مجلة جامعة دمشق للعلوم الهندسية
المجلد41 العدد (4) – 2025: 290-303

Damascus University Journal for engineering sciences
Vo 41 number (4) -2025: 290-303

كشف هجوم حقن SQL باستخدام Fasttext

زاهر الشامي*[1] رائوف حمدان[2]

[1*] طالب ماجستير، مهندس في قسم هندسة الحواسيب والأتمتة، كلية الهندسة الميكانيكية والكهربائية، جامعة دمشق. zaheralshami81@damascusuniversity.edu.sy

[2] دكتور، مدرس، مهندس في قسم هندسة الحواسيب والأتمتة، كلية الهندسة الميكانيكية والكهربائية، جامعة دمشق. raoufhamdan@damascusunbiversity.edu.sy

الملخص:

يندرج هجوم حقن SQL ضمن الهجمات المستندة للويب والتي تم تصنيفها مؤخراً ضمن أهم عشرة نقاط ضعف وفقاً لتقارير أهم مراكز أمن المعلومات والشبكات الدولية خلال السنوات السابقة.

تعتبر Fasttext واحدة من الأدوات المهمة في مجال معالجة اللغة الطبيعية، ومن خلال اعتمادها على مفهوم N-gram تميزت بقدرتها على تمثيل الكلمات الجديدة من خارج النص، والتعرف ليس على التشابه الدلالي بين الكلمات، بل على الترابط القواعدي فيما بينها.

في هذه الورقة تم اقتراح نموذج يعتمد على Fasttext لاستخراج السمات واستخدام خوارزميتي الانحدار اللوجستي والتعزيز المتدرج من أجل كشف هجوم حقن SQL، كما تم استخدام طرق أخرى مستخدمة في التعامل مع النصوص مثل Word2Vec وTF_IDF للمقارنة، أظهرت النتائج بعد تطبيق النموذج على مجموعتين للاختبار تفوقا واضحا لـ Fasttext على نظرائها بدقة تصل لـ99.73%، مما يجعل من النموذج المقترح النموذج الأنسب لكشف هجوم حقن SQL.

الكلمات المفتاحية: هجوم حقن SQL، الأمن السيبراني، معالجة اللغات الطبيعية، التعلم الآلي، تردد الكلمة–تردد المستند العكسي، تضمين الكلمات، النص السريع.

تاريخ الإيداع: 2023/4/2
تاريخ القبول: 2023/5/29

ISSN (online): 2789-6854
http://journal.damascusuniversity.edu.sy/index.php/index/index

SQL Injection Attack Detection Using Fasttext

Zaher Alshami*[1] Raouf Hamdan[2]

*[1]. Master's Student, Eng in the Computer Engineering and Automation Department, Faculty of Mechanical and Electrical Engineering, Damascus University.
zaheralshami81@damascusuniversity.edu.sy.

[2]. Dr, Teacher in the Computer Engineering and Automation Department, Faculty of Mechanical and Electrical Engineering, Damascus University.
RaoufHamdan@damascusunbiversity.edu.sy .

Received: 2/4/2023
Accepted: 29/5/2023

Abstract:

SQL injection attack falls under the web-based attacks, that are recently ranked among the top vulnerabilities according to the reports of the most important international network and information security centers over last years.

Fasttext is one of the important tools in the field of natural language processing (NLP), through its reliance on the concept of N-gram, it is characterized by its ability to represent new words outside the text, or co-called Out of Vocabulary (OOP), and to recognize not only the semantic similarity between words, but also the morphological association between them.

In this paper, a model based on Fasttext is proposed to extract features and use Logistic Regression and Gradient Boost algorithms to detect SQL injection attack. In Addition, two other features extraction techniques such as Word2Vec and TF_IDF are used for comparison, the results show a clear superiority of Fasttext over its counterparts with 99.73% for accuracy, which makes the proposed model the most appropriate model for detecting SQL injection attack.

Keywords: SQL Injection Attack, Cyber Security, Natural Language Processing, Machine Learning, TF-IDF, Word2Vec, Fasttext.

المقدمة:

منذ عدة وسنوات وحتى الآن لا تزال الهجمات المستندة للويب ومن ضمنها هجوم حقن SQL تتربع على قمة أخطر التهديدات الأمنية الإلكترونية عبر شبكة الإنترنت وفق احصائيات مراكز أمن المعلومات والشبكات الدولية، فقد أشار التقرير السنوي لمشروع أمان لتطبيقات الويب المفتوحة OAWSP خلال عام 2021 أن 94% من تطبيقات الويب عرضة لهجمات الحقن بشكل عام [1]، أما وكالة الاتحاد الأوربي لأمن الشبكات والمعلومات ENSIA صنفتها كثاني أخطر تهديد الكتروني لعام2020 [2]، لهذا فهي تشكل التحدي الأبرز أمام خبراء ومراكز أمن المعلومات والشبكات.

إن هجوم حقن SQL عبارة عن هجوم برمجي ضد تطبيقات الويب يتيح للمهاجم حقن تعليمات SQL في إدخالات تطبيق الويب من أجل الوصول غير الشرعي للمعلومات الموجودة في قاعدة البيانات والخادم الرئيسي للشبكة.

