى تقديم حل عملي يساعد في حماية المستخدمين من الهجمات الإلكترونية المتزايدة، من خلال الاعتماد على تحليل بنية الرابط دون الحاجة إلى فحص محتوى الصفحات الخارجية، مما يساهم في تحقيق سرعة وكفاءة أعلى في عملية الكشف.

وانطلاقًا من ذلك، تتمثل أهداف المشروع فيما يلي:

1. تطوير نموذج ذكاء اصطناعي متخصص في تحليل الروابط، بحيث يقوم بفحص هيكل الرابط (URL) ومكوناته لاكتشاف الأنماط الاحتيالية .
2. توفير نظام كشف آلي يعمل بشكل فوري لتحليل الروابط، مما يقلل من الاعتماد على المستخدم ويعزز مستوى الحماية المستمرة .
3. تقديم نتائج واضحة ومباشرة للمستخدم حول حالة الرابط، سواء كان آمنًا أو يشكل تهديدًا.

4.1 مساهمة المشروع:

يساهم المشروع في:

- توفير أداة ذكية قادرة على تقليل الخسائر المالية والمعلوماتية الناتجة عن التصيد.
- دعم المؤسسات والأفراد بحلول فعالة لمواجهة التهديدات الإلكتروني..
- إثراء الجانب الأكاديمي والبحثي في مجال الأمن السيبراني والذكاء الاصطناعي من خلال تطبيق عملي.
- تقديم نموذج أولي يمكن تطويره لاحقًا ليصبح نظامًا متكاملًا يستخدم في الواقع العملي.
- تعزيز وعي المستخدمين بمخاطر الروابط المشبوهة من خلال تقديم نتائج واضحة وسهلة الفهم.
- إمكانية دمج النظام مع تطبيقات أو متصفحات مستقبلًا لتوفير حماية مستمرة أثناء تصفح الإنترنت.
- الاستفادة من تقنيات التعلم العميق في تحليل أنماط الروابط بشكل أكثر دقة مقارنة بالأساليب التقليدية.
- تحسين دقة التمييز بين الروابط الآمنة والخبيثة من خلال دمج أكثر من نموذج تعلم آلي في نظام واحد.
- دعم تحليل الروابط في الزمن الحقيقي (Real-Time) مما يساعد المستخدم على اتخاذ قرار سريع قبل فتح الرابط.

5.1 حدود المشروع:

- المشروع يركز فقط على كشف التصيد الاحتيالي عبر الروابط، ولا يشمل جميع أنواع الهجمات السيبرانية.
- يعتمد على بيانات متاحة ومجمّعة مسبقًا (Public Datasets) لأغراض التدريب والاختبار.
- النموذج سيتم اختباره في بيئة أكاديمية، وقد لا يكون جاهزًا للاستخدام التجاري المباشر دون تطوير إضافي.

6.1 خلاصة الفصل:

تناول هذا الفصل عرضًا عامًا لموضوع البحث، بدءًا من المقدمة التي أوضحت خطورة التصيد الاحتيالي كواحد من أبرز التهديدات السيبرانية،

ثم استعراض مشكلة المشروع المتمثلة في ضعف الطرق التقليدية في كشف هذه الهجمات.

كما تم توضيح فكرة المشروع وأهدافه، مع إبراز مساهمته في المجال الأكاديمي والعملي.

إضافة إلى ذلك، تم تحديد حدود المشروع، وكذلك الأدوات التي ستُستخدم في عملية التنفيذ.

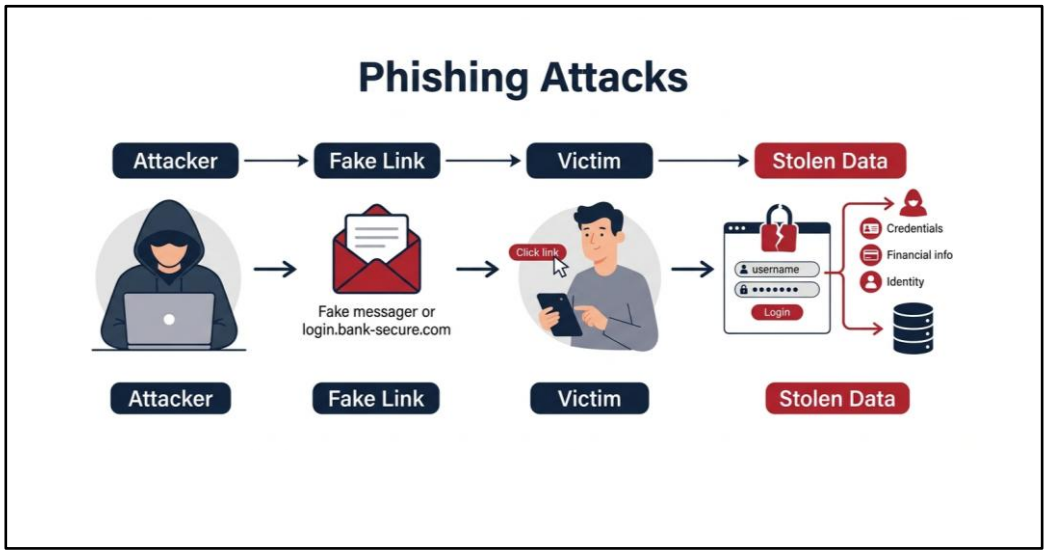
يمثل هذا الفصل الأساس النظري الذي يُبنى عليه باقي فصول البحث، ويهيئ القارئ لفهم المنهجية والتطبيق العملي لاحقًا.

الفصل الثاني

الادبيات والدراسات السابقة

1.2 مقدمـــــة

يعد التصيد الاحتيالي أحد أكثر أشكال الهجمات السيبرانية شيوعًا وخطورة في الوقت الحالي. يقوم على أسلوب خداعي يهدف إلى إقناع الأفراد بالكشف عن معلومات حساسة مثل أسماء المستخدمين وكلمات المرور أو البيانات المالية. ظهر مفهوم التصيد الاحتيالي لأول مرة في منتصف التسعينيات، وجاءت التسمية من كلمة الصيد (Fishing) في إشارة إلى إلقاء طعم لاصطياد الضحايا. ومع مرور الوقت، طور المهاجمون أساليبهم وأصبحوا غالبًا ما ينتحلون شخصيات مؤسسات موثوقة مثل البنوك أو الهيئات الحكومية أو الشركات المعروفة [1].



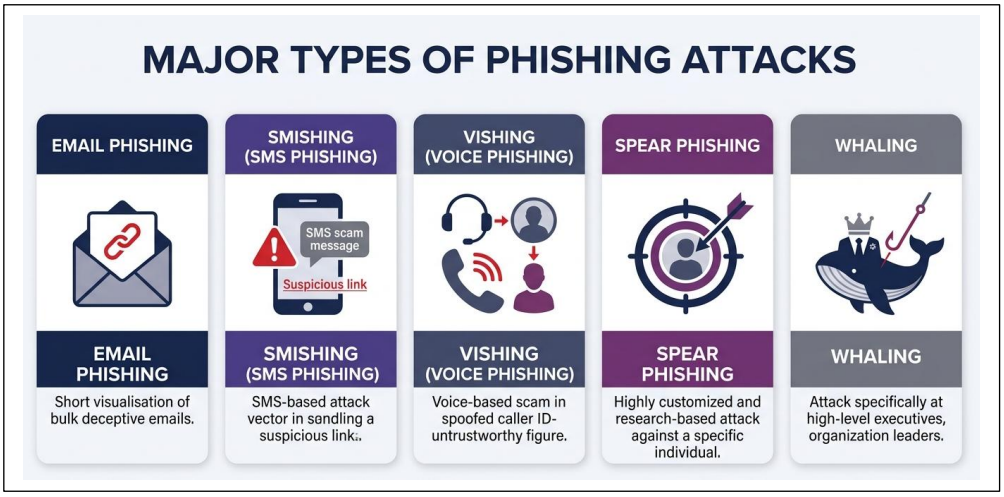
الشكل 1:2 توضيح آلية عمل هجمات التصيد الاحتيالي من خلال استخدام روابط ورسائل مزيفة لخداع المستخدمين

2.2 انواع التصيد

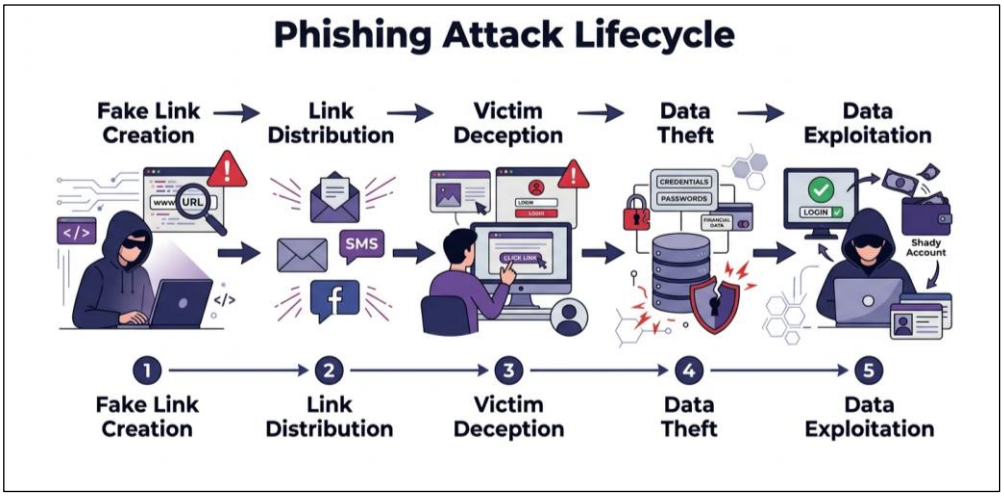
تُعتبر هجمات التصيد الاحتيالي من أكثر التهديدات السيبرانية انتشارًا وتطورًا في عالم اليوم، إذ لم تعد محصورة في شكل واحد أو وسيلة تقليدية، بل تنوعت بشكل كبير لتستهدف الضحايا عبر قنوات متعددة. وغالبًا ما يعتمد المهاجمون على الخداع والإقناع النفسي، مستغلين ثقة المستخدمين في المؤسسات أو المنصات الرقمية، للوصول إلى بياناتهم السرية أو أموالهم. من أبرز هذه الأنواع التصيد عبر البريد الإلكتروني، وهو الشكل الأكثر شيوعًا، حيث يتلقى المستخدم رسائل تبدو وكأنها مرسلة من جهة رسمية مثل البنوك أو الشركات، وتحتوي عادة على روابط تؤدي إلى مواقع مزيفة تهدف إلى سرقة بيانات الدخول [2]. وهناك أيضًا التصيد الموجه، الذي يُعرف بالـ"Spear Phishing"، ويستهدف أشخاصًا أو جهات بعينها باستخدام معلومات شخصية حقيقية تجعل الرسالة أكثر إقناعًا وخطورة [3]. ومن بين الأشكال المتقدمة ما يسمى "صيد الحيتان"، وهو نوع خاص يركز على المسؤولين التنفيذيين وأصحاب المناصب العليا في الشركات، بهدف السيطرة على حساباتهم أو استغلال صلاحياتهم للوصول إلى أموال ومعلومات حساسة [2].

كذلك يبرز التصيّد عبر الرسائل النصية القصيرة أو ما يُعرف بالـ"Smishing" ، والذي أصبح شائعًا مع الانتشار الكبير للهواتف الذكية، حيث تصل رسائل تبدو بريئة للمستخدمين لكنها تحمل روابط خبيثة [4]. أما التصيّد الصوتي، أو"Vishing" ، فيعتمد على مكالمات هاتفية يجريها المهاجمون بانتحال صفة موظفي خدمة العملاء أو جهات حكومية، لإقناع الضحية بالكشف عن معلومات شخصية [4]. ويشبه ذلك التصيّد بالاستنساخ، حيث يقوم المهاجمون بإرسال نسخة مطابقة من بريد إلكتروني شرعي مع تعديل الروابط أو المرفقات لتكون ضارة [2]. كما أن بعض الهجمات تهدف مباشرة إلى تثبيت برمجيات خبيثة عبر مرفقات أو روابط مزيفة، بينما يركز البعض الآخر على صفحات تسجيل دخول وهمية مصممة لسرقة أسماء المستخدمين وكلمات

المرور. كما بدأ المهاجمون يستغلون بروتوكول HTTPS والرمز الأمني لإقناع الضحايا بأن الموقع المزيف آمن [2].



الشكل 2:2 أبرز أنواع هجمات التصيد الاحتيالي التي تستهدف المستخدمين



الشكل 3:2 دورة تنفيذ هجوم التصيد الاحتيالي

3.2 التصيد عبر الروابط

التصيد الاحتيالي عبر الروابط هو فرع من هجمات التصيّد يركّز على استغلال عناوين الإنترنت (URLs) كوسيط لإيهام المستخدمين بالانتقال إلى مواقع تبدو شرعية لكنها مخصصة لسرقة بيانات حساسة أو نشر برمجيات خبيثة. يختلف هذا النمط عن أشكال التصيّد الأخرى بتركيزه على البنية النصية للرابط، معلومات المضيف (host) والوجهة النهائية عند فك أي رابط مختصر أو تتبع سلسلة إعادة التوجيه. نظراً لأن الروابط تمثل نقطة التماس المباشرة بين المهاجم والمستخدم، فإن تحليلها يوفر نقطة دخول عملية لمعادلة خطر الهجوم قبل تحميل الصفحة أو تنفيذ أية إجراءات على جهاز الضحية [5].

1.3.2 كيفية عمل الهجوم

- إنشاء رابط مزيف :قد يضيف المهاجم حرفا واحدًا غير ملحوظ إلى اسم النطاق (مثال: paypa1.com بدلًا من paypal.com) أو يستخدم نطاقًا فرعيًا طويلًا ليوهم المستخدم (paypal.secure-login.example.com).
- استخدام اختصارات الروابط: مثل خدمات bit.ly أو tinyurl لإخفاء عنوان الوجهة الحقيقي ومنع المستخدم من التحقق البصري.
- سلسلة إعادة التوجيه (Redirect Chains): قد يوجه الرابط أولاً إلى موقع عادي أو وسيط ثم يعيد التوجيه تلقائيًا إلى صفحة التصيد، مما يصعب على أنظمة الأمان كشفه بالتحليل السطحي [6].
- استغلال HTTPS: أصبح المهاجمون يستخدمون شهادات SSL مجانية ليظهر الرابط وكأنه آمن (ظهور القفل الأخضر لا يضمن الشرعية) [7].

2.3.2 ما يميزه عن الأنماط الأخرى

- يختلف عن التصيد بالبريد الإلكتروني (Email Phishing) في تركيزه على البنية النصية للـ URL بدلاً من نص الرسالة.
- لا يتطلب أحيانًا تقنيات اجتماعية متقدمة؛ يكفي أن ينقر المستخدم على رابط يبدو مألوفًا.
- يمكن استخدامه في وسائل التواصل الاجتماعي، الإعلانات، أو الرسائل الفورية، وليس في البريد الإلكتروني فقط.

3.3.2 أسباب خطورته

- التفاعل الفوري: الرابط هو نقطة اللمس الأولى بين المهاجم والمستخدم، ويمكن أن ينفذ الهجوم بمجرد النقر دون الحاجة لفتح مرفقات.
- انتشاره السهل: يمكن مشاركة روابط التصيد بسرعة وبطرق يصعب تتبعها.

- تجاوز بعض أنظمة الأمان التقليدية: مثل الفلاتر النصية للبريد الإلكتروني التي قد تفحص النصوص دون تحليل متعمق لبنية الروابط.

التركيز على الروابط يجعل من التصيد عبر الـ URL نقطة تحذير مبكرة يمكن فحصها قبل أن يتعرض المستخدم أو النظام لأي ضرر. تحليل مكونات الرابط -بدءًا من البروتوكول والمضيف وحتى المعلمات وسلسلة التوجيه- يعد أداة فعالة ضمن أنظمة كشف التصيد المعتمدة على الذكاء الاصطناعي [8].

4.2 الأدبيات والدراسات السابقة

شهدت السنوات الأخيرة تطوّرًا ملحوظًا في تقنيات الكشف عن عناوين URL الاحتيالية، حيث قدّم العديد من الباحثين مناهج مختلفة لحماية المستخدمين من المواقع المزيفة. واستفادت دراسات كثيرة من أساليب التعلم الآلي والتعلم العميق لتحقيق دقة مرتفعة في اكتشاف التصيد.

1.4.2 الطرق التقليدية

طرحت عدة أساليب تقليدية للكشف عن التصيد يمكن تصنيفها إلى خمس فئات رئيسية :القوائم البيضاء، القوائم السوداء، قوائم المحتوى، التشابه البصري، والاستراتيجيات القائمة على عنوان URL.

1.1.4.2 نهج القوائم البيضاء

اقترحت دراسة أجراها عدد من الباحثين في مجال الأمن السيبراني عام (2021) نهجًا يعتمد على استخدام المواقع الموثوقة ضمن ما يُعرف بالقوائم البيضاء (Whitelist)، حيث يتم إجراء فحص تشابه يعتمد على تحليل عنوان الرابط (URL) ومقارنة استعلامات نظام أسماء النطاقات (DNS) بهدف التمييز بين المواقع الحقيقية والمواقع الاحتيالية [9].

كما قدمت دراسة أخرى في [10] نهجًا مختلفًا يعتمد على إنشاء قائمة بيضاء ديناميكية تقوم بحفظ بيانات تسجيل الدخول السابقة للمستخدم، ومن ثم تنبيهه عند حدوث أي نشاط غير مألوف قد يشير إلى محاولة تصيد احتيالي.

وعلى الرغم من فعالية هذه الأساليب في الحد من الهجمات، إلا أنها تواجه بعض القيود، من أبرزها صعوبة مواكبة التغيرات المستمرة في المواقع الموثوقة، بالإضافة إلى إمكانية انتحال هذه المواقع من قبل المهاجمين.

2.1.4.2 نهج القوائم السوداء

تستخدم متصفحات كبرى مثل جوجل القوائم السوداء لتحديث المواقع المحظورة دوريًا. اقترح [11] آلية جديدة لإنشاء هذه القوائم لتجاوز مشاكل التحديث والصيانة، لكنه اعتمد على خدمة خارجية للبحث عن اسم النطاق، مما سبّب بطئًا في الأداء. كما تبقى هذه الطرق ضعيفة أمام هجمات "الساعة صفر" حيث لا تكون المواقع الاحتيالية الجديدة مدرجة بعد[12]. وفي دراسة لاحقة، قدّم [13] نموذج PhisNet الذي اعتمد على خصائص مثل عنوان IP وبنية الدليل واسم العلامة التجارية، وحقق دقة بلغت 95% [14].

3.1.4.2 نهج قوائم المحتوى

قدم [15] أسلوب CANTINA، الذي استخدم خوارزمية TF-IDF للكشف عن مواقع التصيد من خلال تحليل المحتوى النصي. غير أن الاعتماد على محركات البحث أدّى إلى ارتفاع معدل الإيجابيات الخاطئة. ولتحسين الأداء، طور CANTINA+ بواسطة [16] الذي رفع الدقة إلى أكثر من 92% وخفض الإيجابيات الخاطئة إلى 0.4%، إلا أن الاعتماد على خدمات خارجية بقي عائقًا [17].