مع التصعيد المستمر لهذا الهجوم، واجهت أنظمة التصفية التقليدية وجدران حماية تطبيقات الويب العديد من المشكلات في السنوات الأخيرة، مما دفع الباحثين للجوء إلى تقنيات التعلم الآلي لاقتراح حلول أكثر ملاءمة للتصدي لهذا الهجوم، وقد تم إجراء العديد من الأبحاث حول استخدام خوارزميات مختلفة للتعلم الآلي، مع ذلك لا تزال الحاجة ملحة إلى تحسين النماذج التي تم تطويرها في السابق للتعامل مع هذا الهجوم، ذلك أنها لم تصل لحل شامل من حيث الدقة والشمولية في ظل نجاح المخترقين في تطوير أساليبهم لاجتياز آليات الدفاع المختلفة.

1-مفهوم هجوم حقن SQL:

يستند هجوم حقن SQL على لغة الاستعلام الهيكلية وهي اللغة المستخدمة لإجراء عمليات على قواعد البيانات والتحكم بها، حيث يحوي كل تطبيق من التطبيقات على قاعدة البيانات الخاصة به والتي تحتوي بدورها على المعلومات الحساسة للمستخدمين والشركات والعملاء والمؤسسات وما إلى ذلك.

يتم دمج محتوى مدخلات المستخدم في تطبيق الويب مع الاستعلام المحدد مسبقاً من قبل المطور لإجراء وظيفة معينة داخل قاعدة البيانات مثل (select, insert, update, etc.) ومن المفترض لهذه المدخلات أن تحوي على معلومات نصية أو رقمية، إلا أن المهاجم يحاول استغلالها في حقن تعليمات جديدة بهدف تغيير صيغة الاستعلام المسبق الذي ستندمج تلك التعليمات المضافة معه ليتحول لاستعلام آخر يعيد نتيجة مختلفة كلياً عن النتيجة التي أرادها المطور. لهذا فإن السبب الأساسي لهجوم حقن SQL هو الثقة الزائدة في البيانات المقدمة من قبل المستخدمين، حيث يقوم المبرمجون بكتابة الكود البرمجي الخاص بتطبيقات الويب دون تصفية مدخلات المستخدم أو إجراء تحقق معقول من جانب الخادم، الأمر الذي يسمح بحقن بعض تعليمات SQL في تلك المدخلات من أجل التأثير على تنفيذ الاستعلامات المحددة مسبقاً من قبل، وبالتالي تغيير صيغة الاستعلام الأصلي وتنفيذ عمليات ليس من المصرح للمهاجم تنفيذها.

2-الدراسات المرجعية:

في الآونة الأخيرة تم تطوير العديد من النماذج للتعامل مع هجمات حقن SQL استناداً إلى خوارزميات التعلم الآلي، وسيتم مناقشة البعض منها.

طرح Zhuang Chen وآخرون (2018) نموذجاً لاكتشاف هجمات حقن SQL باستخدام Word2Vec لاستخراج السمات وتمثيل الاستعلام ومن ثم تصنيفه عبر خوارزمية آلة المتجه الداعم Support Vector Machine، بلغ معدل الكشف الإيجابي 95.4%، تحوي مجموعة البيانات في هذه الورقة على 2000 عينة فقط، 1000 عينة سليمة و1000 خبيثة، أوصت الدراسة بتدريب النموذج على مجموعة بيانات تضم عدد أكبر من الاستعلامات الخبيثة والسليمة من أجل الحصول على أداء فعال [3].

اقترح Hoang (2020) نموذجاً باستخدام طريقة TF-IDF لاستخراج السمات وخوارزمية شجرة القرار Decision Tree من أجل التصنيف، وذلك للكشف عن أربع أنواع من الهجمات (SQLi, XSS, CMDi, and path traversal)، بلغت الدقة الإجمالية للنموذج 98.56٪، بلغ حجم مجموعة البيانات التي تضم الهجومات الأربعة 31,067 [4].

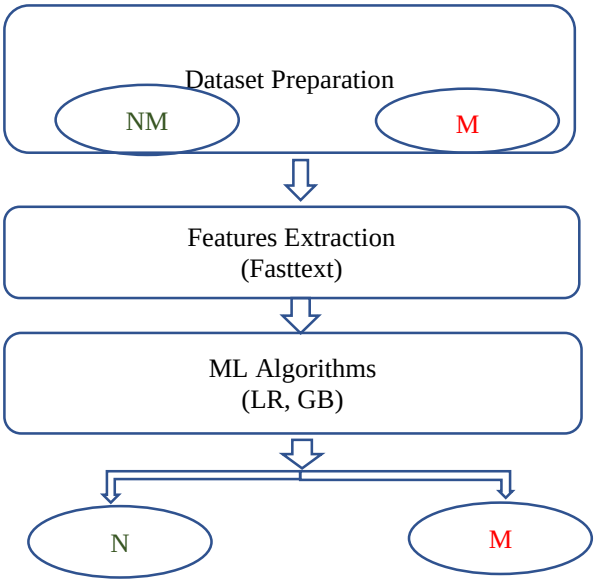
استخدم Ding Chen وآخرون (2020) طريقة Word2Vec لاستخراج السمات وخوارزميتي MLP وCNN من أجل التصنيف، حققت خوارزمية MLP الدقة الأفضل 98.57٪، تكونت مجموعة التدريب من 25487 عينة خبيثة و24500 عينة حميدة، أما مجموعة الاختبار فقد ضمت 8000 عينة تم توزيعها مناصفة بين عينة حميدة وعينة خبيثة [5].

أعطت النماذج المستخدمة في الأبحاث السابقة نتائج جيدة، مع ذلك ثمة سبب جوهري قد يقلل من فعاليتها في بعض الأحيان، ذلك أن معظم الطرق المستخدمة في استخراج السمات مثل Word2Vec وTF-IDF لا تستطيع تمثيل الكلمات التي لم تتدرب بشكل مسبق عليها خلال مرحلة التدريب، الأمر الذي قد يحد من دقة النموذج وفعاليته عند التعميم والتطبيق العملي على عينات جديدة، تسمى هذه الكلمات Out Of Vocabulary اختصاراً OOP وغالباً ما يتم الاستعاضة عنها عند تطبيق النموذج على أرض الواقع بقيمة فارغة أو متجه فارغ Null Vector.

لذا ولتلافي مثل هذا السيناريو تم في هذه الورقة استخراج السمات باستخدام تقنية حديثة في مجال معالجة اللغة الطبيعية تستطيع التعامل مع كلمات OOV، تدعى هذه الطريقة Fasttext كما تم أيضا استخدام الطرق السابقة وذلك من أجل مقارنة أدائها مع أداء Fasttext، حيث أثبتت Fasttext كفاءتها العالية كما سيتضح ذلك في الفقرات اللاحقة.

3-النموذج المقترح:

الشكل (1) يوضح بنية النموذج المقترح وفقاً للطرق والخوارزميات التي سيتم توضيحها في فقرات لاحقة.



الشكل (1) بنية النموذج المقترح.

3-1 تحضير مجموعة البيانات:

شكل إيجاد مجموعة البيانات المناسبة للبحث تحدياً نظراً لعدم توفر مجموعة بيانات معيارية يمكن الوصول إليها والاعتماد عليها في استخراج السمات وتدريب خوارزميات التعلم الآلي، مما اقتضى إجراء بحث مكثف وقضاء وقت طويل نسبياً في سبر العديد من المصادر والتأكد من احتوائها على العينات المناسبة، وذلك بهدف تكوين مجموعة بيانات شاملة وكافية تتضمن مختلف أنواع هجوم حقن SQL، وبالاعتماد على المصادر [6] و[7] و[8] و[9]، تم تشكيل مجموعة بيانات نهائية لهذه الورقة من خلال دمج العينات الواردة في هذه المصادر معاً.

بعد الانتهاء من مرحلة الجمع تم الانتقال لمرحلة التجهيز، حيث تم تنظيف جميع الاستعلامات والنصوص الصريحة وتحويلها إلى حالة الأحرف الصغيرة lower case = true، ومن ثم فرزها إلى عينة خبيثة M وعينة حميدة NM، كذلك تم التخلص من الاستعلامات أو النصوص المتكررة التي قد تؤثر على أداء النموذج وتجعله ينحاز في النتائج، والاحتفاظ بجميع

الأقواس وعلامات الترقيم التي يمكن أن ترد ضمن العينة الواحدة نظراً لأن الاستعلام الخبيث عادة ما تكثر فيه الأقواس وعلامات الترقيم على خلاف النص الصريح والاستعلام الحميد، والتي قد يشكل كثرة تكرارها دليلاً يساعد النموذج على التصنيف، في النهاية وبعد إنجاز كل الخطوات السابقة تم حفظ مجموعة التدريب ضمن ملف واحد بلاحقة CSV.

بما أن مدخلات المستخدم في تطبيقات الويب ستحوي خلال الاستثمار الفعلي على عدد أكبر من العينات الحميدة مقارنةً بالعينات الخبيثة، فقد وقع الخيار على أن تكون نسبة العينات الحميدة ضمن مجموعة البيانات ضعف نسبة العينات الخبيثة وذلك لتلافي الانحياز في النتائج ولتحقيق قدر تقريبي من الانسجام مع الواقع الفعلي للسياق الذي سيتم توظيف النموذج فيه لاحقاً، ومن أجل الإقلال من حالات False Positive وهي الحالات التي يقوم فيها النموذج بتصنيف عينة حميدة بشكل خاطئ على أنها عينة خبيثة، فقد تم إضافة عدد من الاستعلامات السليمة إلى مجموعة التدريب واعتبارها عينات حميدة، وذلك من أجل منح النموذج قدرة أفضل على التمييز بين العينة الحميدة والعينة الخبيثة والإقلال قدر ما أمكن من حالات FP.

بلغ عدد العينات الإجمالي في مجموعة البيانات المستخدمة 44036 عينة، مقسمة إلى 14211 عينة خبيثة و29825 عينة حميدة تضم كل من نص صريح واستعلام سليم، الجدول (1) يوضح بنية مجموعة البيانات المستخدمة في هذه الورقة.

الجدول (1) مجموعة البيانات المستخدمة.