4.1.4.2 أساليب التشابه البصري

اعتمدت بعض الدراسات [18] على مقارنة البنية البصرية للصفحات، باستخدام مستويات متعددة من مصفوفات التشابه (كتلي، تخطيطي، وأسلوب عام). وقد استخدم [11] مقياس مسافة حركة الأرض (EMD) لمقارنة بصمات الصور، وبلغت نسبة الكشف الصحيحة 89% مع نسبة إيجابيات كاذبة 0.71%. ورغم دقتها، واجهت هذه الأساليب تحديات في زمن المعالجة. دراسات أخرى اعتمدت الانحدار اللوجستي لمحاكاة التشابه الإدراكي وحققت كشفًا بنسبة 100%، لكن بمعدل إيجابيات كاذبة 0.74% [19].

5.1.4.2 الأساليب القائمة على عنوان URL

تناقش أدناه عدة مقاربات تعتمد على التعلم الآلي والتعلم العميق لاكتشاف هجمات التصيد الاحتيالي القائمة على عناوين URL.

2.4.2 أساليب الذكاء الاصطناعي

- **الأساليب المعتمدة على التعلم الآلي**

في هذه الدراسة، تم اقتراح نهج يعتمد على المسح والكشف الهيكلي لتقنيات اكتشاف عناوين URL الضارة باستخدام التعلم الآلي. للكشف عن عناوين URL الخبيثة، استخدمت تقنيات مختلفة مثل :استخراج السمات، تمثيل السمات، تصميم الخوارزميات، القوائم السوداء، والأساليب الاستدلالية. وتم تطبيق التصنيف الثنائي لتصنيف النتائج إلى فئتين: "ضارة" أو "حميدة". يهدف التعلم الآلي إلى تعظيم دقة التنبؤ من خلال تمثيل السمات متبوعًا بعملية التصنيف، باستخدام نهج تحسين قائم على البيانات [20]. في هذه الدراسة، تمت مراقبة 19,066 هجوم تصيد احتيالي، وكان أكثر من 90% من هذه الهجمات حقيقيًا. استخدمت عدة تقنيات لمواجهة هجمات التصيد، حيث تحقق كشف سريع وفعال باستخدام أداة وقائية. كما تم إنشاء موقع ويب لاكتشاف التصيد الاحتيالي باستخدام نهج التجميع الهرمي، حيث جمعت أعداد متجهات من العلامات. تم توليد هذه الأعداد من نماذج DOM لمواقع الويب التي تم استهدافها وفقًا لمسافاتها النسبية. واستخدمت خوارزميات التعلم الآلي لاحقًا لتصنيف مواقع الويب المكتشفة [21].

إجراء مسح حول أساليب كشف التصيد التي تم تناولها في هذه الدراسة وتقنيات كشف عناوين URL الخبيثة باستخدام التعلم الآلي. تم تقديم العديد من الصيغ لمهمة الكشف، ومن خلال الاستفادة من نماذج التعلم الآلي، حددت الدراسة التصنيفات، والمساهمات، والمشكلات المختلفة التي سببت فجوة في كشف عناوين URL الخبيثة. علاوة على ذلك، استعرضت المقالة مجموعة من الدراسات الحديثة والشاملة، والتي تعد أيضًا مفيدة للممارسين في مجال الأمن السيبراني. بالإضافة إلى ذلك،

تم ذكر العديد من مجالات البحث المفتوحة والتحديات غير المحلولة التي يمكن العمل عليها مستقبلًا[22] . في مكان آخر، تم اقتراح نظام أداة مكافحة التصيد في الوقت الفعلي للكشف عن عناوين URL الخبيثة، في دراسة استخدمت سبع خوارزميات تصنيف مختلفة وخصائص تعتمد على معالجة اللغة الطبيعية (NLP). استند هذا العمل إلى دراسات سابقة. واقترحت الدراسة نظامًا مختلفًا، مستقلًا عن اللغة، باستخدام كمية ضخمة من بيانات التصيد والبيانات الشرعية، وتنفيذًا فعليًا من عمليات كشف المواقع. تم استخدام مصنفات تعلم آلي غنية بالخصائص لاكتشاف الخدمات بشكل مستقل. تم استخدام خوارزمية الغابة العشوائية (Random Forest) مع خصائص تعتمد على NLP لتقييم الأداء، حيث حققت دقة بلغت 97.89% [23]. في حالة أخرى، في دراسة حول كشف عناوين URL للتصيد، اعتمد الباحثون على تمثيل موزّع للكلمات باستخدام عنوان URL معطى. تم استخدام سبع خوارزميات تعلم آلي مختلفة للتنبؤ بما إذا كانت المواقع عبارة عن مواقع تصيد أم لا. قدمت هذه الخوارزميات مستوى أداء مرضٍ، متفوقًا على الدراسات السابقة. كما تم التعرف على أحرف غير مرئية، وتم اكتشاف فقدان للمعلومات الدلالية باستخدام نموذج حقيبة الكلمات Bag-of-Wordsالذي احتفظ بالمعلومات الدلالية [24].

قدمت دراسة أخرى إضافة لمتصفح قائمة على التعلم الآلي لكشف التصيد القائم على عنوان URL. درب الباحثون مجموعة بيانات UCI باستخدام نموذج الغابة العشوائية وطبعوا 30 خاصية، مما يعني أنه تطلب خصائص إضافية لتدريبه في بيئة الوقت الفعلي. ووجدت الدراسة أن من بين 30 خاصية، هناك 16 خاصية لا تعتمد على خدمة طرف ثالث؛ ومع ذلك، أظهرت النتائج أن النظام حقق معدل دقة أعلى بنسبة 89.6% من الأنظمة المتقدمة في كشف التصيد عبر Chrome [25].

في دراسة سابقة أخرى، تم تطوير موقع ويب لاكتشاف هجمات التصيد المعتمدة على عنوان URL. تم استخدام مكتبة JSoup HTML Parser (JHP) في لغة Java للكشف وفقًا لثلاث مراحل: (أ) استخدام JSoup لتحليل هيكل DOM للموقع المراد كشفه. (ب) تحليل عدد الروابط من هيكل DOM وتحليل قيم السمات. (ج) استخدام الآلة الحاسبة المرتبطة لاستخراج القيمة المحددة بين 0 و1. تم استخدام 300 عنوان URL للعمل التجريبي لتقييم الأداء. أظهرت النتائج دقة بلغت 99.97% في كشف التصيد القائم على عنوان URL باستخدام الآلة الحاسبة المرتبطة. ومع ذلك، فإن استخدام مجموعة بيانات كبيرة كهذه يجعل الكشف أكثر خطورة، كما تم تجاهل العديد من الخصائص [26].

- **الطرق المعتمدة على التعلم العميق**

تم اقتراح نموذج للتعلم العميق باستخدام شبكة عصبية مكشوفة. في هذا النموذج، يؤخذ سلسلة قصيرة من الأحرف الخام كمدخل، وتستخرج الميزات وتصنف عبر تمثيل على مستوى الأحرف باستخدام شبكة عصبية التفافية (CNN) لاكتشاف روابط التصيّد. تضمنت السلسلة المدخلة عناصر أمنية، روابط محتملة خبيثة، مسارات للملفات، وغيرها. باستخدام نموذج التعلم العميق، صُمِّمت ميزات عميقة مؤتمتة، وتم استخلاص الروابط الخبيثة، مما حقق أداءً أفضل بنسبة 5% إلى 10 %مقارنة بالنتائج المتقدمة في المجال [27]. في دراسة أخرى، اقتُرحت عدة معماريات جديدة لتصنيف البرمجيات الخبيثة باستخدام نموذج اللغة LSTM (Long Short-Term Memory) ونموذج اللغة GRU (Gated Recurrent Unit).

كما اقتُرح مصنف برمجيات خبيثة من مرحلة واحدة يعتمد على شبكة عصبية التفافية على مستوى الحروف (CNN). أظهرت مقارنة النتائج أن نموذج LSTM حقق نتائج أفضل مع التجميع الزمني الأقصى (temporal max pooling)، وأن الانحدار اللوجستي حسن النتائج بنسبة 31.3%. وقد تبين أنه أكثر قدرة على التقاط الاعتمادات طويلة المدى، وساعدت نماذج اللغة العصبية في رفع أداء كشف البرمجيات الخبيثة [28]. أظهرت الشبكات العصبية البيانية (GNNs) إمكانات كبيرة من خلال استغلال البنى البيانية، حيث تُعد خصائص أساسية مثل الاتصال وقابلية التشخيص مهمة جدًا لأداء نماذج GNN، إذ تؤثر على سلامة البنية البيانية للمدخلات. يوفر البحث في قابلية التشخيص لأنواع مختلفة من الرسوم البيانية رؤى قيّمة حول قوة تحمّل النماذج وقدرتها على مقاومة الأعطال. ومن الدراسات ذات الصلة التي يمكن أن تعزز أداء النماذج في كشف التصيّد: الأبحاث المتعلقة بقابلية تشخيص رسوم-bubble sort starوفق نماذج PMC و *MM [29]. إضافة إلى ذلك، توفر العمليات التكرارية لاستخراج أنماط العلاقات بين الكيانات ونماذج العمليات التجارية (كما في عمل Javed و Lin على إطار iMER)[30] تقنيات قيمة لتحسين استخراج الأنماط البيانية، يمكن تكييفها لتعزيز كفاءة نماذج كشف التصيد.

في دراسة سابقة أخرى، تم اقتراح نهج لاستخراج ميزات روابط URL من مواقع التصيد تلقائيًا باستخدام الشبكات العصبية العميقة. حيث استُخدم نموذج CNN لاستخراج الميزات المكانية على مستوى الحروف من الروابط المدمجة، ثم استخدم Bi-LSTM لتصنيفها وتوليدها على مستوى الكلمات، بينما استُخدم نموذج RNN هرمي مع آلية انتباه زمني لتمثيل الرابط. ولأجل الكشف عن روابط التصيّد، استُخدم تمثيل ميزات مندمج يدمج بين شبكة CNN ثلاثية الطبقات والمُدرك متعدّد الطبقات (MLP) [31]. كما تم اقتراح إطار عمل يعتمد على التعلم العميق لكشف روابط التصيّد، حيث يستطيع إضافة في المتصفح تحديد ما إذا كان خطر التصيّد في الوقت الفعلي مرتفعًا أو منخفضًا. وبناءً على زيارات المستخدم للصفحات، يُظهر له رسالة تحذير. وقد جُمعت خوادم تنبؤ لحظية لتوليد استراتيجيات متعددة بهدف تحسين الكشف عن التهديدات. تمكّن هذا الإطار من التعرف على الإنذارات الكاذبة، وتقليل زمن حساب التنبؤ، والتعرّف على القوائم البيضاء والسوداء بالاعتماد على تنبؤات التعلم الآلي باستخدام مجموعات بيانات متعددة. ساعد نموذج RNN-GRU هذا الإطار على تحقيق دقة بلغت 99.18%، مما يثبت جدواه [29].

وفي دراسة أخرى، استخدمت شبكة عصبية اصطناعية متكيفة ذاتية البنية لتصنيف الروابط الصحيحة والمزيفة. تعد ميزات روابط التصيد مشكلات مستمرة من حيث الكشف والتحديد، نظرًا لتغير أنواع صفحات الويب باستمرار. حاول الباحثون معالجة هذه التحديات عبر معالجة بنية الشبكة تلقائيًا، مما أظهر قدرة عالية على تقبل البيانات المزعجة، وتحمل الأخطاء، ودقة تنبؤية مرتفعة، تم تحقيقها عبر ضبط قيم مختلفة لعدد العصور (epochs) في التجارب [32]. وأخيرًا، اقترح مؤلفون نظامًا لكشف تسلسلات روابط التصيّد الخبيثة في سجلات البروكسي بالاعتماد على نظام يسمى إزالة الضوضاء الحديثة عبر الشبكة العصبية الالتفافية (EDCNN). سعى الباحثون لإزالة التأثيرات السلبية غير الضارة من النظام عبر مقارنة مواقع URL بمواقع أخرى. ساعد هذا النظام على تقليل تكلفة التنفيذ بنسبة 47% باستخدام CNN سريعة لكشف روابط التصيد [33].

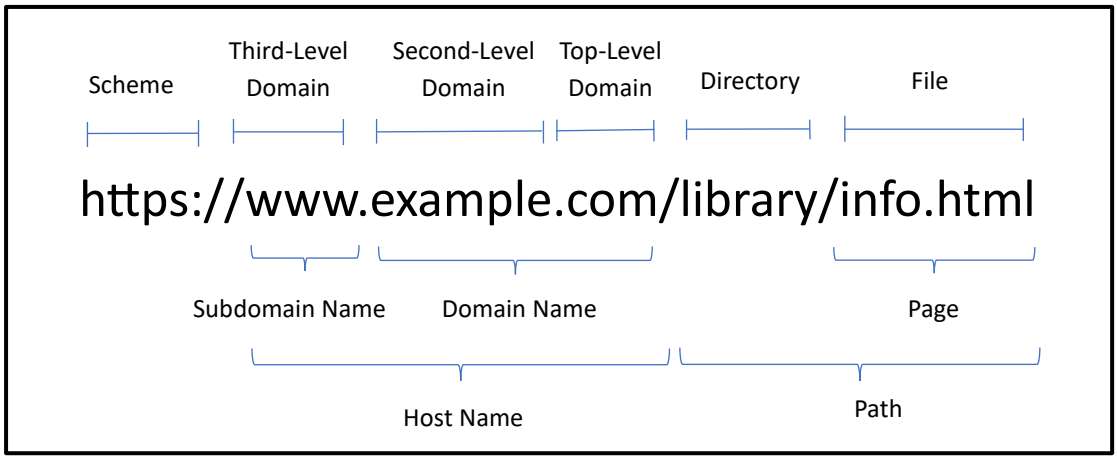
جدول 1:2 خلاصة الدراسة السابقة

المرجع	المؤلفون	السنة	المنهجية	مجموعة البيانات	القيود/التحديات	الدقة
[11]	Fu, A. Y.; Liu, W.; Deng, X.	2021	LightGBM(نموذج تدريج ضوئي معزز)	مواقع تصيد فعليّة (مميزاتURL)	يعتمد على استخراج ميزات من URL	~95%
[12]	Cao, Y.; Han, W.; Le, Y.	2018	مصادقة ثنائية + (محرك بحث) تحليل روابط	بيانات من محركات بحث وروابط تشابهية	يعتمد على محرك بحث خارجي وروابط HTML	~99.09%
[13]	Oest, A.; Safei, Y.; Doupe, A.; Ahn, G. J.; Wardman, B.; Warner, G.	2019	تجميع تخطيط الصفحة (SVM شجرة قرار AdaBoost, غابات عشوائية)	490صفحة تصيد (PhishTank)	استخراج ميزات CSS معقدة، يحتاج عينات كبيرة	~~96%
[14]	Sharifi, M.; Siadati, S. H.	2024	امتداد الميزات + BERT + CNN	مواقع تصيد في مجال العملات الرقمية	التعقيد الحسابي العالي، اعتماد على تمثيلاتBERT	أعلى من النماذج التقليدية
[15]	Zhang, Y.; Hong, J. I.; Cranor, L. F.	2022	اختيار ميزات + نماذج تجميع (RF (CatBoost, LGBM	مجموعات بيانات تصيد عامة (كورساتUCI)	حجم بيانات محدود، حساسية لاختيار الميزات	~97.4%
[16]	Xiang, G.; Hong, J.; Rose, C. P.; Cranor, L.	2023	CNNعلى تمثيلات صور للميزات (من URLs)	1,353 URL (مشروحة إلى 3 فئات)	مجموعة صغيرة من البيانات	86.5%
[17]	Keivanloo, I.; Roy, C. K.; Rilling, J.	2024	ميزات n-gram إحصائية + XGBoost & CNN	بيانات URLs مشروحة (PhishTank)	احتياج لمعالجة نصوص عشوائية بالاعتماد على-n grams	94.09%
[18]	Ozker, U.; Sahingoz, O. K.	2022	ميزات دلالية + مقارنة بين 16 نموذج تعلم آلي	عدد محدود من صفحات الويب	عدد محدود من الميزات الدلالية، قد يحتاج معرفية خاصة	—
[19]	Abdelnabi, S.; Krombholz, K.; Fritz, M.	2023	مراجعة منهجية لحلول بيانات التعدين	–	–	–
[20]	Chen, J. L.; Ma, Y. W.; Huang, K. L.	2023	VAE (المشفّر التبايني) + DNN	~100,000 URL(-ISCX 2016 + (Kaggle	تعقيد عالي (شبكات عميقة)، استجابة أسرع(~1.9s)	97.45%
[21]	Nair, S. M.	2024	NLP(ن-جرامات، ميزات لغوية) + ML	صفحات ويب خبيثة (URL + محتوى نصي)	كمية ميزات كبيرة، يعتمد على معالجة نص المحتوى	—
[23]	Sahingoz, O. K.; Buber, E.; Demir, O.; Diri, B.	2019	MLكلاسيكي (شجرة قرار، NB، SVM، ANN)	1,353 URL (مشروحة: حقيقية، مشبوهة، تصيد)	يعتمد على ميزات URL فقط، بيانات صغيرة	>90%
[24]	Alfouzan, N. A.; Narmatha, C.	2019	ML (تحليل JavaScript)	بيانات نصية لحملات خبيثة من الويب	لربما يحتاج استخراج قوي للسكريبت	—

[26]	Shah, B.; Dharamshi, K.; Patel, M.; Gaikwad, D.	2019	GBDT,)Stacking XGBoost, على (LightGBM ميزات URL وHTML	50K-PD, 50K-IPD (50 ألف صفحة)	يحتاج الوصول لمحتوى HTML وروابط	97.30%
[27]	Abiodun, O.; Sodiya, A. S.; Kareem, S. O.	2023	اختيار ميزات باستخدام تعبيرات نمطية ML + (DT, SVM, RF)	مجموعة من URLs Mendeley) (معدل dataset	يعتمد على استخراج دقيق للميزات عبرRegex	98% (RF)
[28]	Wu, J.; Yang, Z.; Guo, L.; Li, Y.; Liu, W.	2021	دراسة هجمات التهرب على نماذجML	–	يوضح هشاشة النماذج الحالية	—
[29]	Tang, L.; Mahmoud, Q. H.	2020	الكشف عن Smishing (SMS)	رسائل SMS	يدمج فحص محتوى SMS وتحليل URL وAPK	—
[30]	Yerima, S. Y.; Alzaylaee, M. K.	2024	مقارنة إعادة تشكيل البيانات (ROS, RUS, SMOTE) مع XGBoost	مجموعة بيانات تصيد (موازنة الفئات)	قلة الميزات الموزونة تؤثر على الأداء	—
[32]	Huang, Y.; Yang, Q.; Qin, J.; Wen, W.	2020	تجميع شبكات عصبية مدمجة	–	(تفاصيل غير متوفرة)	—
[33]	Mohammad, R. M.; Thabtah, F.; McCluskey, L.	2022	اختيار ميزات ارتباطي	–	(تفاصيل غير متوفرة)	—

5.2 تحليل URL

يعد تشريح عنوان URL أمرًا بالغ الأهمية لفهم كيفية تلاعب المهاجمين به لإطلاق هجمات التصيد. يوضح الشكل 1 بنية عنوان URL شرعي والمكونات المختلفة الموجودة داخله. يمكن للمهاجم التلاعب بأي جزء من أجزاء عنوان URL لإنشاء رابط خبيث يمكن استخدامه في تنفيذ هجوم تصيد.