Malicious Samples	Non-Malicious Samples		Total Samples
	Plain Text	Benign	
	18230	11595	
14211	29825		44036

3-2 استخراج السمات:

حتى تتمكن خوارزميات التعلم الآلي من التعامل مع الاستعلامات والتدرب على تصنيفها إلى استعلام خبيث أو حميد لا بد في البداية من تحويل كل استعلام إلى قيم رقمية يتم التعبير عنها من خلال شعاع رقمي.

إن أي استعلام مصاغ بلغة SQL هو في نهاية المطاف عبارة عن نص طبيعي يتم ترجمته من قبل أنظمة إدارة قواعد البيانات، وبالتالي يمكن استخدام تقنيات معالجة اللغة الطبيعية في استخراج السمات من الاستعلامات وتمثيلها رقمياً واستخدام هذا التمثيل لاحقاً لتدريب النموذج.

ستتناول هذه الفقرة الطرق المستخدمة في استخراج السمات خلال الأبحاث السابقة TF_IDF وWord2Vec، بالإضافة للطريقة الجديدة المستخدمة في هذه الورقة Fasttext.

3-2-1 تردد الكلمة-تردد المستند العكسي (TF-IDF):

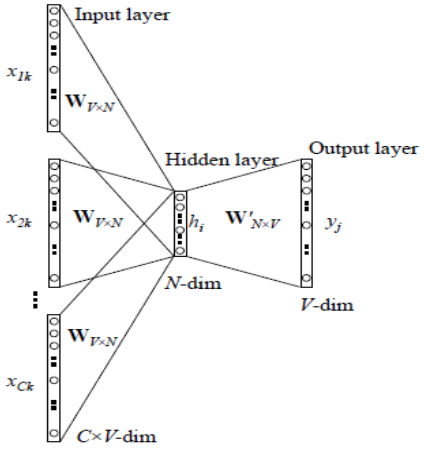
هذه الطريقة عبارة عن مقياس إحصائي يتم من خلاله تشكيل قاموس بالكلمات وحساب تكرارها مع إضافة وزن يحدد مدى أهمية الكلمة بالنسبة لجملة الاستعلام التي تحتويها، تتجلى هذه الأهمية من خلال التركيز على عدد مرات ظهور الكلمة ليس في جملة الاستعلام الواحدة فحسب، بل في بقية الجمل أيضاً، فالكلمات التي تتكرر كثيراً في معظم الاستعلامات ستحصل على وزن منخفض، بالمقابل فإن الكلمات الأقل تكراراً سيكون لها وزن مرتفع [10].

3-2-2 تضمين الكلمات (Word2Vec):

إن الهدف من Word2Vec هو التدرب على تعلم أوجه التشابه بين الكلمات من خلال مجموعة كبيرة من النصوص، وتعتمد على حقيقة مفادها أنه يمكن التنبؤ بكلمة معينة في جملة ما من خلال السياق الذي يمكن أن تأتي به، وبالتالي فإن الكلمات التي تأتي في نفس السياقات تحمل معاني متقاربة من الناحية الدلالية.

تقوم Word2Vec بتمثيل كل كلمة بشعاع رقمي، ومن خلال حساب المسافة بين شعاعين يمكن معرفة درجة التقارب والتشابه بين الكلمات [11].

البنية المستخدمة في هذه الورقة تسمى CBOW، تتكون هذه البنية من شبكة عصبونية بسيطة من ثلاث طبقات، طبقة الدخل وطبقة خفية واحدة وطبقة الخرج، تشكل طبقة الدخل كلمات السياق التي تحوي على الكلمة المراد التنبؤ بها أو الكلمة الهدف، أما الخرج فهو الكلمة الهدف نفسها، بينما يجسد عدد خلايا الطبقة الخفية عدد أبعاد الشعاع الرقمي المطلوب للكلمة، الشكل (2) يوضح بنية CBOW Word2Vec.



الشكل (2) بنية CBOW Word2Vec [12].

3-2-3 النص السريع (Fasttext):

تشبه طريقة Fasttext طريقة Word2Vec من حيث المبدأ، إلا أنها تتميز عنها باعتمادها على مفهوم N-gram، ففي طريقة Word2Vec يتم تقسيم الجملة إلى مجموعة من الكلمات المستقلة التي تتكون منها، والتعامل مع كل كلمة مستقلة على حدى، أما خوارزمية Fasttext تقوم بتجزيء الكلمة المستقلة نفسها لعدد من الكلمات الفرعية وفق عدد محدد من التقسيمات N، ومن ثم يتم التعامل مع الكلمات المستقلة وفقاً لهذه الكلمات الفرعية المشتقة منها [13].

في الوقت الذي تفشل فيه طريقة Word2Vec تماماً في تمثيل الكلمات الجديدة التي لم تتدرب عليها من قبل ولا تستطيع فهم الترابط القواعدي بين الكلمات، يتيح مفهوم N-gram في طريقة Fasttext تمثيل عدد أكبر من الكلمات المستقلة حتى وإن كانت من خارج سياق النص الذي تم التدرب عليه بشكل مسبق، على اعتبار أن الكلمات في قسم لا بأس به من اللغات مثل اللغة العربية والإنكليزية والألمانية تُشتق عملياً من بعضها، كما أنه يسمح بالتعرف ليس على المعاني الدلالية للكلمات فحسب، بل على الترابط القواعدي فيما بينها كذلك الأمر.