الشكل 4:2 تفكيك الـ URL إلى مكوناته الأساسية

عنوان URL الخاص بأي موقع ويب يتكون من أربع مكوّنات رئيسية :المخطط (Scheme)، والمضيف (Host)، والمسار(Path) ، وسلسلة الاستعلام (Query String). يحدد المخطط البروتوكول المستخدم من قِبل عنوان URL، وأكثر البروتوكولات شيوعًا هما بروتوكول نقل النص التشعبي (HTTP) وبروتوكول نقل النص التشعبي الآمن (HTTPS). وعلى الرغم من أن HTTPS أكثر أمانًا، إلّا أنّ المهاجمين يستخدمونه الآن لجعل الرابط يبدو آمنًا. يشير المضيف عادةً إلى الخادم المستهدف الذي توجد فيه موارد التطبيق، ويمكن أن يتبعه رقم المنفذ (Port) الذي يستخدمه التطبيق للتواصل مع الخادم. يُقسَّم المضيف بدوره إلى ثلاثة أجزاء :النطاق الفرعي (Subdomain)، والنطاق من المستوى الثاني (Second-Level Domain)، والنطاق من المستوى الأعلى (Top-Level Domain). يمثّل النطاق الفرعي صفحة محددة ضمن الموقع، بينما يشير النطاق من المستوى الثاني إلى اسم الموقع الذي يتم الوصول إليه. أما النطاق من المستوى الأعلى فيُظهر نوع الكيان الذي تم تسجيل الموقع به على الإنترنت، ويُعتبر الامتداد ".com" الأكثر استخدامًا. بعد ذلك يُستخدم المسار (Path) لتحديد المورد المحدّد الذي يحاول عميل الويب الوصول إليه. وأخيرًا، قد يحتوي عنوان URL على سلسلة استعلام (Query String)، وهي سلسلة من المعلومات يمكن أن يستخدمها المورد للحصول على بيانات محددة من الخادم. يُعد جزء الاستعلام هذا اختياريًا، وليس جميع عناوين URL تحتوي عليه.

يُعدّ فهم المكوّنات المختلفة لعنوان URL أمرًا مهمًا لأنه غالبًا ما يكون نقطة البداية لهجمات التصيّد. يحاول المهاجم استغلال الهندسة الاجتماعية لخداع الضحية بحيث يمر الرابط الخبيث دون أن يلاحظه المستخدم أو حتى خوارزميات الكشف. قبل تنفيذ هجوم التصيّد، يأخذ المهاجم وقته لتجربة العديد من التركيبات والتلاعبات في عنوان URL حتى لا يثير الرابط الاحتيالي أي شكوك. ولتحقيق هذا الهدف، يستخدم المهاجم تقنيات تمويه (Obfuscation) متنوعة.

على سبيل المثال، قد يُموّه المهاجم اسم المضيف باستخدام عنوان IP ويضع اسم الموقع المستهدف في المسار، مثل http://159.203.6.191/servicepaypal/ كما يمكنه تمويه المضيف باستخدام اسم نطاق آخر يبدو شرعيًا، مثل http://a0243562.xsph.ru/servicePayPal/C/ حيث يحتوي اسم المضيف على نطاق يبدو صالحًا، لكن الموقع الاحتيالي موجود في المسار. يمكن أيضًا التمويه باستخدام اسم نطاق طويل مثل https://paypalhelpservice.simdif.com/ حيث يظهر اسم الجهة المستهدفة (PayPal) في النطاق الفرعي (Subdomain) للرابط. بالإضافة إلى ذلك، قد يستخدم المهاجم اسم نطاق مجهولًا أو مكتوبًا بشكل خاطئ مثل: http://paypa1.com حيث تم استبدال الحرف **l** بالرقم **1** ليبدو مشابهاً للنطاق الشرعي. جميع هذه الأساليب تهدف إلى إعادة توجيه الضحية إلى موقع خبيث وإغرائه بتقديم معلومات حساسة يستغلها المهاجم لاحقًا. في جميع الأمثلة، يظهر اسم الشركة المستهدفة (PayPal) في الرابط الخبيث بشكل أو بآخر، مما قد يصعّب على المستخدم التمييز بين الموقع الاحتيالي والموقع الشرعي. عادةً ما يُطلق المهاجم هذه الهجمات بخلق شعور بالإلحاح لدى الضحية، مما يزيد التوتر ويجعلها عرضة للأخطاء في الحكم. لذلك، ولمنع هجمات التصيّد الاحتيالي المعتمدة على عناوين URL، يلزم اتباع نهج آلي للكشف عنها.

6.2 استخراج الميزات

1.6.2 الميزات المعجمية (Lexical Features)

تعتمد الميزات المعجمية على تحليل نص عنوان الـ URL نفسه دون الرجوع لمحتوى الصفحة أو قواعد البيانات الخارجية من الأمثلة الشائعة:

- طول الرابط :(URL Length) طول السلسلة النصية للـ URL كاملاً. غالباً ما تستخدم الروابط الخبيثة عناوين طويلة أو معقدة لإخفاء محتواها.
- وجود عنوان IP بدلاً من اسم نطاق :إذا كان الرابط يحتوي على عنوان IP (مثل https://192.168.1.1/...) فإن ذلك يعد علامة مريبة لأن مواقع التصيد تحاول أحياناً إخفاء هويتها باستخدام IP مباشر.
- عدد النقاط والنطاقات الفرعية: عدد أقسام النطاق الفرعي في الـ URL (كمثال sub.sub2.example.com فيه مستويان فرعيان). عدد كبير من النقاط أو مستويات فرعية غير معتادة يرتبط غالباً بروابط تصيد تحاول تقليد مواقع موثوقة بكتابة أسماء فرعية طويلة.
- وجود رموز خاصة أو عناصر نصية مشبوهة :مثل الرمز (@) أو (-) أو (_) أو المسار المضاعف (//) داخل العنوان. على سبيل المثال، يحذف المتصفح أي شيء قبل (@) في الرابط، مما يستغله المهاجم لإخفاء الموقع الحقيقي. كذلك توجد دراسات تشير إلى أن روابط التصيد غالباً ما تحتوي على كلمات مثل "login" أو "secure" لجذب المستخدمين.

- خدمات اختصار الروابط :استخدام منصات اختصار الروابط (مثل bit.ly) قد يخفي عنوان URL الحقيقي. وتظهر ميزة shortining_service في بيانات UCI الشهيرة كميزة معجمية، إذ يميل المهاجمون إلى اختصار الروابط لتسهيل توزيعها دون الكشف عن مقصدها الحقيقي.
- امتداد النطاق (TLD) ووجود HTTPS في العنوان :مثلاً روابط تحتوي على نص "https" في اسم النطاق نفسه (http://https://site.com) أو استخدام ملحقات نطاق غير شائعة (.ru, .xyz, ...) قد تكون مؤشراً إضافياً، خاصة إذا تعارضت مع العلامة التجارية.

كل هذه الميزات تحلل بنية الرابط ودلالاته النصية لكشف الشذوذ: فالرابط الطويل والمليء بالنقاط والرموز غير المألوفة، أو استخدام عناوين IP مباشرة وخدمات الاختصار، كلها تغيرات مميزة عن روابط المواقع الموثوقة.

2.6.2 ميزات معتمدة على النطاق(Domain-Based Features)

تتعلق ميزات النطاق بمعلومات تسجيل أو ملكية النطاق المرتبط بالـ URL من قاعدة بيانات الـ WHOIS وغيرها. من الأمثلة:

- عمر النطاق (Age): عمر النطاق منذ تسجيله. غالباً ما تسجل مواقع التصيد أسماء نطاق حديثة أو قصيرة العمر، لأن النطاقات المغشوشة تُستخدم لفترة قصيرة وسرعان ما تغلق.
- مدة التسجيل المتبقية وطول فترة التسجيل: وهي مقياس لمدى المدة التي تم دفعها للنطاق. عادةً النطاقات المضمونة طويلة الأجل بينما التصيديات تسجل لوقت قصير فقط.
- معلومات شهادة :SSL مدى صحة شهادة HTTPS للموقع. مثلاً الشهادة ذاتية التوقيع أو من مصدر مشبوه يعد مؤشراً للتصيد. إذا كان اسم الجهة المصدرة للشهادة (CA) نفسه كاسم مالك النطاق، فقد يكون دليلاً على شهادة مزيفة.
- معلومات WHOIS عامة :(مثل مزود التسجيل، امتداد البلد). استخدام خدمة خصوصية WHOIS أو تسجيلات بأسماء خادعة قد يشير لرباط تصيدي.
- مطابقة اسم النطاق مع العلامة التجارية: قياس تشابه اسم النطاق مع اسم العلامة التجارية المستهدفة. فمثلاً، نطاق بعيد عن الاسم الحقيقي للعلامة أو يضيف كلمات غير متوقعة غالباً ما يستخدمه المحتالون.
- موقع استضافة النطاق جغرافيًا :وجود استضافة في بلدان غير متوقعة بالنسبة للموقع الشرعي أو استخدام شبكات CDN معروفة قد يدل على تصيد.

تسهم هذه الميزات في تحديد عدم مصداقية النطاق؛ فعمر النطاق القصير والشهادة غير الصحيحة ومعلومات التسجيل الملتوية كلها معايير حديثة معتمدة في الأبحاث لكشف التصيد.

3.6.2 ميزات معتمدة على الشبكة (Network-Based Features)

تشمل هذه الميزات معلومات شبكية أو خارجية متعلقة بالـ URL والنطاق:

- سجل DNS ومعلومات IP (Feature: DNS record): مثلاً عدد سجلات DNS المرتبطة بالنطاق أو سرعة استجابتها. قد تشير سجلات DNS عديدة أو متغيرة بسرعة (DNS Fast Flux) لمحاولات اختراق.
- الربط بخدمات خارجية :مثلاً إذا كان الرابط يقع في قاعدة بيانات سوداء (مثل PhishTank/OpenPhish) أو خوادم تعتبر مشبوهة فإن ذلك يشير إلى تصيد. كما يشمل هذا فحص وجودات للنطاق في محركات البحث أو فهرس) Google ميزة.

- رقم النظام المستقل (ASN) وموقع الخادم :أحياناً يُنظر إلى شركة الاستضافة أو موقع الخادم؛ على سبيل المثال، استخدم العديد من مواقع التصيد خوادم «سحابية» مجهولة أو مواقع في دول بعيدة.

باختصار، أي ميزة تتعلق بالشبكة مثل معلومات DNS أو حركة المرور أو ترتيب الموقع تُساعد على تمييز الروابط المشبوهة من الموثوقة.

يعتمد الباحثون غالباً على مجموعات بيانات مفتوحة لدراسة هذه الميزات. من الأمثلة المعتمدة :مجموعة UCI Phishing Website (2012) التي تضم 11,055 رابطاً مصنفة و30 ميزة مثل وجود عنوان IP وطول الرابط ووجود @ وغير ذلك. ومجموعة Hannousse (2021) التي تتضمن 11,430 عنوان URL مع 87 ميزة مصنفة (56 معجمية، 24 محتوى، 7 من خدمات خارجية). كما تستخدم قواعد بيانات مثل (*PhishTank* و*OpenPhish*) لإثراء بيانات التدريب. وقد أظهرت الأبحاث الحديثة كفاءة عالية لاستخدام هذه الميزات في نماذج تعلم آلي متطورة (مثل XGBoostو Deep Learning) لكشف الروابط الخبيثة

7.2 دور الذكاء الاصطناعي وتعلم الآلة في كشف التصيد عبر الروابط

ساهمت تقنيات الذكاء الاصطناعي والتعلم الآلي في تحسين اكتشاف روابط التصيد بشكل كبير من خلال بناء نماذج تصنيف قادرة على التنبؤ بكون الرابط خبيثا أم لا اعتمادا على بيانات موسومة سابقة. يعتبر كشف التصيد مسألة تصنيف ثنائي خاضع للإشراف، حيث تدرب النماذج على بيانات من الروابط (مصنفة على أنها *تصيد* أو *شرعية*) لاستخلاص الأنماط والسمات المميزة لكل فئة. ومن الخوارزميات الشائعة في هذا المجال غابات القرار العشوائية (Random Forest)، وآلات الدّعم الناقلة (SVM)، والشبكات العصبية الاصطناعية، بالإضافة إلى خوارزميات بسيطة أخرى مثل نايف بايز (Naive Bayes)، وشجرة القرار (Decision Tree)، والقريب النائي (KNN). تعلم النماذج هذه الأنماط بهدف تحقيق دقة تنبّؤ عالية وتقليل الإنذارات الكاذبة؛ وقد أظهرت دراسات أن غابات القرار العشوائية عادةً ما تتفوق في الأداء على أشجار القرار الفردية لتحقيق دقة أعلى. والأهم أن النماذج المدربة قادرة على تمييز روابط التصيّد الجديدة التي لم تكن موجودة في القوائم البيضاء الثابتة، مما يُعالج عيبًا أساسيًا في الأساليب التقليدية القائمة على القوائم الثابتة.

1.7.2 الخوارزميات الشائعة المستخدمة

1.1.7.2 تعلم الالة

- غابات القرار العشوائية (Random Forest): تجمع هذه الخوارزمية عدداً من أشجار القرار لتقليل خطر الإفراط في التعميم (overfitting). عند تصنيف رابط ما، تدلي كل شجرة برأيها ثم تجمع القرارات (إما بالتصويت أو بالمتوسط)، مما ينتج نموذجاً أكثر استقراراً ودقة.

- شجرة القرار (Decision Tree): تبني الشجرة نموذجًا شجريًّا من قواعد شرطية تُقسّم بيانات التدريب إلى مجموعات متجانسة بناءً على ميزات مفصلة (مثل طول الـ URL أو وجود كلمات معينة). كل عقدة داخل الشجرة تمثل اختبارًا لميزة، والأوراق تمثل تصنيفًا نهائيًا؛ الشجرة سهلة الفهم والتفسير لكنها قد تتعرّض للإفراط في التعلّم إذا لم تُقَطع أو تُنظّم بشكل مناسب. تستخدم كـ baseline مفيدة ولها قابلية تفسير عالية عند شرح قرارات الكشف.

- نايف بايز (Naive Bayes): يعتمد على قاعدة بايز لحساب احتمال انتماء العينة لكل فئة بافتراض استقلالية الميزات (تبسيط قوي). سريع التنفيذ وفعّال لميزات نصية (مثل خصائص الحروف في URL) خاصة عند نقص البيانات، لكنه قد يفشل إذا كانت الميزات مرتبطة ارتباطًا قويًا بحيث يكسر فرضية الاستقلال.

- الانحدار اللوجستي (Logistic Regression): نموذج إحصائي يبني علاقة خطية بين الميزات واحتمال انتماء العينة للفئة الإيجابية (تصيّد)، ثم يطبّق دالة لوجستية لإخراج احتمالية بين 0 و1. بسيط وقابل للتفسير، يعمل جيدًا عندما تكون الحدود بين الفئات تقارب الخطية أو بعد تحويل الميزات، ويُستخدم كثيرًا كخط أساس لقياس أداء خوارزميات أعقد.

- آلة الدعم الناقلة (SVM (Support Vector Machine: تعمل على إيجاد مستوى فصل (hyperplane) في فضاء متعدد الأبعاد يفصل بين فئتين (تصيد/مشروع) بأكبر هامش ممكن بينها. يناسب SVM البيانات الخطية ثنائية التصنيف ويستخدم في كثير من الأنظمة نظرًا لقوته في فصل الفئات عند توفر مميزات واضحة.

- الجيران الأقربون (K-Nearest Neighbors — KNN): يصنف عينة جديدة بناءً على تصنيفات أقرب k عينات في فضاء الميزات (التصويت بالأغلبية). بسيط وغير متطلب للتدريب(lazy learning) ، لكنه يتطلب حساب المسافات لكل استعلام لذا يمكن أن يكون بطيئًا في التشغيل على مجموعات كبيرة، ويعتمد بشكل رئيسي على مقياس المسافة واختيارk.

- تعزيز التدرج (مثل XGBoost/LightGBM — Gradient Boosting): تجمع عدة نماذج ضعيفة (عادة أشجار قرار ضحلة) بشكل تتابعي، حيث يحاول كل نموذج تصحيح أخطاء سابقه عبر تعلم المتبقيات (residuals). ينتج عادةً نموذجًا قويًا ودقيقًا للغاية، مع قدرة جيدة على التعامل مع ميزات غير خطية وتفاعلاتها؛ يُستخدم على نطاق واسع في مسابقات ومهمات تصنيف عملية، لكن ضبط معاييره قد يتطلب خبرة وموارد حسابية.

- الشبكات العصبية متعددة الطبقات (Multilayer Perceptron — MLP): نموذج عميق/كلاسيكي يتألف من طبقات مترابطة من الخلايا العصبية الاصطناعية التي تتعلّم تمثيلات غير خطية للبيانات عبر الانحدار العكسي(backpropagation) . مناسب لاكتشاف أنماط معقدة في الميزات المركبة، ويتطلب ضبط معماريات ومعاملات تدريبية، كما يحتاج عادة بيانات كبيرة نسبيًا.

2.1.7.2 التعلم العميق المتخصص:

- شبكات التلافيف (CNN): تستخدم لاستخلاص ميزات مكانية أو محلية (مفيدة عند معالجة لقطات شاشة للصفحات أو تمثيلات مصفوفية للنص).
- الشبكات العودية / LSTM (RNN): مناسبة لمعالجة سلاسل أحرف الـ URL أو الحقول النصية حيث يعتمد تسلسل الحروف على المعنى.
- المحولات (Transformers): نماذج حديثة قوية لمعالجة النصوص (حتى سلاسل قصيرة كالـ URL) قادرة على التقاط علاقات بعيدة النطاق داخل السلسلة.

تظهر النماذج العميقة أداءً ممتازًا على مهام الاستخلاص التلقائي للميزات لكن تكلفتها الحسابية والتفسيرية أعلى.

2.7.2 أنواع الميزات المستخدمة

تعتمد نماذج كشف التصيد على مجموعة من الميزات المستخرجة من روابط الـ URL ومحتوى الصفحة. من أهم هذه الميزات:

- سمات معجمية (Lexical Features): مثل طول الرابط، عدد النقاط (.) أو الشرط (–)، استخدام البروتوكول (http/https)، وجود أحرف أو كلمات مشبوهة في العنوان، كما تندرج تحتها خصائص شريط العنوان مثل احتواء الرابط على رمز "@" أو على رابط مختصر. تُمثّل هذه السمات أساساً قوياً في كشف التّلاعب بعنوان الرابط.
- سمات غير اعتيادية (Abnormal Features): تشمل مؤشرات إضافية مثل وجود تجاوزات (redirects) أو فقدان أجزاء متوقعة من الرابط، مما يوحي بتغيير هيكلية الرابط. غالبًا ما يُدرج في هذه الفئة تضمين روابط مشبوهة ضمن نص الرابط ذاته أو على صفحة الويب.
- سمات متعلقة بمحتوى الموقع (Content-based Features): تستخلص من شيفرة الصفحة HTML/JavaScript، مثل وجود عناصر مشبوهة (iframes، أو سكريبتات تتلاعب بالبيانات)، أو تشابه تصميم الصفحة مع مواقع موثوقة. تساعد هذه السمات على رصد التزييف في مستوى المحتوى.
- سمات إحصائية وشبكية (Statistical/Network Features): مثل عمر النطاق من سجلات WHOIS وتصنيف الموقع ترتيب Alexa، حيث يمكن أن يدل عمر نطاق قصير أو ترتيب عال على احتمالية وجود تصيد
- سمات نصية (Character-level Features): تستفيد بعض النماذج من تمثيلات إحصائية لنص الرابط نفسه، كاحتساب TF-IDF على مستوى الأحرف لكل رابط. تعطي هذه المعالجة تمثيلاً عدّديًا يظهر مدى ندرة أو شيوع الحروف داخل الروابط الاحتيالية مقارنة بالشرعية.