يمكن توضيح الصورة أكثر من خلال تطبيق عملي:

بفرض وجود نص ترد فيه الكلمات الآتية (injection, injected, injector)، ستقوم Word2Vec بتمثيل كل كلمة على حدى بشكل مستقل وفقاً للسياق الذي جاءت فيه مع تجاهل تام للترابط القواعدي الذي قد يجمعها بالكلمات الأخرى، علاوة على ذلك فإنها ستفشل تماماً في تمثيل كلمة جديدة من خارج النص مثل كلمة (inject) على الرغم من أنها متواجدة كتسلسل محرفي ضمن كلمات النص الأساسي، أما في Fasttext يساعد تقسيم الكلمة إلى مقاطع فرعية وفق مفهوم N-gram على تلافي مثل هذه الثغرات.

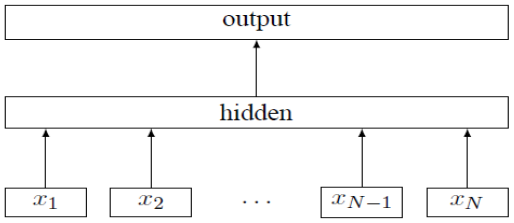
يوضح الجدول (2) تقسيم الكلمات السابقة إلى كلمات فرعية عندما يكون N=3، حيث يعتمد التقسيم على وضع الكلمة ضمن أقواس من نوع <> والتي تعتبر جزءا من الكلمة ومن ثم تقسيمها لكلماتها الفرعية.

الجدول (2) مفهوم N-gram.

Words	3-gram
<injection>	<in, inj, nje, jec, ect, cti, tio, ion, on>
<injected>	<in, inj, nje, jec, ect, cte, ted, ed>
<injector>	<in, inj, nje, jec, ect, cto, tor, or>

يبين الجدول اشتراك الكلمات السابقة بالمقاطع الفرعية الخمسة الأولى، يسمح هذا التقسيم لخوارزمية Fasttext بكشف الترابط القواعدي فيما بينها بصورة دقيقة، كما أنه يعطيها فرصة لتمثيل كلمات جديدة من خارج النص مثل (inject) في حال كانت الخوارزمية مدربة على واحدة من مشتقاتها فقط.

مؤخراً قام مختبر الذكاء الصنعي في شركة Facebook ببرمجة طريقة Fasttext ضمن مكتبة مفتوحة المصدر متاحة للأغراض العامة، تقدم هذه المكتبة نموذجين أحدهما مستخدم لتضمين الكلمات والآخر لتضمين الجمل وتصنيفها.



الشكل (3) بنية Fasttext [14].

في هذه الورقة تم الاعتماد على خوارزمية Fasttext المستخدمة في تضمين الجمل وتصنيفها، تتألف هذه البنية كما يوضحها الشكل (3) من شبكة عصبونية بسيطة مكونة من ثلاث طبقات، تمثل طبقة الدخل الكلمات الفرعية التي تتألف منها الجملة(X1, X2, XN-1, XN)، يتم تشكيل شعاع واحد من هذه المقاطع إما بطريقة الضم أو أخذ المتوسط، ومن ثم يتم ضربها بمصفوفة الأوزان بين طبقة الدخل والطبقة الخفية، تستقبل الطبقة الخفية التي تجسد عدد أبعاد الشعاع المطلوب لتمثيل الجملة هذا الدخل الموزون، وبدلاً من استخدام تابع تفعيل ضمن الطبقة الخفية كما هو شائع في الشبكات العصبونية، يتم الاكتفاء بتمرير هذا الدخل الموزون لطبقة الخرج بعد ضربه بمصفوفة الأوزان بين الطبقة الخفية وبين طبقة الخرج (تسمى هذه الخطوة Linear)، في طبقة الخرج يتم تطبيق تابع Softmax لحساب احتمال انتماء الجملة لصف معين [14].

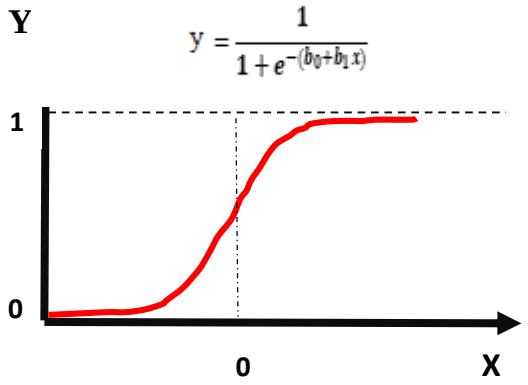
تجسد Fasttext خوارزمية تصنيف كاملة بإمكانها حساب احتمال انتماء الجملة لصف معين، وبنفس الوقت فإنها تدعم استخراج السمات لجملة معينة بعد تدريب الخوارزمية عبر استخدام الوظيفة get_sentence_vector، لذا ومن أجل إتاحة الفرصة لمقارنة هذه الخوارزمية مع طرق استخراج السمات الأخرى تم في هذه الورقة الاستعانة بها لاستخراج السمات فقط.