3.7.2 استخراج البيانات وتدريب النماذج

تبنى نماذج التعلم الآلي على بيانات كبيرة من الروابط. تستخدم الأبحاث مجموعات بيانات منشورة مثل مجموعة بيانات UCI (11,055 رابطًا مع 30 سمة) ومجموعة Mendeley (10,000 رابط مع 48 سمة) والمجموعات التي نشرتها مشاريع الأمن الإلكتروني المختلفة. بالإضافة إلى ذلك، يجمع الباحثون روابطاً خبيثة مباشرةً من خدمات متخصصة (PhishTank، OpenPhish، Spamhaus، وما شابه) وروابط شرعية من قواعد بيانات مثل Alexa وCommon Crawl. بعد تجميع الروابط، يستخلص لكل رابط السمات (كما سبق) ويوسم كـ(تصيد/شرعي). ثم تُقسَّم البيانات إلى مجموعتي تدريب واختبار، عادةً بنسبة 80% للتدريب و20% للاختبار. وقد يستخدم الباحثون التصديق المتقاطع (Cross-Validation) بطرق مثل 5-fold أو 10-fold لتقدير الأداء بشكل أكثر موثوقية. بعد ذلك يتم تدريب النموذج على مجموعة التدريب وضبط معاييره (كبُعد الأشجار أو معاملات الشبكة العصبية)، ثم يُقيّم على مجموعة الاختبار. ثمة تقنيات لمعالجة عدم التوازن في البيانات مثل إعادة التوازن (Resampling) أو خوارزميات خاصة لضبط الأولويات، نظرًا لأن عدد روابط التصيّد عادةً أقل من الشرعية.

4.7.2 آليات التقييم

يقاس أداء النماذج بعدة مؤشرات رئيسية، من أبرزها الدقة (Accuracy) والحساسية (Recall) والدقة النوعية (Precision)، وأحيانًا معيار F1. فالدقة تمثل نسبة التنبؤات الصحيحة إلى إجمالي الحالات، بينما الحساسية تعبر عن نسبة روابط التصيّد الحقيقية التي تم اكتشافها بنجاح، حسب المعادلة (1) التالية:

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

وتعرف الدقة النوعية بأنها نسبة الروابط التي تم التنبؤ بأنها تصيد والتي كانت فعلاً تصيد، حسب المعادلة (2)

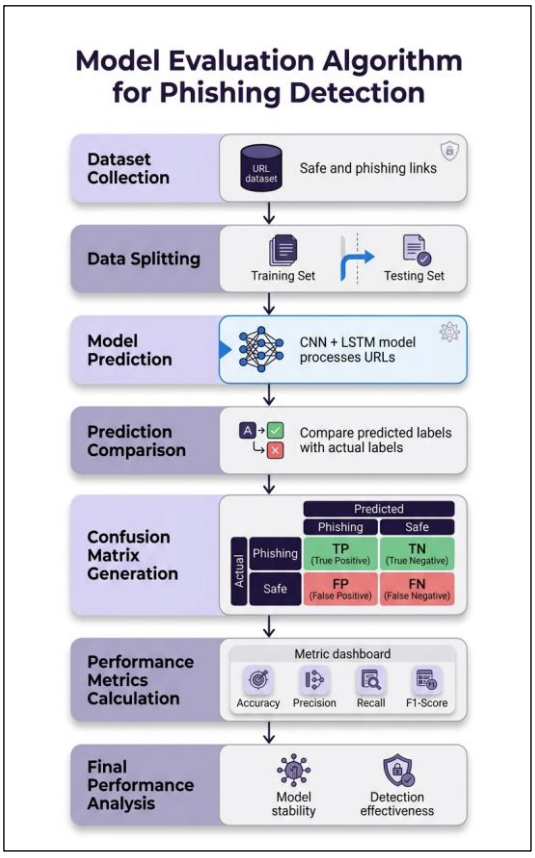
$$Precision = \frac{TP}{TP + FP} \quad (2)$$

اما الحساسية تظهر مدى قدرة النموذج على كشف جميع الروابط الاحتيالية، حسب المعادلة (3) التالية:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

أما معيار F1 هو المتوسط التوافقي بين الحساسية والدقة النوعية، ويستخدم عندما نريد موازنة الاثنين خصوصًا في حالة البيانات غير المتوازنة، حسب المعادلة (4) التالية:

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



الشكل 5:2 مخطط خوارزمية تقييم أداء النموذج

ويعد انخفاض معدل الإنذارات الكاذبة (False Positive Rate) مهما في أنظمة الكشف، إذ إن زيادة إنذارات كاذبة كثيرة تؤدي إلى إحباط المستخدمين وتقليل ثقتهم بالنظام. لأجل حساب هذه المؤشرات، يقسم الباحثون عادةً البيانات إلى مجموعات تدريب واختبار (كما ذُكر) ويحسبون القيم الأساسية: صحيح إيجابي (TP)، صحيح سلبي (TN)، خاطئ إيجابي (FP)، وخاطئ سلبي (FN). كما يستخدم التصديق المتقاطع للمساعدة في التأكد من ثبات الأداء على عينات غير مرئية.

TP = عدد الروابط الاحتيالية التي تم تصنيفها بشكل صحيح كاحتيالية.

TN = عدد الروابط الشرعية التي تم تصنيفها بشكل صحيح كشرعية.

FP = عدد الروابط الشرعية التي صنفت خطأً كاحتيالية.

FN = عدد الروابط الاحتيالية التي صنفت خطأً كشرعية.

1.4.7.2 مثال توضيحي لمقاييس التقييم:

لتوضيح كيفية عمل مقاييس التقييم السابقة بشكل عملي، يمكن عرض مثال مبسط يوضح حالات التصنيف المختلفة التي قد ينتجها النموذج.

فعلى سبيل المثال، عند إدخال رابط معين إلى النظام، قد يقوم النموذج بتصنيفه على أنه رابط احتيالي(Phishing) ، بينما يكون في الحقيقة رابطًا آمنًا، وتمثل هذه الحالة ما يُعرف بالإيجابي الكاذب(False Positive - FP) ، وهي من الحالات التي يجب تقليلها لتجنب إزعاج المستخدم.

وفي حالة أخرى، قد يتم تصنيف رابط احتيالي على أنه آمن، وهو ما يُعرف بالسلبي الكاذب(False Negative -FN) ، وتُعد هذه الحالة أكثر خطورة لأنها قد تؤدي إلى تعرض المستخدم لهجمات تصيد فعلية.

وبالتالي، فإن تحليل هذه الحالات يساعد في فهم أداء النموذج بشكل أعمق، وتحديد نقاط القوة والضعف فيه، مما يساهم في تحسين دقته وموثوقيته.

5.7.2 مزايا التعلّم الآلي مقابل الطّرُق التقليدية

توفر أساليب التعلم الآلي مزايا مهمة مقارنة بالطرق التقليدية المبنية على قواعد صريحة أو قوائم تصنيف ثابتة. فهي قادرة على التعامل مع روابط تصيد جديدة وغير معروفة سابقًا بكفاءة أكبر، إذ يستنبط النموذج الأنماط من البيانات نفسها بدلاً من الاعتماد على تحديث يدوي للقوائم السوداء/البيضاء. وبذلك يمكن للنموذج التنبؤ باحتمالية أن يكون الرابط خبيثًا حتى لو لم يكن قد واجه مثيلاً له من قبل. بالإضافة إلى ذلك، يسمح التعلّم الآلي بمعالجة عدد ضخم من السمات وتحليلها دفعة واحدة، مما يزيد من دقّة التصنيف. كما أن نماذج التعلم العميق (CNN، RNN) المستندة إلى نص الرابط تتيح استخلاص معلومات من الـ URL دون الحاجة إلى خدمات خارجية أو خبرة أمنية مسبقة. بالمقابل، تعتمد الأساليب التقليدية على قواعد جامدة وقوائم محدودة، وقد تظل عاجزة عن مواكبة تقنيات المهاجمين الجديدة (مثل روابط تصيّد مبتكرة أو غير متوقعة).

6.7.2 التحديات المحتملة

على الرغم من المزايا، تواجه أنظمة التعلم الآلي تحديات فريدة. من أبرزها الهجمات الجديدة (Zero-day): إن الروابط الاحتيالية التي تظهر لأول مرة لن يكون لها أمثلة سابقة في بيانات التدريب، مما يصعب اكتشافها دون تحديث النموذج أو توافر بيانات تدريب حديثة وذات جودة عالية. كذلك يعاني الكثير من المجموعات التدريبية من عدم التوازن في البيانات، حيث تكون الأمثلة المخادعة أقل عدداً من الشرعية؛ وقد يؤدي هذا إلى تحيز النموذج نحو الفئة الأكبر (الشرعية) مع تحقيق دقة ظاهرية عالية ولكنه يفشل في اكتشاف الجزء الصغير من الهجمات. جانب آخر متعلق بالأمان هو التفسير (Explainability): فالنماذج المعقدة كالشبكات العصبية غالبا ما توصف بأنها "صندوق أسود" يصعب تفسير لماذا قررت اعتبار رابط ما خبيثا. هذا الأمر يضعف القدرة على فهم سلوك النموذج ومبرراته، مما يقلل من قابليته للتبني في حالات تتطلب الشفافية. إضافة إلى ذلك، يواصل المهاجمون تطوير تقنيات التهرب (مثل استخدام روابط قصيرة أو تقنيات تشويش/تعتيم Fast-Flux) التي يمكن أن تطمس السمات المميزة للروابط المخادعة وتقلل من فعالية النماذج القائمة. أخيرا، تتطلب بعض الخوارزميات المتقدمة كالشبكات العميقة موارد حسابية كبيرة وكمية هائلة من البيانات للتدريب، مما قد يمثل تحديًا عمليا في تطبيقها في أنظمة في الوقت الحقيقي.

الفصل الثالث

منهجية العمل وخطة المشروع

1.3 مقدمة عن منهجية Agile

تُعد Agile منهجية تطوير مرنة تُستخدم في إدارة وتنفيذ المشاريع التقنية من خلال تقسيم العمل إلى مراحل قصيرة ومتكررة تُعرف باسم (Sprints) ، بحيث يتم خلال كل مرحلة التخطيط والتنفيذ والتقييم بصورة مستمرة.تهدف هذه المنهجية إلى تعزيز القدرة على التكيف السريع مع التغييرات وتحسين جودة المنتج بشكل تدريجي، مما يجعلها مناسبة للمشاريع التي تتطلب تطويرا مستمرا واستجابة فعالة للملاحظات والتحديات المستجدة. وفي هذا المشروع، تم اعتماد Agile نظرا لملاءمتها لطبيعة مشاريع التعلم الآلي، حيث تسمح ببناء نماذج أولية واختبارها بشكل دوري، مع تحسين الأداء في كل دورة تطوير، مما يساهم في الوصول إلى نظام أكثر كفاءة ودقة.

2.3 مراحل منهجية إعداد النموذج:

Sprint 0: إعداد المشروع وبيئة العمل

الهدف: تجهيز بيئة التطوير، وثائق الخطة.

المهام التفصيلية:

- تثبيت الأدوات الأساسية (Python، Jupyter، pytorch، scikit-learn ،Pandas).
- إعداد قائمة مصادر البيانات (PhishTank, Mendeley, HuggingFace، Top-sites).

Sprint 1: جمع وتجميع البيانات

الهدف: الحصول على مجموعة بيانات متوازنة قدر الإمكان من روابط تصيد وروابط شرعية.

المهام:

- تنزيل dumps من PhishTank (CSV/JSON).
- تنزيل/جمع مجموعات من Kaggle وHugging Face وMendeley.
- جمع قائمة روابط شرعية من قوائم Top-Sites (Majestic/Tranco).
- توحيد صيغ الملفات (CSV/JSON) ودمجها في ملف واحد خام (raw_dataset.csv).

Sprint 2: المعالجة المسبقة واستخراج الخصائص الأولية

الهدف: تنظيف البيانات، فك الاختصارات (short URL expansion)، واستخراج السمات البنيوية الأولية.

المهام:

- إزالة duplicates وURLs الفارغة.
- التحقق من صلاحية روابط عينة (HTTP status) وتوثيق الحالات.
- كشف الروابط المختصرة (قائمة shortener domains) ومحاولة فك الاختصار آمنًا مع تسجيل expanded_url, redirect_count, expansion_success.
- حفظ مجموعة معالجة (processed_dataset.csv) مع الأعمدة الجديدة.

Sprint 3: التحليل الاستكشافي للبيانات

الهدف: فهم سلوك البيانات بعد جمعها وتنظيفها، واكتشاف الأنماط والعلاقات التي تساعد على تحسين أداء النموذج الذكي (CNN + LSTM).

المهام:

- التحقق من توازن الفئات، تحليل نسبة الروابط الآمنة إلى الروابط الاحتيالية بعد التنظيف.
- تحليل السمات الإحصائية العامة
- تحليل الكلمات المفتاحية داخل الروابط
- تحليل البنية الهيكلية للروابط (Structural Patterns)
- اكتشاف القيم الشاذة تحديد الروابط ذات أطوال أو رموز غير معتادة جدًا.
- تجهيز البيانات لمرحلة التسلسلات (Sequence Preparation)
- بعد التحليل، تحديد الأعمدة التي ستستخدم لإنشاء تسلسلات إدخال للنموذج.

Sprint 4: بناء النموذج CNN + LSTM

الهدف: تنفيذ نموذج هجيني وظيفي (Prototype) يجمع بين CNN وLSTM.

المهام:

- تحويل الروابط إلى تمثيل رقمي (char-level tokenization، padding).
- تصميم بنية النموذج: Conv1D ← Embedding (متعدد النوى) ← MaxPooling ← Sigmoid ← Dense ← LSTM .
- تسجيل سجلات التدريب (loss/accuracy curves).

Sprint 5: ضبط المعاملات واختبارات العزل (Tuning & Ablation)

الهدف: تحسين أداء النموذج عبر ضبط hyperparameters وإجراء اختبارات ablation توضح قيمة كل مكوّن (CNN فقط، LSTM فقط، CNN+LSTM).

المهام:

- تجربة قيم مختلفة لـ embedding_dim, filters, lstm_units, learning_rate.
- تجربة تقنيات التعامل مع imbalance (class_weight, oversampling).
- اختيار أفضل إعداد وفقًا لمقاييس متعددة (Precision/Recall/F1/AUC).

Sprint 6: تقييم عام، اختبار التعميم، والتوثيق النهائي, إعداد الوجهة النهائية

الهدف: تقييم النموذج على بيانات خارجية، إعداد النتائج النهائية، وتجميع فصل النتائج والملاحق.

المهام:

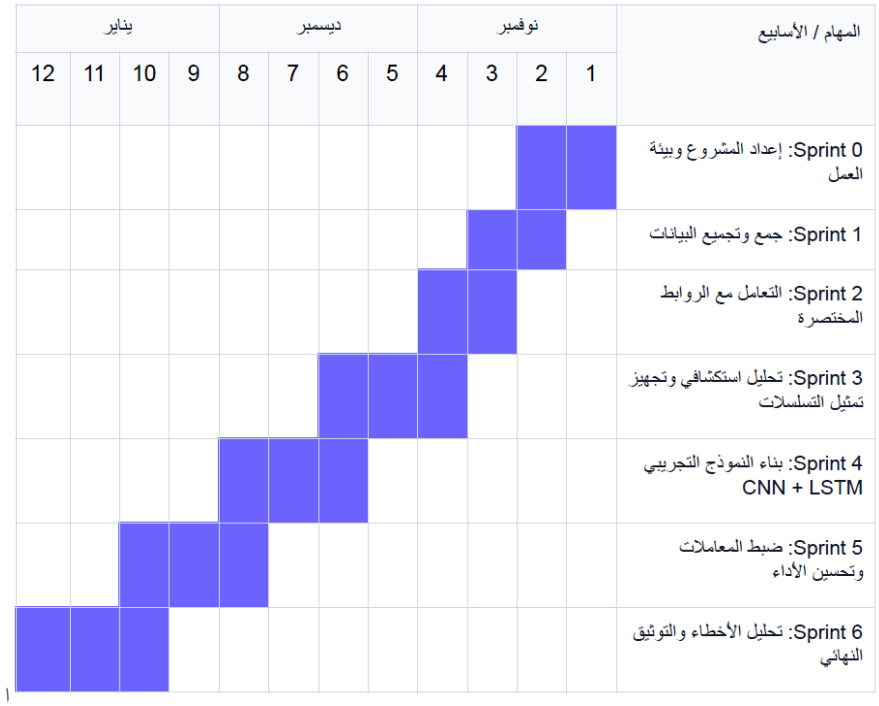
- إعداد واجهة الويب
- تقييم النموذج على hold-out set مستقل ومن مصدر آخر (generalization test).
- تحليل الأخطاء (Error Analysis): ما أنواع الروابط التي تفشل؟ short URLs؟ domains محددة؟
- إعداد فصل النتائج (الجداول، مصفوفة الالتباس، ROC curves).
- توثيق كاملة: كود الملاحق، وصف datasets، توضيح سياسات الـ expansion وأخلاقيات البيانات.
- تحضير العرض التقديمي (Slides) ونسخة Word من الفصل الثالث/الرابع.

3.3 دراسة الجدوى

1. **الجدوى المالية:** يعتمد المشروع على أدوات وبرمجيات مفتوحة المصدر (Open-Source)، مثل لغة (Python) ومكتبات (Pytorch) و(Pandas) و(NumPy)، بالإضافة إلى مجموعات بيانات متاحة مجانًا عبر منصات مثل Kaggle. لذلك لا تتطلب الدراسة أي تكاليف مالية كبيرة، ويمكن تنفيذها باستخدام حاسب شخصي مجهز بموارد متوسطة.
2. **الجدوى التقنية:** تتوفر جميع الأدوات اللازمة بشكل مجاني ومستمر التحديث، مما يسهل تنفيذ المشروع دون عناء الحصول على تراخيص باهظة. على سبيل المثال، تتيح مكتبة (pytorch) بناء نماذج تعلم عميق (Deep Learning) بسهولة وتقديم نماذج قابلة للتشغيل على أي بيئة.
3. **الجدوى الزمنية:** يمكن إكمال مراحل المشروع ضمن فصل دراسي واحد باتباع خطة زمنية منظمة. إذ يسمح تقسيم العمل إلى مراحل مستقلة (جمع بيانات، معالجة، تصميم نموذج، تدريب، تقييم، توثيق) بإتمام كل منها في وقت محدد، مع إمكانية توازي بعض الخطوات إذا لزم الأمر.
4. **الجدوى الأكاديمية:** يكتسب موضوع الكشف عن التصيد الاحتيالي (Phishing Detection) أهمية كبيرة في الأدبيات البحثية والأمنية. فالتصيد يمثل أحد أشكال الجرائم الإلكترونية الفعالة وواسعة الانتشار، وغالبًا ما يؤدي إلى سرقة معلومات شخصية وحساسة. وُثّق أن التصيد يُعتَبَر من بين أكثر أشكال الاحتيال شيوعًا على الإنترنت، مما يستلزم البحث الأكاديمي في طرق مبتكرة لكشفه والتصدي له. لذلك فإن تنفيذ هذا المشروع يسهم في تطوير أدوات ذكاء اصطناعي قادرة على مواجهة مشكلة حيوية وتحمل قيمة تطبيقية وعلمية.

4.3 المخطط الزمني للمشروع

تم تقسيم المشروع إلى مراحل رئيسية موزّعة زمنياً لتحقيق الإنجاز في فترة الدراسة (قرابة ثلاثة أشهر):



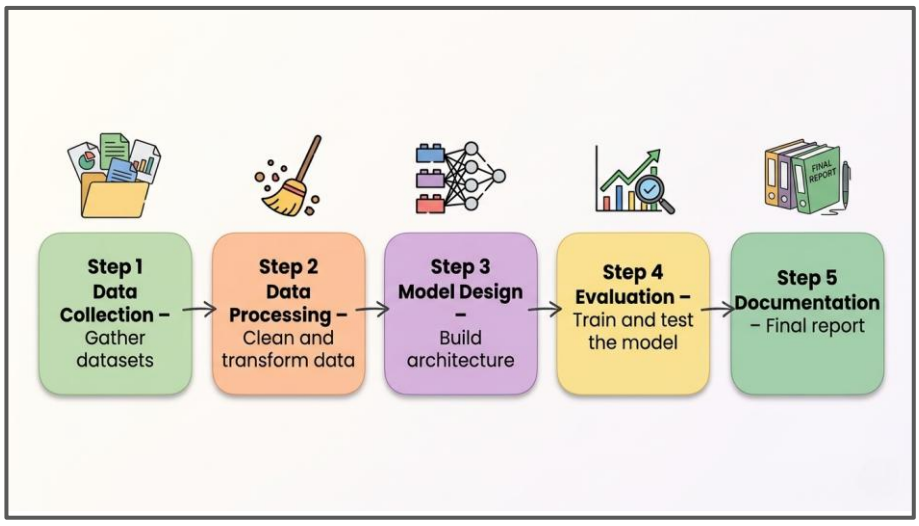
الشكل 1:3 رسم توضيحي للمخطط الزمني

5.3 أدوات وتقنيات المشروع

- **لغة البرمجة:** Python؛
- **مكتبة التعلم العميق:** pytorch؛ لبناء وتدريب الشبكة العصبية العميقة.
- **مكتبات تحليل البيانات:** Pandas وNumPy؛ لتنظيم ومعالجة البيانات (Data Manipulation).
- **بيئة العمل:** Jupyter Notebook؛ لكتابة التعليمات البرمجية التفاعلية وتنظيم سير العمل.
- **مجموعات البيانات:** مصادر مفتوحة مثل منصة Kaggle التي توفر قواعد بيانات (Datasets) للروابط الاحتيالية والحقيقية، إضافةً إلى مجموعات بيانات من مواقع اخرئ.
- **مكتبات إضافية:** Scikit-learn؛ للاستفادة من أدواتها في التقسيم (Train/Test Split) وتقييم النماذج (Metrics)، وغيرها من المهام المعاونة في التعلم الآلي.
- **واجهة التفاعل:** Flask؛ لإعداد صفحة التفاعل مع المستخدم وأدخال الرابط.

6.3 إجراءات التنفيذ

سير العمل التفصيلي للمشروع يمر عبر المراحل الموضحة في الصورة التالية:



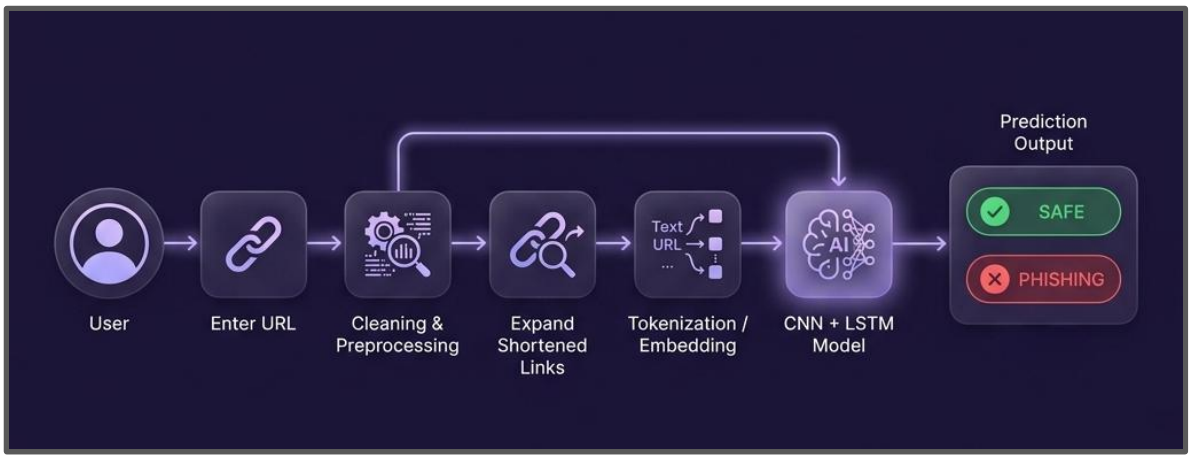
الشكل 2:3 المخطط العام للمشروع

1) جمع البيانات: تجميع مجموعة بيانات واسعة من الروابط المشروعة والمشبوهة من مصادر متعددة (مثل Kaggle وPhishTank ومنصات أخرى). يتضمن ذلك بيانات نصية عن الروابط (URL) ومعلومات مثل المحتوى المرتبط بها أو سمات مستخلصة مسبقًا.
2) المعالجة المسبقة للبيانات: تشمل تنظيف البيانات وإزالة التكرارات. تُركّز في هذه المرحلة على فك الروابط

لأن الروابط المختصرة قد تخفي الوجهة الحقيقية للروابط الخبيثة. كما يتم استخراج السمات الأساسية من الرابط ونصه (مثل الكلمات المفتاحية، والامتدادات، وطول الرابط، وغير ذلك) لتحويل البيانات إلى شكل مناسب للنموذج.

3) تصميم نموذج التعلم العميق: بناء نموذج شبكات عصبية هجين يستخدم الشبكة الالتفافية (CNN) لاستخراج السمات المكانية من تمثيل الرابط، والذاكرة طويلة المدى (LSTM) لمعالجة المعلومات التسلسلية إن وجدت (مثلاً عند التعامل مع سلسلة أحرف أو كلمات في الرابط). تسمح هذه البنية باستغلال نقاط القوة لكل تقنية: حيث تعالج CNN الأنماط الثابتة بينما تلتقط LSTM التسلسل الزمني في البيانات.
4) تدريب النموذج: تقسيم البيانات إلى مجموعات للتدريب والاختبار (على سبيل المثال 80% تدريب، 20% اختبار) باستخدام Scikit-learn. ثم يتم تدريب نموذج الـ CNN+LSTM على مجموعة البيانات التدريبية، مع ضبط المعاملات (Parameters) وتحسينها (مثل معدل التعلم وعدد الطبقات) لضمان أداء جيد. يستخدم التدريب خوارزمية (مثل Adam optimizer) وقياس الخسارة المناسب (Loss Function) لتصنيف الروابط إلى فئتين: حقيقية أو مخادعة.
5) تقييم النموذج: بعد التدريب، يُختبر النموذج على مجموعة الاختبار لتقييم دقته (Accuracy) ومعايير أخرى (مثل دقة التنبؤ Precision، والاستدعاء Recall، ومُعامل F1). يُسجّل أداء النموذج، للتأكد من فاعليته في كشف التصيد.
6) توثيق النتائج: في نهاية كل مرحلة يتم توثيق الخطوات والنتائج. يشمل ذلك إعداد تقارير حول أداء النموذج بعد كل تجربة، ومناقشة كيفية تحسين النتائج، بالإضافة إلى كتابة فصل الخاتمة (التقرير) الذي يلخص العمل ويعرض النتائج المستخلصة.

7.3 مخطط بنية النظام



الشكل 3:3 مخطط بنية النظام

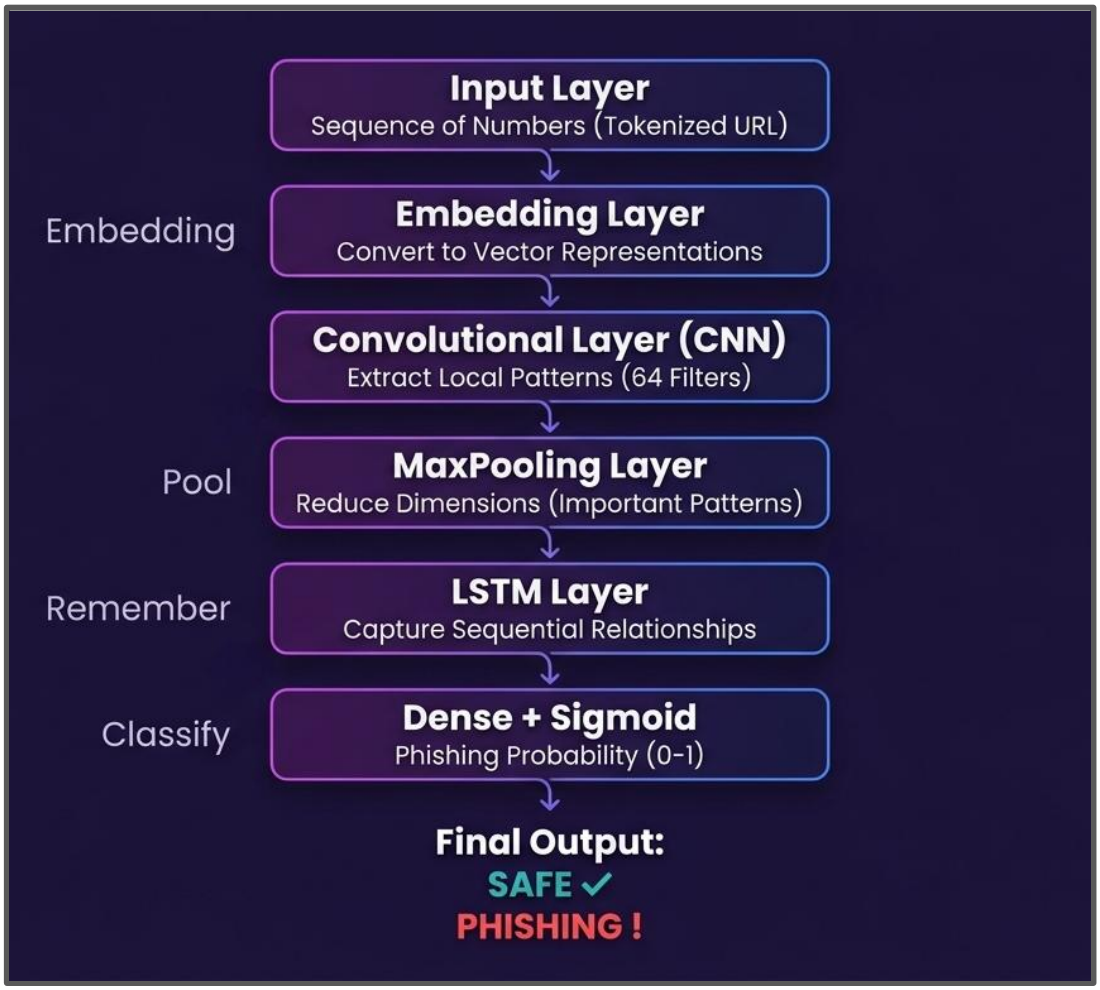
يتكون النظام بشكل عام من المكونات التالية: مدخلات من الروابط، وحدة المعالجة المسبقة، وحدة التصنيف (نموذج CNN+LSTM)، والمخرجات (تصنيف الروابط). التوضيح النصي أدناه يبيّن بنية النظام الرئيسية وطريقة تدفق البيانات داخله:

(إدخال) رابط URL ← [وحدة المعالجة المسبقة (Data Processing)] ← [نموذج التصنيف (CNN+LSTM)] ← (إخراج: رابط حقيقي أم مزيف)

تدفق البيانات (Data Flow): تبدأ العملية بجمع الروابط من مصادرها، ثم تقوم وحدة المعالجة المسبقة بتنظيف البيانات وتقصيرها وفك الروابط المختصرة، يلي ذلك استخراج الميزات (Features) مثل تمثيل الكلمات أو السمات العددية من كل رابط. بعد ذلك تُقسّم البيانات إلى مجموعات للتدريب والاختبار. البيانات المعالجة تخرج كناتج لهذه المرحلة وتدخل في النموذج.

بنية النموذج (Model Architecture): يتكون النموذج من طبقة تضمين (Embedding layer) تمهيدية إذا دعت الحاجة لتحويل النص إلى متجهات عددية، تليها عدة طبقات Convolutional لاستخلاص السمات المكانية من البيانات، ثم طبقة (أو أكثر) من LSTM لمعالجة التسلسلات المحتملة للسمات. بعد ذلك طبقة كثيفة (Dense) أخيرة مع دالة تفعيل Softmax أو Sigmoid لإخراج احتمال انتماء الرابط لكل فئة (حقيقي/مزيف). التمثيل التخطيطي المبسط للنموذج هو كالتالي:

طبقة Embedding ← طبقات CNN (Convolution + Pooling) ← طبقة LSTM ← طبقة إخراج (Softmax)



الشكل 4:3 مخطط بنية النموذج

سير العمل الكامل (Workflow): يمكن تلخيص خطوات النظام المتكاملة على النحو الآتي:

- جمع البيانات (روابط حقيقية ومشبوهة) من المصادر المحددة.
- المعالجة المسبقة (تنظيف البيانات، فك الروابط المختصرة، استخراج الميزات).
- تحضير البيانات (ترميز الميزات، تقسيم مجموعة التدريب/الاختبار).
- بناء وتصميم نموذج CNN+LSTM وفق البنية الموصوفة.
- تدريب النموذج على البيانات المعالجة.
- تقييم النموذج وتحسينه حسب النتائج.
- إخراج النتيجة النهائية (تصنيف الرابط) وتوثيق الأداء.

8.3 مراحل تدريب النموذج

قبل التدريب الفعلي، نقوم بتقسيم البيانات إلى مجموعات تدريب واختبار (مثل توزيع 80% تدريب و20% اختبار). يمكن أيضًا تخصيص جزء للتحقق (validation) لضبط المعلمات. أثناء التدريب، يستخدم النموذج خوارزمية التحسين Adam لتحديث الأوزان بشكل تدرجي. وبما أن المهمة تصنيف ثنائي، فإن دالة الخسارة المختارة هي Binary Cross-Entropy (ثنائية التقاطع)، التي تقيس الفرق بين التوقعات الحاصلة (احتمالات التصيد) والقيم الحقيقية (0 أو 1). من المعلمات التجريبية المهمة: حجم فضاء التضمين (Embedding Dimension)، عدد الفلاتر وأحجامها في طبقة الـ CNN، عدد وحدات LSTM وطبقات الـ Dense، ومعدل التعلم (Learning Rate)، وحجم الدفعات (Batch Size)، وعدد العصور (Epochs) اللازمة للتدريب. يتم ضبط هذه المعلمات بعناية لتعزيز دقة النموذج وتجنّب الإفراط في التعلّم (Overfitting)، مثلاً عبر إضافة طبقة Dropout إذا لزم الأمر.

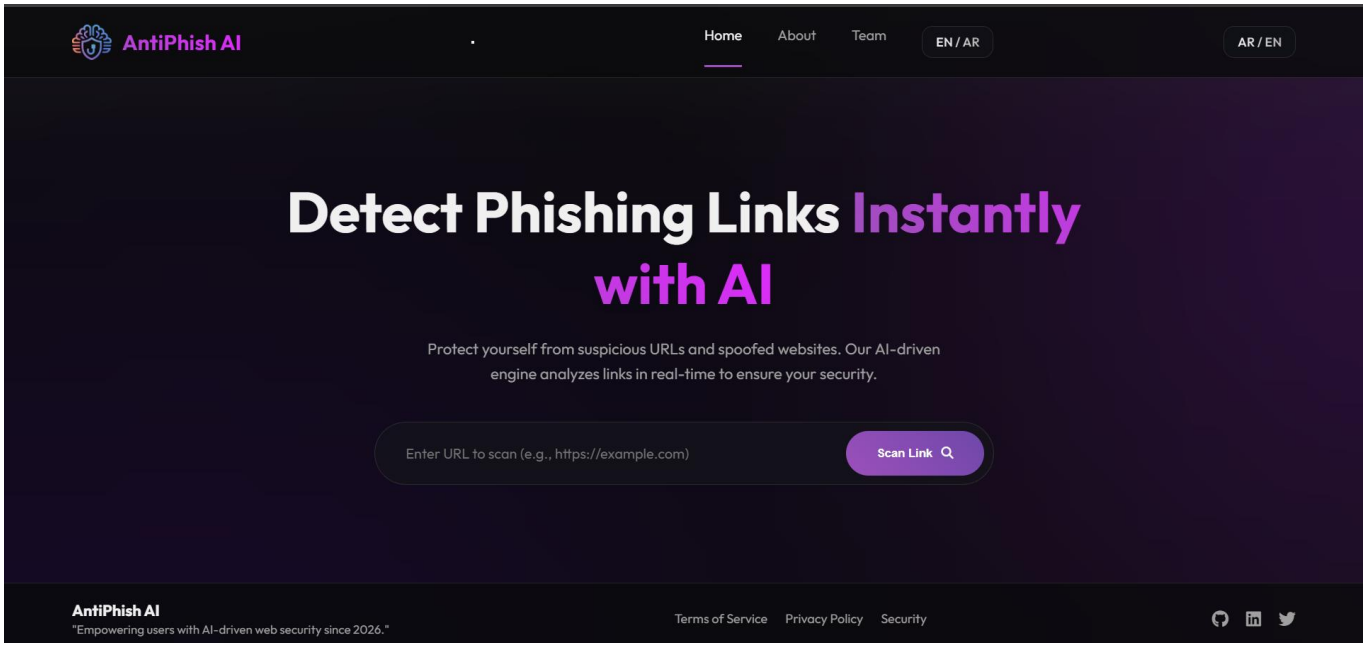
9.3 آلية التقييم

لقياس أداء النموذج، نعتمد مجموعة من المعايير الإحصائية المعروفة:

- الدقة (Accuracy): هي نسبة جميع التصنيفات الصحيحة من كل العينات.
- الدقة النوعية (Precision): تعبر عن نسبة التصنيفات الصحيحة للحالات الإيجابية مقارنة بكل الحالات التي صنّفها النموذج إيجابيًا.
- نسبة الاستدعاء (Recall): تمثل نسبة الكشف عن الحالات الإيجابية الصحيحة (أي من بين كل الروابط الاحتيالية الحقيقية).
- مؤشر F1: هو المتوسط الهندسي (الهاشمي) بين الدقة النوعية والاستدعاء، ويعطي مقياسًا واحدًا يعادل أدائهما معًا.

- مساحة تحت منحنى ROC (AUC): تقيس قدرة النموذج على التمييز بين الفئتين عبر كافة العتبات الممكنة. قيمة AUC الأقرب إلى 1 تعني نموذجًا متميزًا في الفصل، بينما 0.5 تدل على أداء عشوائي.
- باستخدام هذه المقاييس، يمكن مقارنة النموذج المُدرّب مع نماذج أخرى أو تقييمه عبر مجموعات اختبار مختلفة لضمان موثوقية النتائج.