3-3 خوارزميات التصنيف:

إن الهدف من هذا البحث هو تصنيف الاستعلام إلى استعلام خبيث أو حميد، ومن أجل الحصول على أفضل النتائج والمقارنة بينها تم الاعتماد على خوارزميتين للتصنيف، خوارزمية LR التي تتميز بقدرتها على التعامل مع البيانات القابلة للفصل خطياً وخوارزمية GB التي بوسعها إيجاد الأنماط غير الخطية بين البيانات.

3-3-1 الانحدار اللوجستي (LR):

تُعتبر خوارزمية الانحدار اللوجستي نموذج إحصائي يعتمد على نمذجة المتغيرات وفق تابع رياضي يدعى Sigmoid، وظيفة هذا التابع اشتقاق العلاقة بين المتغيرات التي تمثل السمات والخرج الذي يمثل صف معين، إذ يقوم التابع بالاعتماد على قيم المتغيرات بإعطاء قيمة تقع ضمن المجال بين 0 و1 من أجل التنبؤ باحتمال انتماء الخرج لهذا الصف أو ذاك. تتميز بالبساطة والسرعة الكبيرة في تصنيف البيانات القابلة للفصل خطياً، يوضح الشكل (4) شكل تابع Sigmoid باللون البرتقالي بالنسبة للمتغير x.



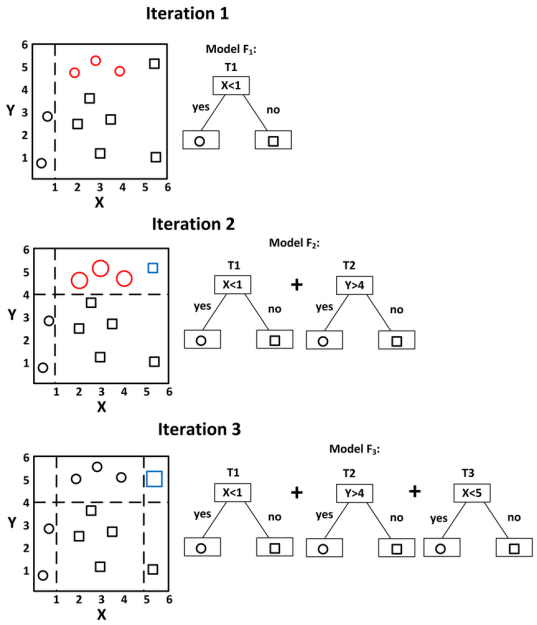
الشكل (4) تابع Sigmoid.

3-3-2 التعزيز المتدرج (GB):

وهي خوارزمية تعلم آلي تنتمي للنهج التجميعي الذي يقوم بتحويل المتعلم الضعيف Weak Learner إلى متعلم قوي Strong Leaner عن طريق التجميع Ensemble، إن نموذج

المتعلم الضعيف هو النموذج القادر على التنبؤ بمعدل خطأ أفضل بقليل من التخمين العشوائي، الفكرة الرئيسية للتعزيز هي إضافة نماذج متعلم ضعيف بشكل متتابع مع كل تكرار وتدريبه على تقليل الخطأ الناتج عن مجموعة المتعلمات الضعيفة السابقة والتي تم تدريبها حتى الآن.

تعتمد الخوارزمية على أخذ مجموعة من النماذج ذات الفعالية الضعيفة والتي تسمى بالمتعلم الضعيف وتدريبها الواحد تلو الآخر، وعند كل تكرار جديد يولي كل نموذج لاحق اهتمامه لنقاط البيانات التي أخطأ النموذج السابق في تصنيفها، وتتكرر هذه العملية إلى أن تصبح الدقة الإجمالية للمجموعة ككل أفضل من دقة المتعلمات الضعيفة التي تعمل بطريقة منفردة [15] [16].



الشكل (5) خوارزمية التعزيز المتدرج [17].

يبين الشكل (5) الطريقة التي تعمل بها خوارزمية التعزيز المتدرج باستخدام شجرة القرار كمتعلم ضعيف، وهي الحالة الأكثر استخداماً عند استخدام هذه الخوارزمية، حيث يمثل الحرف F نموذج المتعلم الضعيف في كل تكرار Iteration، الهدف النهائي هو تصنيف الدوائر والمربعات الموجودة في فضاء ثنائي الأبعاد(X, Y)، في البداية يقوم النموذج F1 عبر شجرة القرار T1 بتصنيف الرموز التي تأخذ قيم أصغر من 1 على المحور X كدوائر بينما يتم تصنيف القيم الأكبر من 1 كمربعات، من الواضح أن النموذج F1 قد صنف بشكل خاطئ الدوائر الثلاث المميزة باللون الأحمر على أنها مربعات، هذه الدوائر الثلاث هي أخطاء النموذج F1، في مثل هذه الحالة يتم تخصيص أوزان أعلى لهذه الدوائر الثلاث (تبدو أكبر من الرموز الأخرى) بحيث يمكن للتكرار الثاني إنشاء شجرة قرار بسيطة أخرى للتنبؤ بها بشكل صحيح، هذا ما يقوم به النموذج المحدث F2 وهو مزيج من الأشجار T1 وT2، وعلى الرغم من ارتكابه خطأ في تصنيف المربع الأزرق، إلا أنه تم بالفعل تقليل الخطأ الإجمالي لـ F2 مقارنة بـ F1.