10.3 الواجهة المقترحة



الشكل 5:3 الواجهة المقترحة

11.3 ملخص الفصل

في هذا الفصل تم شرح المنهجية الكاملة لتنفيذ مشروع الكشف عن التصيد الاحتيالي عبر الروابط باستخدام نموذج هجيني (CNN+LSTM). اشتمل الشرح على اعتماد منهجية Agile في إدارة المشروع، والأدوات البرمجية المستخدمة (مثل Python و pytorch)، بالإضافة إلى مصادر بيانات الروابط (Hugging Face وKaggle وغيرهما). تم تفصيل خطوات معالجة البيانات الأولية بما فيها التنظيف وتوسيع الروابط المختصرة وتحويل النص إلى متسلسلات رقمية. كما تم تقديم وصف تفصيلي لبنية النموذج مقترنةً برسم تخطيطي توضيحي، وذكر عمليات التدريب والخوارزمية المعتمدة ودالة الخسارة. أخيرًا، استعرضنا معايير تقييم الأداء المستخدمة لقياس فاعلية النموذج. يعكس هذا الفصل الإطار العملي والنظري المتكامل للعملية البحثية للمشروع، مع توفير الأساس اللازم للانتقال إلى فصل النتائج التطبيقية وتحليل أداء النموذج.

الفصل الرابع

التجارب العملية ونتائج التدريب

1.4 مقدمة الفصل

يتناول هذا الفصل الجانب التطبيقي من المشروع، حيث يتم عرض مراحل تنفيذ نموذج كشف التصيّد الاحتيالي عبر الروابط، بدءًا من تجهيز البيانات ومعالجتها، مرورًا ببناء النموذج وتدريبه، وصولًا إلى تقييم أدائه باستخدام مقاييس معيارية. ويهدف هذا الفصل إلى تحليل مدى كفاءة النموذج المقترح في التمييز بين الروابط الشرعية وروابط التصيّد، مع تقديم مناقشة علمية للنتائج المستخلصة.

يعتمد النموذج على بنية هجينة تجمع بين الشبكات الالتفافية (CNN) والشبكات العصبية المتكررة (LSTM)، حيث تم استخدام تمثيل على مستوى الأحرف (Character-Level Representation) لمعالجة الروابط النصية.

2.4 إعدادات التجربة وبيئة العمل

تم تنفيذ كافة التجارب البرمجية باستخدام لغة Python (الإصدار 3.13)، وبالاعتماد على إطار العمل PyTorch المتخصص في بناء الشبكات العصبية. تم استخدام بيئة التطوير Visual Studio Code عبر ملحق Notebook Jupyter. ونظراً لعدم توفر وحدة معالجة رسومية (GPU)، تم إجراء عمليات التدريب والتقييم على وحدة المعالجة المركزية (CPU)، مع تحسين أداء الذاكرة باستخدام تقنية المعالجة بالدفعات (Batches).

1.2.4 وصف مجموعة البيانات (Dataset)

اعتمد المشروع على دمج مجموعتي بيانات لضمان تنوع الأنماط وزيادة قدرة النموذج على التعميم:

1. **المجموعة الأولى**: مستخرجة من منصة (Hugging Face) تحت اسم

   .ealvaradob/phishing-dataset

2. **المجموعة الثانية**: مستخرجة من موقع (Mendeley Data).

يوضح الجدول التالي توزيع الروابط في مجموعة البيانات النهائية بعد عملية الدمج:

جدول 1:4 توزيع عينات مجموعة البيانات

الفئة	العدد (Samples)	النسبة المئوية
روابط تصيد (Phishing)	474,553	~38%
روابط آمنة (Legitimate)	772,708	~62%
المجموع الكلي	**1,247,261**	**%100**

تم تقسيم البيانات بشكل عشوائي بنسبة 80% للتدريب (Training) و20% للاختبار (Testing) لضمان تقييم النموذج على بيانات لم يسبق له رؤيتها.

2.2.4 معالجة عدم توازن البيانات

بما أن البيانات تميل لصالح الروابط الآمنة (62%) مقارنة بروابط التصيد (38%)، تم استخدام أسلوب التعلم الحساس للتكلفة (Cost-Sensitive Learning). تم تطبيق ذلك عبر دالة الخسارة (BCEWithLogitsLoss) باستخدام معامل الترجيح (pos_weight) لتعويض النقص في عينات التصيد.

يعتمد هذا الأسلوب على إعطاء وزن أكبر للفئة الأقل تمثيلًا في البيانات (روابط التصيّد)، بحيث يتم معاقبة النموذج بشكل أكبر عند الخطأ في تصنيف هذه الفئة. وقد تم حساب قيمة معامل الترجيح بناءً على نسبة عدد العينات بين الفئتين كما يلي:

$$pos\_weight = \frac{Nnegative}{Npositive} \qquad (1)$$

حيث يمثل (*Nnegative*) عدد العينات الآمنة، و(*Npositive*) عدد عينات التصيد. يضمن هذا الإجراء أن يعطي النموذج أهمية أكبر للأخطاء المتعلقة بالفئة الأقل (التصيد)، مما يحسن من قيمة الاستدعاء (Recall).

3.2.4 إعداد البيانات والمعالجة المسبقة

تم بناء قاموس (Vocabulary) يحتوي على كافة الأحرف والأرقام والرموز الخاصة المتكررة في الروابط. تم تخصيص رقم فريد (Unique ID) لكل حرف، مع إضافة رمزين خاصين لضمان كفاءة المعالجة:

- رمز الحشو (<PAD>): يستخدم لتوحيد أطوال الروابط بجعلها جميعاً بنفس الطول.
- رمز المجهول (<UNK>): يستخدم لتمثيل أي أحرف نادرة أو غريبة تظهر في الروابط ولا توجد في القاموس الأساسي.

تم تحديد الطول الأقصى للروابط بناءً على النسبة المئوية 95%، حيث بلغ **108** حرفًا، وذلك لتجنب تأثير القيم الشاذة. بعد ذلك، تم تحويل جميع الروابط إلى تسلسلات رقمية ثابتة الطول.

كما تم تقسيم البيانات إلى مجموعة تدريب واختبار باستخدام أسلوب التقسيم الطبقي (Stratified Split) لضمان الحفاظ على توزيع الفئات.

4.2.4 بنية النموذج

1. طبقة التضمين (Embedding)

تحول هذه الطبقة كل حرف إلى متجه عددي بحجم 128، مما يسمح للنموذج بفهم العلاقات بين الأحرف.

2. طبقات الالتفاف (CNN)

تم استخدام طبقتين التفافيتين لاستخراج الأنماط المحلية من الروابط، مع طبقات تجميع (MaxPooling) لتقليل الأبعاد.

3. طبقة (LSTM) ثنائية الاتجاه

تم استخدام (LSTM) ثنائي الاتجاه بحجم مخفي 64 لالتقاط العلاقات التسلسلية بين الأحرف من كلا الاتجاهين.

4. طبقات التصنيف

تم استخدام طبقات (Fully Connected) مع دالة تنشيط (ReLU) و(Dropout) لتقليل الإفراط في التعلم، وتنتهي بطبقة إخراج واحدة تمثل درجة احتمال التصيد.

والجدول التالي يوضح كل طبقة على حده ومعاملاتها:

جدول 2:4 طبقات النموذج المصممة

الطبقة	النوع	المعاملات
1	Embedding	vocab_size=261, embedding_dim=128, padding_idx=0
2	Conv1D	in_channels=128, out_channels=64, kernel=3, padding=1
3	MaxPool1D	kernel=2
4	Conv1D	in_channels=64, out_channels=128, kernel=3, padding=1
5	MaxPool1D	kernel=2
6	Bidirectional LSTM	input_size=128, hidden_size=64, num_layers=2, dropout=0.3
7	Fully Connected	128 → 64
8	Dropout	0.3
9	Fully Connected	64 → 1
10	Sigmoid	-

5.2.4 معلمات التدريب

لضمان استقرار النموذج ووصوله إلى حالة التقارب (Convergence) بأفضل صورة ممكنة، تم تحديد معلمات التدريب (Hyperparameters) بعناية بناءً على التجارب الأولية. يوضح الجدول التالي تفاصيل هذه المعلمات:

جدول 3:4 معلمات تدريب نموذج CNN-BiLSTM

المعامل (Parameter)	القيمة (Value)	الملاحظات
عدد العصور (Epochs)	5	تم اختيار عدد قليل لتجنب الإفراط في التعلم.
حجم الدفعة (Batch Size)	256	موازنة بين استهلاك الذاكرة واستقرار التدرج.
دالة الخسارة (Loss Function)	BCEWithLogitsLoss	دالة معيارية للتصنيف الثنائي مع استقرار عددي.
المُحسّن (Optimizer)	Adam	معدل تعلم ابتدائي قدره 0.001.
جدولة معدل التعلم	ReduceLROnPlateau	خفض معدل التعلم عند ثبات الخسارة.

3.4 بيئة تطوير النظام

تم تنفيذ هذا المشروع باستخدام مجموعة من الأدوات والتقنيات الحديثة التي تدعم تطوير نماذج الذكاء الاصطناعي وتطبيقها عمليا في مجال الأمن السيبراني. حيث تم الاعتماد على لغة البرمجة Python نظرا لمرونتها وسهولة استخدامها، بالإضافة إلى توفر عدد كبير من المكتبات المتخصصة في مجالات التعلم الآلي والتعلم العميق.

وقد تم استخدام مكتبات مثل TensorFlow وKeras لبناء النموذج المقترح وتدريبه، حيث توفر هذه المكتبات بيئة قوية لتصميم الشبكات العصبية ومعالجة البيانات بكفاءة عالية.

كما تم الاستفادة من مكتبات إضافية مثل NumPy وPandas في معالجة البيانات وتنظيمها، مما ساعد في تسهيل عمليات التحليل والتحويل قبل إدخالها إلى النموذج.

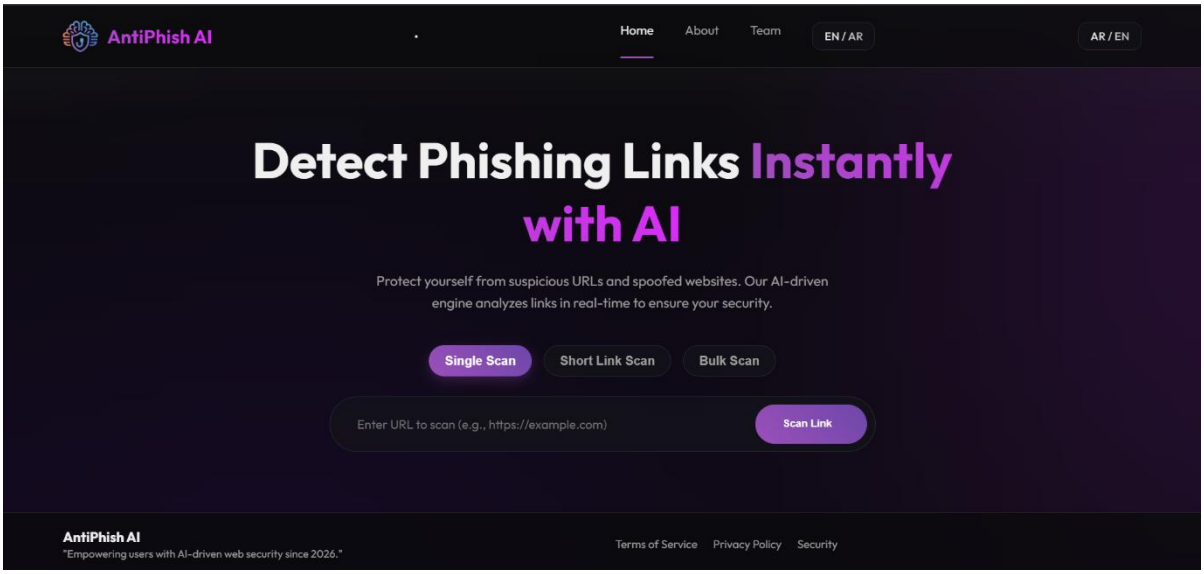
أما على مستوى واجهة المستخدم، فقد تم استخدام إطار العمل Flask لتطوير تطبيق ويب بسيط يتيح للمستخدم إدخال الرابط المراد فحصه والحصول على نتيجة فورية.

وقد ساهم هذا الإطار في ربط النموذج بالواجهة بشكل عملي، مما يعكس إمكانية تطبيق النظام في بيئة واقعية.

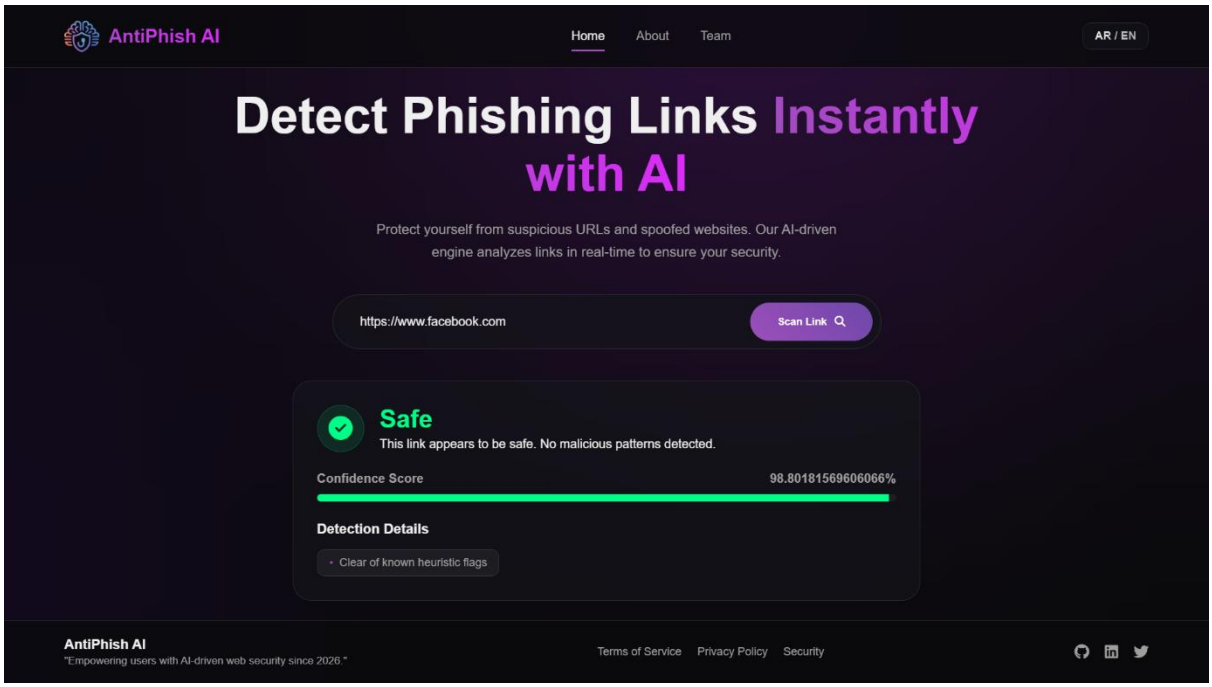
تم تنفيذ التجارب على بيئة حاسوبية مناسبة تضمن أداء جيدا أثناء عملية التدريب، حيث تم ضبط إعدادات النظام بما يتوافق مع متطلبات النموذج من حيث سرعة المعالجة واستهلاك الموارد. كما تم مراعاة تنظيم بيئة العمل وتكامل الأدوات المستخدمة لضمان استقرار النظام وسهولة تشغيله.

بشكل عام، ساهمت هذه البيئة المتكاملة في دعم جميع مراحل تطوير النظام، بدءا من معالجة البيانات، مرورا ببناء النموذج وتدريبه، وانتهاء بتطبيقه واختباره ضمن واجهة استخدام عملية.

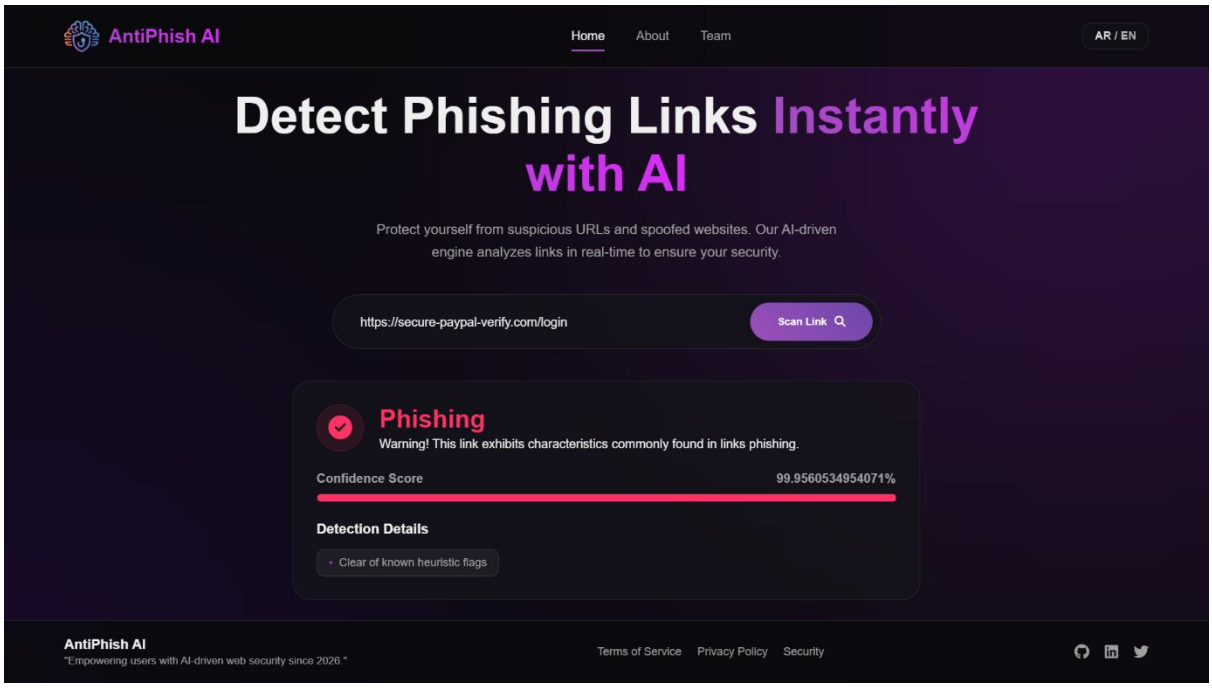
4.4 الواجهة الرسومية للنظام



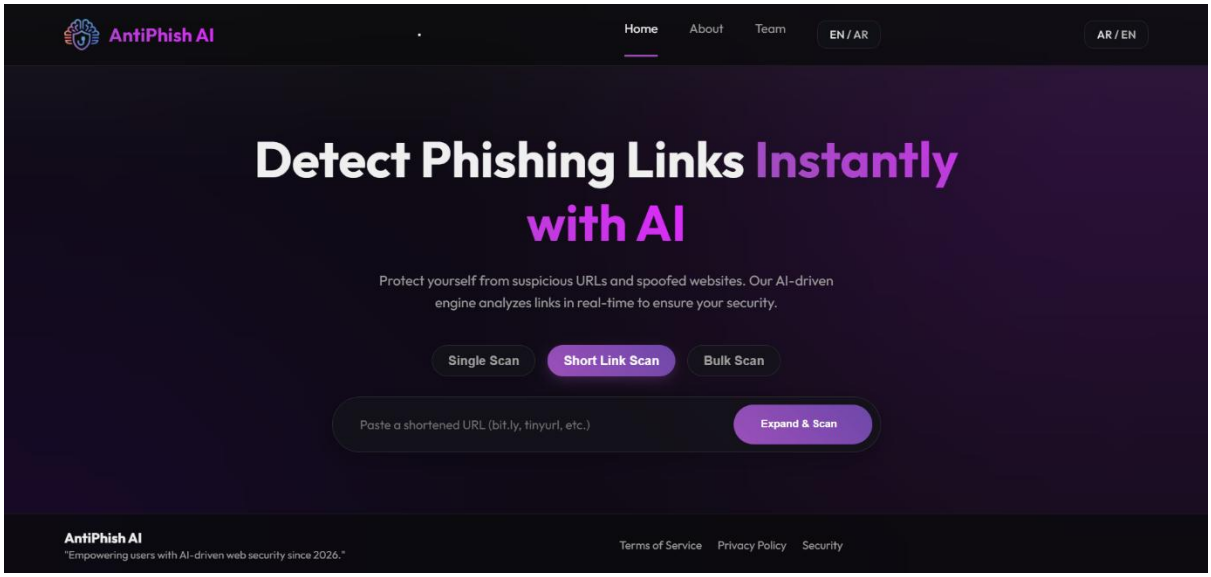
الشكل 1:4 الواجهة الرئيسية للنظام



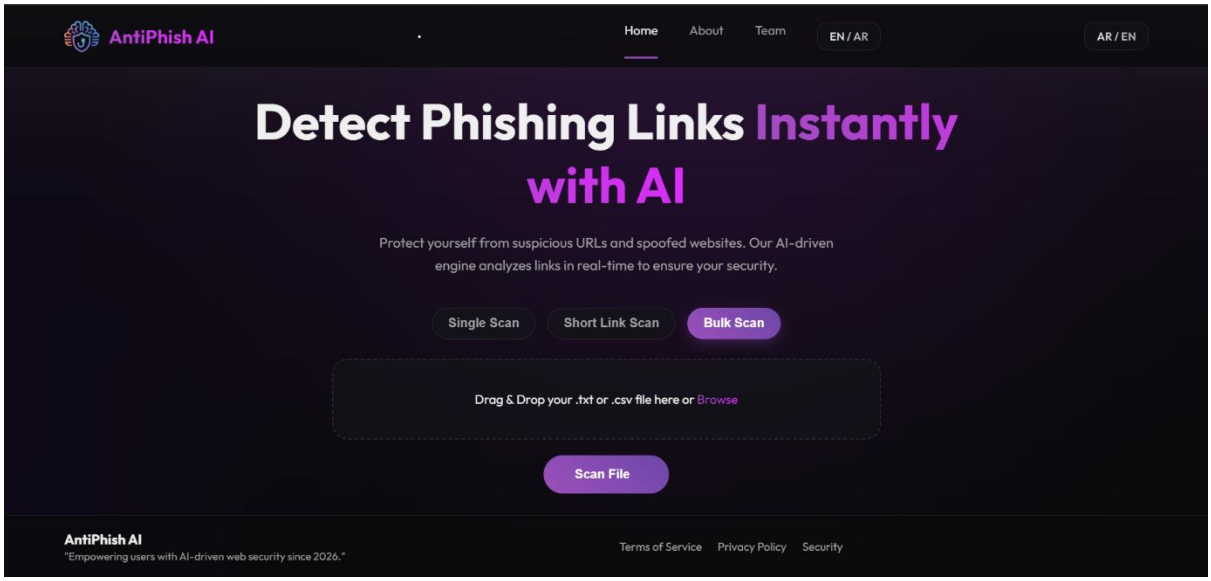
الشكل 2:4 واجهة عرض نتيجة فحص الرابط (آمن)



الشكل 3:4 واجهة عرض نتيجة فحص الرابط (غير آمن)



الشكل 4:4 فحص الروابط المختصرة



الشكل 5:4 فحص ملف كامل يحتوي على الروابط

5.4 خلاصة الفصل

استعرض هذا الفصل الجانب التطبيقي للمشروع، بدءا من معالجة البيانات التي تضم أكثر من مليون رابط، مرورا ببناء وتدريب النموذج المقترح (CNN-BiLSTM) ، وصولا إلى تقييم أدائه باستخدام مقاييس معيارية.

وقد أظهرت النتائج أن النموذج حقق دقة بلغت 98.46% واستدعاء (Recall) بنسبة 97.44%، مما يعكس كفاءته العالية في التمييز بين الروابط الآمنة وروابط التصيد الاحتيالي.

ورغم هذه النتائج المتميزة، لا تزال هناك إمكانية لإجراء تحسينات مستقبلية، مثل تطوير خصائص إضافية للنموذج، بما يسهم في تقليل الأخطاء وتعزيز قدرته على التعامل مع الحالات المعقدة.

الفصل الخامس

النتائج والتحليل

1.5 مقدمة الفصل

يُعد هذا الفصل خاتمةً تحليلية للمشروع، حيث تُناقش فيه النتائج التي تم التوصل إليها بعد بناء نموذج كشف التصيّد الاحتيالي عبر الروابط، وتقييم أدائه على مجموعة الاختبار وعلى روابط جديدة غير مرئية أثناء التدريب. كما يتضمن هذا الفصل تفسيرًا لنتائج النموذج، وتحليلًا للأخطاء، والإجابة عن أسئلة البحث، ثم استخلاص أهم الاستنتاجات، وأخيرًا طرح مجموعة من التوصيات والأعمال المستقبلية التي يمكن أن تُسهم في تطوير النموذج وتحسين فعاليته.

2.5 مناقشة النتائج

1.2.5 أداء النموذج المقترح

تشير النتائج إلى تحقيق توازن ملحوظ بين جميع مقاييس التقييم، حيث سجل النموذج قيمًا مرتفعة ومتقاربة في كل من الدقة (Accuracy) والدقة الإيجابية (Precision) والاستدعاء(Recall) ، وهو ما يعكس استقرار النموذج وقدرته على التمييز الفعّال بين الروابط الآمنة وروابط التصيّد.

كما أن ارتفاع قيمة الاستدعاء (Recall) يُعد مؤشرًا مهمًا على نجاح النموذج في اكتشاف معظم روابط التصيّد، وهو عامل حاسم في التطبيقات الأمنية.

أظهر نموذج **CNN-BiLSTM** المقترح أداءً مرتفعًا جدًا في مهمة تصنيف الروابط إلى شرعية وروابط تصيّد، حيث حقق **دقة كلية بلغت 98.46%**على مجموعة الاختبار، مع **Precision = %98.49** و**Recall = %97.44** و**F1-Score = %97.97**. وتُعد هذه النتائج مؤشرًا قويًا على فاعلية النموذج في التمييز بين الروابط الخبيثة والآمنة، كما تؤكد أن الدمج بين الشبكات الالتفافية (CNN) والشبكات العصبية المتكررة ثنائية الاتجاه (BiLSTM) كان اختيارًا ناجحًا لهذه المهمة.

ويُعزى هذا الأداء المرتفع إلى أن **التمثيل على مستوى الأحرف (Character-Level Representation)** يسمح للنموذج بالتعامل مع الروابط بصفتها سلاسل نصية ذات بنية خاصة، دون الاعتماد على الكلمات فقط. وهذا الأمر بالغ الأهمية في مجال كشف التصيّد، لأن الروابط غالبًا ما تحتوي على رموز خاصة، ونطاقات فرعية، وتسلسلات غير مألوفة، لا يمكن تمثيلها بكفاءة باستخدام طرق التمثيل التقليدية على مستوى الكلمات.

كما أسهمت طبقات **CNN** في التقاط الأنماط المحلية داخل الرابط، مثل تكرار بعض المقاطع أو ظهور رموز مشبوهة، في حين ساعدت طبقات **BiLSTM** على فهم السياق التسلسلي للعناصر داخل الرابط، وهو ما عزز قدرة النموذج على اتخاذ قرار تصنيفي أكثر دقة.

2.2.5 تحليل الأخطاء

أظهرت مصفوفة الالتباس وجود نوعين من الأخطاء، وهما:

أولًا: الإنذارات الكاذبة (False Positives)
بلغ عدد الروابط الشرعية التي صُنفت خطأً على أنها تصيّد **1415 رابطًا** .وتُعزى هذه الأخطاء إلى تشابه بعض الروابط الشرعية من الناحية الشكلية مع الروابط التصيدية، خاصة عندما تحتوي على كلمات مثل login أو secure أو عندما تكون طويلة نسبيًا أو تضم نطاقات فرعية متعددة. ورغم أن هذا النوع من الأخطاء قد يسبب بعض الإزعاج للمستخدم، إلا أنه أقل خطورة من الناحية الأمنية مقارنة بتمرير رابط تصيّد حقيقي دون كشفه.

ثانيًا: الروابط التصيدية الفائتة (False Negatives)
بلغ عدد الروابط التصيدية التي لم يكتشفها النموذج **2425 رابطًا** .ويعد هذا النوع من الأخطاء أكثر خطورة، لأنه يعني أن بعض الروابط الخبيثة استطاعت تجاوز آلية الكشف. وقد يعود ذلك إلى أن بعض الروابط التصيدية الحديثة تستعمل بنيات نصية غير تقليدية أو تتخفى داخل روابط قصيرة أو نطاقات تبدو طبيعية نسبيًا. كما أن اعتماد النموذج على التحليل الحرفي فقط دون الاستفادة من ميزات إضافية مثل عمر النطاق أو شهادة SSL أو السمعة الخارجية قد يحد من قدرته على اكتشاف بعض الحالات المعقدة.

ومع ذلك، فإن نسبة الخطأ بقيت منخفضة نسبيًا مقارنة بحجم مجموعة الاختبار، وهو ما يعكس جودة النموذج وفاعليته العملية.

3.2.5 تحليل أداء النموذج على روابط جديدة

عند اختبار النموذج على روابط جديدة غير موجودة ضمن بيانات التدريب، أظهر أداءً جيدًا في معظم الحالات. فقد صنّف الروابط التصيدية الواضحة مثل secure-paypal-verify.com وapple-id-verify.xyz على أنها تصيد بثقة عالية جدًا، كما صنّف الروابط الشرعية المعروفة مثل google.com وfacebook.com بشكل صحيح على أنها آمنة.

وقد أظهرت نتائج الاختبار على بيانات جديدة أن النموذج تمكن من اكتشاف 52,110 رابط تصيّد من أصل 54,807 رابط، مع وجود عدد محدود من الحالات التي لم يتم التعرف عليها. ويعكس هذا الأداء قدرة النموذج على التعميم والعمل بكفاءة خارج بيانات التدريب.

ويظهر هذا السلوك أن النموذج قوي في اكتشاف الروابط التصيدية ذات البنية الواضحة، لكنه قد يحتاج إلى تحسينات إضافية لزيادة دقته في التعامل مع المواقع الشرعية الشهيرة ذات البنية النصية البسيطة.

4.2.5 تأثير عدد دورات التدريب (Epochs)

أظهرت نتائج التدريب انخفاضًا تدريجيًا في قيمة الخسارة، حيث انخفضت خسارة التدريب (Training Loss) من 0.0925 إلى 0.0369، بينما انخفضت خسارة التحقق (Validation Loss) من 0.0592 إلى 0.0445. ويشير هذا الانخفاض المتوازن إلى أن النموذج تعلم بشكل تدريجي ومستقر دون ظهور مؤشرات واضحة على فرط التعلّم (Overfitting)، مما يدل على ملاءمة عدد دورات التدريب المستخدمة.

كما أن الفارق المحدود بين خسارة التدريب وخسارة التحقق يدل على أن النموذج لم يحفظ بيانات التدريب فقط، بل اكتسب قدرة معقولة على التنبؤ على بيانات لم يرها من قبل. ويؤكد ذلك أن عدد العصور المستخدم كان كافيًا للوصول إلى أداء مرتفع، وأن الزيادة العشوائية في عدد العصور لا تؤدي بالضرورة إلى تحسن جوهري في الأداء ما لم تُدعَم بتحسينات في البيانات أو البنية النموذجية.

5.2.5 تأثير معالجة عدم توازن البيانات

تم التعامل مع مشكلة عدم توازن البيانات باستخدام أسلوب **التعلم الحساس للتكلفة (Cost-Sensitive Learning)** عبر إدخال **pos_weight** داخل دالة الخسارة **BCEWithLogitsLoss**. وقد ساعد هذا الأسلوب على إعطاء

وزن أكبر للفئة الأقل تمثيلًا، وهي فئة روابط التصيّد، مما حدّ من انحياز النموذج نحو الفئة الأكبر، وأسهم في تحسين قدرته على اكتشاف الحالات الخبيثة.

وقد انعكس هذا الإجراء إيجابيًا على قيمة **Recall**، التي بلغت نسبة مرتفعة، وهو ما يؤكد أن النموذج أصبح أكثر حساسية تجاه الروابط التصيدية، مع المحافظة في الوقت نفسه على مستوى جيد من الدقة العامة.

بشكل عام، تؤكد النتائج أن النموذج المقترح يتمتع بقدرة عالية على استخراج الأنماط النصية المعقدة من الروابط، بفضل الدمج بين طبقات (CNN) و .(BiLSTM)ومع ذلك، لا تزال هناك بعض التحديات المرتبطة بالروابط القصيرة أو التي تحاكي نطاقات موثوقة، مما يستدعي تطوير النموذج مستقبلاً من خلال دمج ميزات إضافية أو تحسين جودة البيانات.

3.5 الإجابة على أسئلة البحث

استنادًا إلى النتائج التي تم التوصل إليها، يمكن الإجابة عن أسئلة البحث الرئيسة كما يلي:

هل يمكن لنموذج CNN-LSTM تحقيق دقة عالية في كشف روابط التصيّد؟
نعم، فقد حقق النموذج دقة بلغت **98.46%**، وهو أداء مرتفع جدًا ويُظهر فعاليته في التمييز بين الروابط الشرعية وروابط التصيّد.

هل يُعد التمثيل على مستوى الأحرف مناسبًا لهذا النوع من البيانات؟
نعم، لأن الروابط تحتوي على رموز خاصة وبنية نصية مميزة لا تتناسب مع التمثيل القائم على الكلمات فقط، في حين يتيح التمثيل على مستوى الأحرف التقاط الأنماط الدقيقة داخل الرابط.

ما أثر عدم توازن البيانات على الأداء؟
تمت معالجة هذا الأثر عبر أسلوب الترجيح داخل دالة الخسارة، مما ساعد النموذج على تحقيق توازن أفضل بين الفئتين، وخاصة في رفع قدرته على اكتشاف التصيّد.

هل أدت زيادة عدد العصور إلى تحسن كبير في الأداء؟
أظهرت النتائج أن النموذج وصل إلى مستوى جيد من الأداء خلال عدد محدود من العصور، وأن التحسن بعد ذلك كان تدريجيًا ومحدودًا، مما يشير إلى أن جودة البيانات وبنية النموذج كانتا أكثر تأثيرًا من مجرد زيادة العصور.

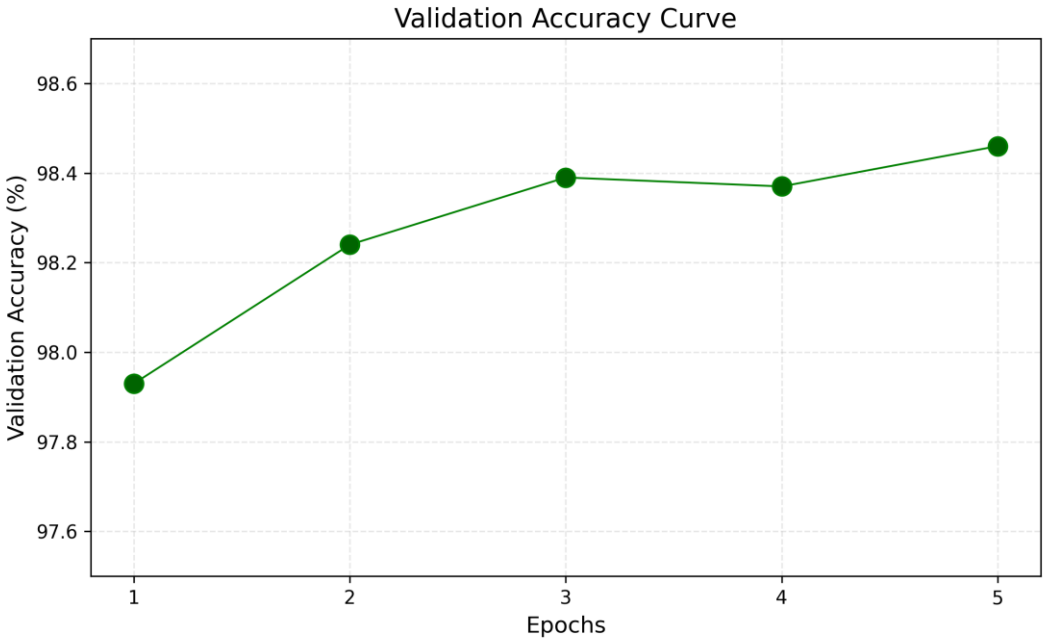
4.5 الاستنتاجات

من خلال هذا المشروع، تم التوصل إلى عدد من الاستنتاجات المهمة، أبرزها:

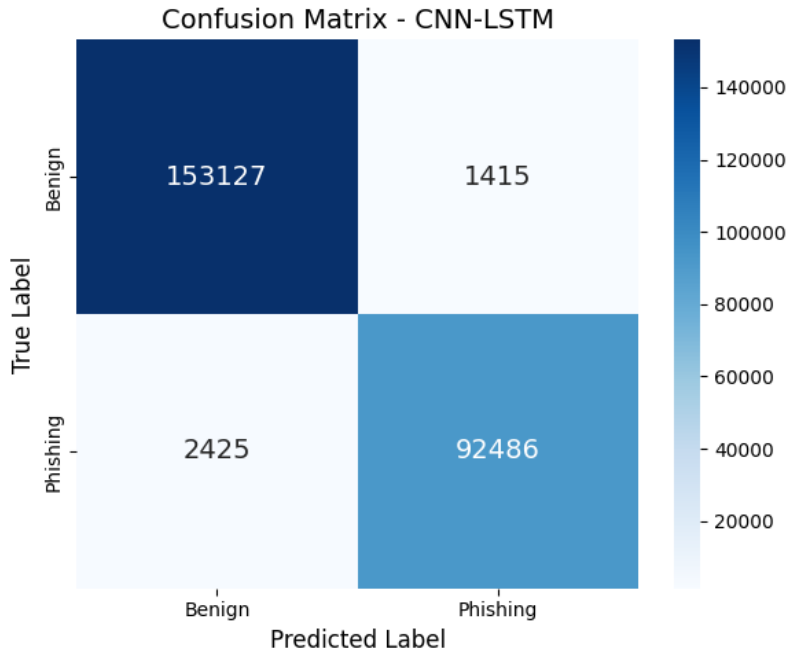
1. نجاح تصميم وتنفيذ نموذج **CNN-LSTM** لتصنيف روابط الويب اعتمادًا على التمثيل الحرفي .
2. تحقيق أداء مرتفع في التصنيف، حيث بلغت الدقة الكلية **98.46%**، وبلغت قيمة **F1-Score 97.97%.**
3. أثبتت المعالجة الحرفية فعاليتها في تمثيل الروابط واستخراج الأنماط المفيدة منها .
4. ساهمت آلية **pos_weight** في تحسين التعامل مع عدم توازن البيانات .

5. أظهر النموذج قدرة قوية على اكتشاف الروابط التصيدية الواضحة، مع بعض التحديات في الروابط الشرعية القصيرة أو الشهيرة.
6. إن دمج النموذج مع ميزات إضافية قد يرفع من دقته ويقلل من الأخطاء في الحالات الحدّية .