بعد ذلك يتم تقليل أوزان الدوائر الحمراء الثلاث المصنفة بشكل صحيح بواسطة F2 إلى المستوى الطبيعي، ويزداد وزن المربع الأزرق المصنف بشكل خاطئ، خلال التكرار الثالث يستطيع النموذج F3 الذي يتألف من مجموع الأشجار T1 وT2 وT3 فصل المربع الأزرق عن جميع الدوائر وتصنيف جميع الرموز بنجاح [17].

في الواقع أشارت بعض الدراسات إلى قوة وكفاءة النهج التجميعي بشكل عام وإلى خوارزمية التعزيز المتدرج بشكل خاص، فقد جاء ترتيب الخوارزمية في المقدمة من بين 13 خوارزمية أخرى، وذلك خلال تطبيق هذه الخوارزميات على 163 مجموعة بيانات تابعة لمسائل تصنيف مختلفة [18].

4-خطوات بناء النموذج المقترح:

اعتمد بناء النموذج على حاسب محمول LENOVO ideapad 110 بمعالج AMD Radeon R4 وذاكرة رئيسية تبلغ 8 جيجابايت ونظام تشغيل Windows 10 Home، وقد تم استخدام لغة Python كبيئة برمجية مناسبة في تصميم النموذج نظراً لمرونتها واحتوائها على مكتبات متنوعة تساعد في إنجاز النماذج المستندة إلى تعلم الآلة، فضلاً عن توفر شروحات تفصيلية على الإنترنت لتعلم اللغة، المكتبات المستخدمة

هي:(numpy, pandas, nltk, matplotlib, sklearn, genism, fasttext).

يمكن توضيح أهم الإجراءات المتبعة خلال التصميم بالخطوات الآتية:

قراءة ملف مجموعة البيانات: قراءة الملف ووضعه ضمن جدول pandas من أجل التعامل مع العينات، بالإضافة للتأكد مرة أخرى من استخدام حالة الأحرف الصغيرة وعدم احتواء المجموعة على فراغات أو عينات متكررة.

إجراء عملية تجزئة للمفردات Tokenization: تحويل كل عينة في جدول مجموعة البيانات إلى قائمة بالكلمات أو المفردات tokens الواردة فيها، ستشكل هذه الخطوة الأساس الذي ستقوم طرق استخراج السمات بالاستناد عليه من أجل توليد السمات، تم كذلك في هذه الخطوة حساب عدد الكلمات أو المفردات الإجمالي ضمن جدول مجموعة البيانات، بلغ عدد الكلمات الإجمالي 38872 كلمة.

تقسيم مجموعة البيانات: بعثرة العينات ضمن جدول مجموعة البيانات بشكل عشوائي ومن ثم تقسيمها إلى مجموعتين فرعيتين: 67% لمجموعة التدريب، و33% لمجموعة الاختبار، الجدول (3) يوضح عدد العينات في كل من مجموعة التدريب ومجموعة الاختبار.

الجدول (3) مجموعة التدريب ومجموعة الاختبار.

Dataset	Malicious Samples	Non-Malicious Samples	Total Samples
Train	9450	20054	29504
Test	4761	9771	14532
Sum	14211	29825	44036

تجهيز تقنيات استخراج السمات وخوارزميتي التعلم الآلي: بالاستعانة بمكتبة sklearn بالنسبة لـ TF_IDF وLR وGB، وبمكتبة genism بالنسبة لـWord2Vec ، وبمكتبة fasttext بالنسبة لـFasttext تم بناء وتجهيز الصفوف المناسبة.

ضبط المعاملات الفائقة: بشكل عام يشكل ضبط المعاملات مرحلة مهمة ومحيرة بعض الشيء أحياناً، نظراً لأن بعض تلك المعاملات قد يأخذ نطاقاً واسعاً من القيم، مع ذلك ثمة بعض الأساليب المنهجية التي قد تساهم في معرفة القيمة أو المجال الأنسب للمعامل.

يتيح الصف GridSearchCV في مكتبة sklearn إيجاد القيم الأفضل للمعاملات الخاصة لكل من الخوارزمية وطريقة استخراج السمات معاً، حيث يمكن جمع الخوارزمية المستخدمة مع طريقة استخراج السمات سويةً من خلال Pipeline واختيار مجال أو قيم محددة لأهم

المعاملات في كل منهما، ومن ثم إجراء بحث شامل عن القيم الأنسب للمعاملات التي تعيد الدقة الأفضل.

يبين الجدول (4) التطبيق العملي لمعرفة القيم الأنسب لأهم المعاملات في خوارزمية GB وطريقة Word2Vec لاستخراج السمات وذلك عند cv=3.

الجدول (4) ضبط معاملات كل من GB وWord2Vec.