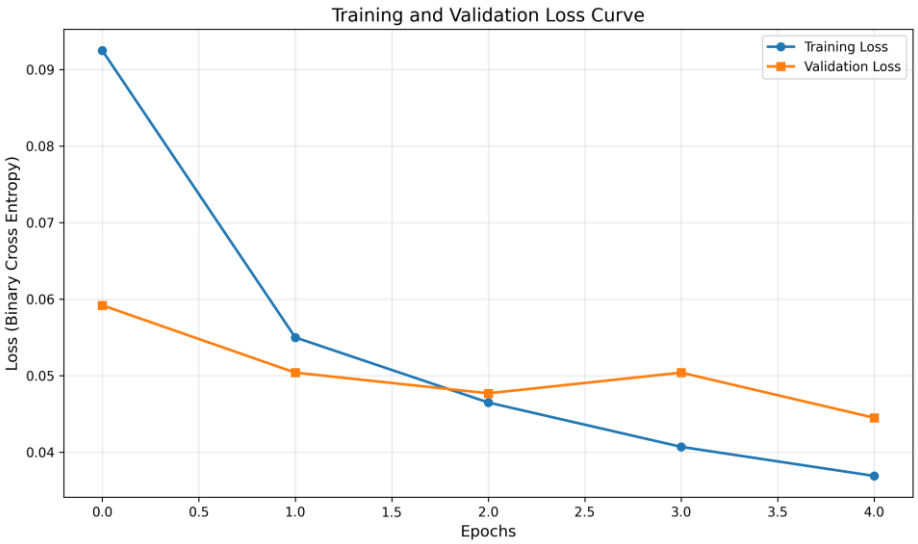
وبناءً على ذلك، يمكن القول إن النموذج المقترح يمثل حلاً فعّالًا وواعدًا في مجال كشف التصيّد الاحتيالي عبر الروابط، وقابلًا للتطوير نحو أنظمة أكثر شمولًا وواقعية.



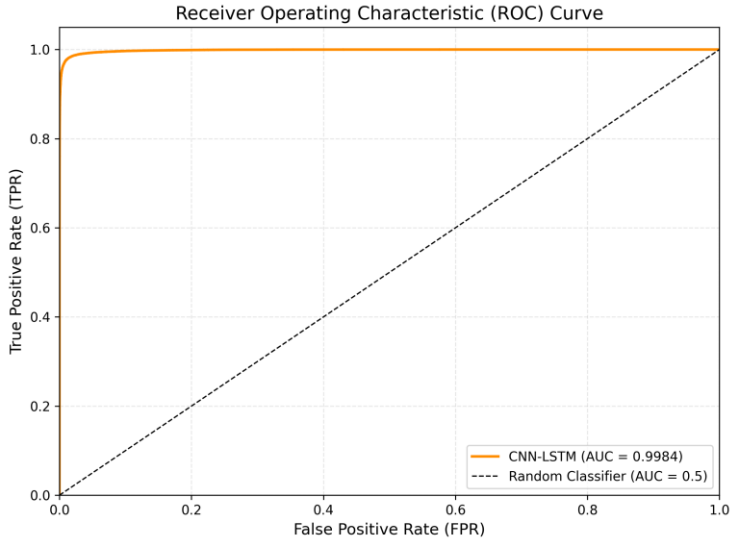
الشكل 1:5 يوضح منحنى دقة التحقق تحسن أداء النموذج واستقراره عبر مراحل التدريب.



الشكل 2:5 يوضح مصفوفة الارتباك كفاءة النموذج في تصنيف الروابط الآمنة والاحتيالية بدقة عالية.



الشكل 3:5 يوضح منحنى خسارة التدريب والتحقق استقرار عملية التعلم وتحسن أداء النموذج.



الشكل 4:5 يوضح منحنى ROC القدرة العالية للنموذج على التمييز بين الروابط الآمنة وروابط التصيد.

5.5 خلاصة الفصل

في نهاية هذا الفصل، يمكن التأكيد على أن المشروع نجح في تحقيق هدفه الأساسي المتمثل في بناء نموذج فعال لكشف التصيّد الاحتيالي عبر الروابط باستخدام تقنيات التعلم العميق. وقد أظهرت النتائج أن نموذج **CNN-BiLSTM** حقق أداءً قويًا وموثوقًا، مع قدرة جيدة على التمييز بين الروابط الآمنة والضارة، مما يجعله أساسًا مناسبًا للتطوير في تطبيقات أمنية عملية مستقبلًا.

كما أن هذا العمل يبرز أهمية توظيف تقنيات الذكاء الاصطناعي في حماية المستخدمين من التهديدات الإلكترونية المتزايدة، ويؤكد أن الجمع بين المعالجة الحرفية والنماذج الهجينة يمكن أن يحقق نتائج واعدة في هذا المجال.

الفصل السادس

النتائج والمناقشة

1.6 مقدمة

في هذا الفصل يتم عرض النتائج التي تم الحصول عليها من خلال تطبيق النموذج المقترح لتحليل وتصنيف الروابط الإلكترونية، وذلك بهدف الكشف عن هجمات التصيد الاحتيالي. كما يتناول هذا الفصل تحليل هذه النتائج ومناقشتها، ومدى توافقها مع أهداف المشروع، بالإضافة إلى تقييم أداء النموذج المستخدم.

2.6 عرض النتائج

بعد الانتهاء من تدريب النموذج باستخدام مجموعة البيانات(Dataset) ، تم اختبار النموذج على مجموعة من الروابط غير المستخدمة في التدريب، وذلك للتحقق من كفاءة النموذج في التنبؤ.

وقد أظهرت النتائج أن النموذج قادر على التمييز بين الروابط الآمنة والروابط الخبيثة بنسبة دقة مرتفعة، حيث تمكن من تصنيف معظم الروابط بشكل صحيح.

كما تم ربط النموذج بواجهة ويب باستخدام إطار العملFlask ، مما أتاح للمستخدم إدخال أي رابط إلكتروني والحصول على نتيجة فورية توضح ما إذا كان الرابط آمن أو يمثل تهديدًا.

3.6 تحليل النتائج

من خلال النتائج التي تم الحصول عليها، يمكن ملاحظة أن النموذج يتمتع بقدرة جيدة على اكتشاف الروابط الاحتيالية، ويعود ذلك إلى عدة عوامل، من أبرزها:

- استخدام مجموعة بيانات مناسبة ومتنوعة
- اختيار خوارزمية تعلم آلي فعالة
- تدريب النموذج بشكل جيد

ومع ذلك، لوحظ وجود نسبة بسيطة من الأخطاء في التصنيف، حيث قام النموذج أحيانًا بتصنيف بعض الروابط الآمنة على أنها خبيثة، أو العكس. وهذا أمر متوقع في مثل هذه الأنظمة.

4.6 تقييم أداء النموذج

تم تقييم أداء النموذج باستخدام مجموعة من المقاييس، من أهمها:

- الدقة (Accuracy)
- الاسترجاع (Recall)
- الدقة الإيجابية (Precision)

وقد أظهرت النتائج أن النموذج يحقق أداءً جيدًا في جميع هذه المقاييس، مما يدل على كفاءته في التعامل مع مشكلة كشف التصيد الاحتيالي.

كما أن زمن الاستجابة كان سريعًا عند اختبار الروابط عبر واجهة الموقع، وهو ما يعزز من إمكانية استخدام النظام بشكل عملي.

5.6 مناقشة النتائج

تُظهر النتائج أن استخدام تقنيات الذكاء الاصطناعي في كشف هجمات التصيد الاحتيالي يعتبر حلًا فعالًا وعمليًا، خاصة عند دمجه مع واجهة استخدام سهلة مثل موقع الويب.

ومن خلال التجربة، تبين أن النظام يمكن أن يساعد المستخدمين في اتخاذ قرارات أكثر أمانًا عند التعامل مع الروابط، مما يقلل من احتمالية التعرض للهجمات الإلكترونية.

ومع ذلك، لا يمكن الاعتماد على النظام بشكل كامل، حيث أن المهاجمين يطورون أساليبهم باستمرار، مما يتطلب تحديث النموذج بشكل دوري.

6.6 التحديات التي واجهت المشروع

خلال تنفيذ المشروع، تم مواجهة عدد من التحديات، من أهمها:

- صعوبة الحصول على بيانات دقيقة ومحدثة
- الحاجة إلى تنظيف البيانات ومعالجتها قبل التدريب
- اختيار النموذج المناسب من بين عدة خوارزميات
- ربط النموذج بواجهة المستخدم بشكل عملي

ورغم هذه التحديات، تم التغلب عليها من خلال البحث والتجربة المستمرة.

7.6 التوصيات والأعمال المستقبلية

استنادًا إلى ما أظهرته النتائج من نقاط قوة وبعض القيود، يمكن تقديم التوصيات التالية:

1.7.6 تحسينات مقترحة

- **توسيع البيانات الخاصة بالروابط الشرعية القصيرة:**
من المفيد إضافة المزيد من الأمثلة لمواقع مشروعة قصيرة جدًا مثل المواقع العالمية المعروفة، حتى يتعلم النموذج التمييز بينها وبين الروابط التصيدية .
- **إضافة ميزات يدوية (Feature Engineering):**
يمكن دمج خصائص مثل طول الرابط، وعدد النطاقات الفرعية، ووجود HTTPS، وتاريخ النطاق، وسمعة الموقع مع التمثيل النصي الحالي، ضمن نموذج متعدد المدخلات .

2.7.6 تحسينات متوسطة المدى

- **تجربة نماذج أكثر تقدمًا:**
يمكن اختبار نماذج مثل Transformers أو نماذج هجينة تجمع بين النص والميزات اليدوية، ومقارنة نتائجها مع النموذج الحالي.

- **التحليل الديناميكي للمحتوى:**
يمكن فحص محتوى الصفحة نفسها بعد فتح الرابط في بيئة آمنة، والتحقق من النماذج، والنصوص، والشعارات، وسلوك JavaScript، مما قد يعزز دقة الكشف.
- **ناء Ensemble Models:**
يمكن دمج مخرجات أكثر من نموذج، مثل CNN-BiLSTM وXGBoost، للحصول على قرار نهائي أكثر متانة.

3.7.6 توصيات بحثية مستقبلية

- دراسة تأثير حجم البيانات وجودتها على الأداء بشكل منهجي .
- تقييم النموذج في بيئة آنية لمعرفة مدى قابليته للاستخدام العملي .

8.6 خلاصة الفصل

في هذا الفصل تم عرض وتحليل النتائج التي تم الحصول عليها من النموذج المستخدم في كشف الروابط الاحتيالية. وقد أثبتت النتائج أن النظام قادر على تحقيق أداء جيد، مما يدعم فكرة استخدام الذكاء الاصطناعي في هذا المجال.

كما تم مناقشة التحديات التي واجهت المشروع، مما يعطي تصورًا واضحًا عن الجوانب العملية للتنفيذ.

الخاتمة

في ختام هذا المشروع، تم تناول مشكلة هجمات التصيد الاحتيالي التي تُعد من أبرز التهديدات الأمنية في العصر الرقمي، حيث تستهدف المستخدمين عبر روابط مزيفة تهدف إلى سرقة البيانات والمعلومات الحساسة. وقد سعى هذا المشروع إلى تقديم حل عملي يعتمد على تقنيات الذكاء الاصطناعي لتحليل الروابط الإلكترونية والتفريق بين الروابط الآمنة والخبيثة.

من خلال العمل على جمع البيانات ومعالجتها، واستخدام مجموعة من خوارزميات التعلم الآلي والتعلم العميق، تم التوصل إلى نموذج قادر على تحقيق دقة عالية في الكشف عن روابط التصيد. كما تم تطوير واجهة استخدام بسيطة تُمكّن المستخدم من إدخال الرابط والحصول على نتيجة فورية، مما يعزز من قابلية استخدام النظام في الواقع العملي.

وقد أظهرت النتائج أن الاعتماد على النماذج المتقدمة، خاصة النماذج الهجينة، يساهم بشكل كبير في تحسين دقة الكشف وتقليل الأخطاء مقارنة بالطرق التقليدية. كما بيّن المشروع أهمية الاستفادة من تحليل خصائص الروابط وأنماطها للوصول إلى نتائج أكثر موثوقية.

ورغم النتائج الإيجابية التي تم تحقيقها، إلا أن المشروع يظل خطوة أولية يمكن تطويرها مستقبلاً، خاصة مع التطور المستمر في أساليب الهجمات الإلكترونية. ومن هنا تبرز أهمية العمل على تحديث النموذج بشكل دوري، وتوسيعه ليشمل أنواعًا أخرى من التهديدات.

في النهاية، يمثل هذا المشروع نموذجًا تطبيقيًا يوضح كيف يمكن توظيف تقنيات الذكاء الاصطناعي في تعزيز الأمن السيبراني، ويُسهم في رفع مستوى الحماية الرقمية للمستخدمين، مما يجعله أساسًا يمكن البناء عليه في تطوير أنظمة أكثر تقدمًا في المستقبل.

المراجع

[1] https://phishing.org/history-of-phishing.html

[2] https://www.geeksforgeeks.org/types-of-phishing-attacks-and-how-to-identify-them

[3] https://www.studysmarter.co.uk/explanations/computer-science/cybersecurity-in-computer-science/phishing

[4] https://www.csoonline.com/article/514515/what-is-phishing-examples-types-and-techniques.html

[5] Check Point Software Technologies, What is URL Phishing?. [Online]. Available: https://www.checkpoint.com

[6] OWASP, Testing for Unvalidated Redirects and Forwards (WSTG-INPV-10), Web Security Testing Guide. 2023. [Online]. Available: https://owasp.org

[7] Wired, "Why that HTTPS lock icon doesn’t always mean 'safe'," Oct. 2023. [Online]. Available: https://www.wired.com

[8] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," ACM Computing Surveys, vol. 52, no. 1, pp. 1–36, 2019.

[9] Bu, S.J.; Cho, S.B. Deep Character-Level Anomaly Detection Based on a Convolutional Autoencoder for Zero-Day Phishing Url Detection. Electron 2021, 10, 1492.

[10] Kang, J.M.; Lee, D.H. Advanced White List Approach for Preventing Access to Phishing Sites. In Proceedings of the 2007 International Conference on Convergence Information Technology (ICCIT 2007), Gwangju, Republic of Korea, 21–23 November 2007.

[11] Fu, A.Y.; Liu,W.; Deng, X. Detecting PhishingWeb Pages with Visual Similarity Assessment Based on Earth Mover’s Distance (EMD). IEEE Trans. Dependable Secur. Comput. 2006, 3, 301–311.

[12] Cao, Y.; Han,W.; Le, Y. Anti-Phishing Based on Automated Individual White-List. In Proceedings of the ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 27–31 October 2008.

[13] Oest, A.; Safei, Y.; Doupe, A.; Ahn, G.J.; Wardman, B.; Warner, G. Inside a Phisher’s Mind: Understanding the Anti-Phishing Ecosystem through Phishing Kit Analysis. In Proceedings of the 2018 APWG Symposium on Electronic Crime Research (eCrime), San Diego, CA, USA, 15–17 May 2018.

[14] Sharifi, M.; Siadati, S.H. A Phishing Sites Blacklist Generator. In Proceedings of the AICCSA 08–6th IEEE/ACS International Conference on Computer Systems and Applications, Doha, Qatar, 31 March–4 April 2008.

[15] Zhang, Y.; Hong, J.I.; Cranor, L.F. Cantina: A Content-Based Approach to Detecting PhishingWeb Sites. In Proceedings of the 16th International WorldWideWeb Conference (WWW2007), Banff, AB, Canada, 8–12 May 2007.

[16] Xiang, G.; Hong, J.; Rose, C.P.; Cranor, L. CANTINA+: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites. ACM Trans. Inf. Syst. Secur. 2011, 14, 1–28.

[17] Keivanloo, I.; Roy, C.K.; Rilling, J. SeByte: Scalable Clone and Similarity Search for Bytecode. Sci. Comput. Program. 2014, 95,426–444.

[18] Ozker, U.; Sahingoz, O.K. Content Based Phishing Detection with Machine Learning. In Proceedings of the 2020 International Conference on Electrical Engineering (ICEE 2020), Istanbul, Turkey, 25–27 September 2020.

[19] Abdelnabi, S.; Krombholz, K.; Fritz, M. VisualPhishNet: Zero-Day PhishingWebsite Detection by Visual Similarity. In Proceedings of the ACM Conference on Computer and Communications Security, Virtual Event, 9–13 November 2020.

[20] Chen, J.L.; Ma, Y.W.; Huang, K.L. Intelligent Visual Similarity-Based Phishing Websites Detection. Symmetry 2020, 12, 1681.

[21] Nair, S.M. Detecting Malicious URL Using Machine Learning: A Survey. Int. J. Res. Appl. Sci. Eng. Technol. 2020, 8, 2670–2677.

[22] Cui, Q.; Jourdan, G.V.; Bochmann, G.V.; Couturier, R.; Onut, I.V. Tracking Phishing Attacks over Time. In Proceedings of the 26th International WorldWideWeb Conference (WWW 2017), Perth, Australia, 3–7 April 2017.

[23] Sahingoz, O.K.; Buber, E.; Demir, O.; Diri, B. Machine Learning Based Phishing Detection from URLs. Expert Syst. Appl. 2019, 117,345–357.

[24] Alfouzan, N.A.; Narmatha, C. A Systematic Approach for Malware URL Recognition. In Proceedings of the 2022 2nd International Conference on Computing and Information Technology (ICCIT 2022), Tabuk, Saudi Arabia, 25–27 January 2022.

[25] Atimorathanna, D.N.; Ranaweera, T.S.; Devdunie Pabasara, R.A.H.; Perera, J.R.; Abeywardena, K.Y. NoFish; Total Anti-Phishing Protection System. In Proceedings of the ICAC 2020 2nd International Conference on Advancements in Computing, Colombo, Sri Lanka, 10–11 December 2020.

[26] Shah, B.; Dharamshi, K.; Patel, M.; Gaikwad, D.; Professor, A. Chrome Extension for Detecting Phishing Websites. Int. Res. J. Eng.Technol. 2020, 7, 2958–2962.

[27] Abiodun, O.; Sodiya, A.S.; Kareem, S.O. LINKCALCULATOR–AN EFFICIENT LINK-BASED PHISHING DETECTION TOOL. Acta Inform. Malaysia 2020, 4, 37–44.

[28] Wu, J.; Yang, Z.; Guo, L.; Li, Y.; Liu, W. Convolutional Neural Network with Character Embeddings for Malicious Web Request Detection. In Proceedings of the Proceedings–2019 IEEE Intl Conf on Parallel and Distributed Processing with Applications, Big Data and Cloud Computing, Sustainable Computing and Communications, Social Computing and Networking, ISPA/BDCloud/SustainCom/SocialCom 2019, Xiamen, China, 16–18 December 2019; pp. 622–627.

[29] Tang, L.; Mahmoud, Q.H. A Deep Learning-Based Framework for Phishing Website Detection. IEEE Access 2022, 10, 1509–1521.

[30] Yerima, S.Y.; Alzaylaee, M.K. High Accuracy Phishing Detection Based on Convolutional Neural Networks. In Proceedings of the ICCAIS 2020–3rd International Conference on Computer Applications and Information Security, Riyadh, Saudi Arabia, 19–21 March 2020.

[31] Athiwaratkun, B.; Stokes, J.W. Malware Classification with LSTM and GRU Language Models and a Character-Level CNN. In Proceedings of the ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing–Proceedings, New Orleans, LA, USA, 5–9 March 2017; pp. 2482–2486.

[32] Huang, Y.; Yang, Q.; Qin, J.; Wen, W. Phishing URL Detection via CNN and Attention-Based Hierarchical RNN. In Proceedings of the Proceedings–2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering, TrustCom/BigDataSE 2019, Rotorua, New Zealand, 5–8 August 2019; pp. 112–119.

[33] Mohammad, R.M.; Thabtah, F.; McCluskey, L. Predicting Phishing Websites Based on Self-Structuring Neural Network. Neural Comput. Appl. 2014, 25, 443–458.