الصف	المكتبة	المعامل	القيم المحددة	القيمة الأفضل
Word2Vec	gensim	vector_size	(10، 50، 100، 150)	100
		epochs	(25، 50، 100، 200)	200
GB	sklearn	n_estimators	(10، 50، 100)	100
		learning_rate	(0.01، 0.05، 0.1)	0.1

بالنسبة لـWord2Vec يمثل vector_size أبعاد الشعاع الرقمي المطلوب للكلمة، بينما يمثل epochs عدد التكرارات اللازم للوصول للقيمة الدنيا لتابع الخسارة، أما في خوارزمية GB يمثل n-estimators عدد الأشجار المراد تطبيقها خلال التعزيز، بينما يمثل learning_rate معدل الاقتراب من القيمة الدنيا لتابع الخسارة.

تدريب النموذج: بعد الانتهاء من بناء وتجهيز جميع الصفوف والوظائف البرمجية اللازمة لإنجاز الخطوات المشار إليها أعلاه، تم تدريب النموذج على مجموعة التدريب ومن ثم تطبيق النموذج المدرب على مجموعة الاختبار وحساب مقاييس الأداء Performance Metrics.

حساب مقاييس الأداء: تم استخدام مقاييس الأداء (Accuracy Precision, Recall, F1-Score) لتقييم النموذج المقترح وفقاً لمصفوفة الارتباك Confusion Matrix، يبين الجدولان (5) و (6) مصفوفة الارتباك والمقاييس التي يتم حسابها بناءً على هذه المصفوفة على التوالي.

الجدول (5) مصفوفة الارتباك.

True Positive (TP) الاستعلامات الخبيثة التي تم تصنيفها بشكل صحيح على أنها خبيثة	False Negative (FN) الاستعلامات الخبيثة التي تم تصنيفها بشكل خاطئ على أنها حميدة
False Positive (FP) الاستعلامات الحميدة التي تم تصنيفها بشكل خاطئ على أنها خبيثة	True Negative (TN) الاستعلامات الحميدة التي تم تصنيفها بشكل صحيح على أنها حميدة

الجدول (6) مقاييس الأداء.

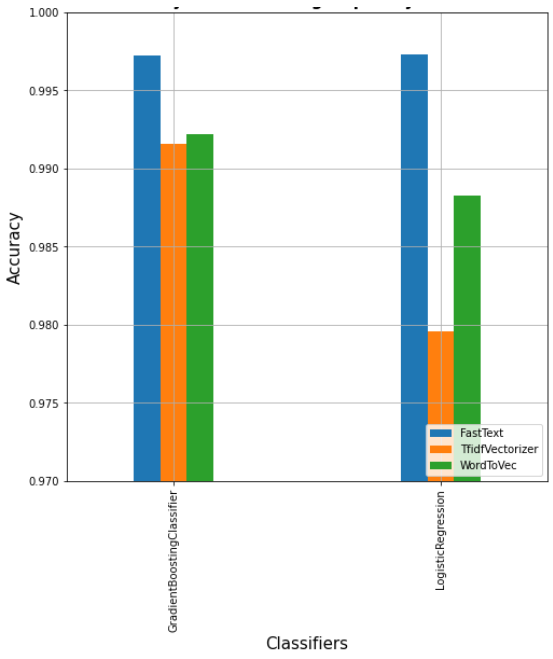
المقياس	الوصف
$\mathbf{Accuracy} = \frac{\mathbf{TP+TN}}{\mathbf{TP+TN+FP+FN}}$	معدل الكشف الإجمالي بالنسبة لجميع العينات الخبيثة والحميدة في مجموعة الاختبار، يُعتبر المقياس الأساسي لقياس أداء النموذج بشكل عام.
$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP+FN}}$	معدل الكشف الإيجابي بالنسبة للعينات الخبيثة في مجموعة الاختبار، يعكس هذا المقياس مدى قدرة النموذج على تصنيف الاستعلامات الخبيثة بشكل صحيح.
$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP+FP}}$	معدل الكشف الإيجابي بالنسبة للعينات التي صنفها النموذج على أنها خبيثة (وقد لا تكون جميعها كذلك)، كلما كان هذا المؤشر أعلى كلما كان هذا دليلاً على أن عدد الاستعلامات الحميدة المصنفة بشكل خاطئ على أنها خبيثة قليل.
$\mathbf{F1\_Score} = 2 \times \frac{\mathbf{Recall} \times \mathbf{Precision}}{\mathbf{Recall+Precision}}$	المتوسط التوافقي لـ Recall وPrecision، يُعتبر المقياس الأهم في تقييم أداء النموذج ومدى قدرته على التعميم.

5- النتائج والمناقشة:

يبين الجدول (7) النتائج الكاملة التي تم الحصول عليها عند تطبيق النموذج المدرب على مجموعة الاختبار، مرتبة بشكل تنازلي وفق مقياس الدقة Accuracy.

الجدول (7) النتائج العملية للنموذج المقترح.

Classifier	Features Extraction Tech	Accuracy	Precision	Recall	F1-Score
LR	Fasttext	0.9973	0.9983	0.9935	0.9959
GB	Fasttext	0.9972	0.9981	0.9935	0.9958
GB	Word2Vec	0.9922	0.9893	0.9868	0.9880
GB	TF-IDF	0.9915	0.9942	0.9798	0.9870
LR	Word2Vec	0.9882	0.9809	0.9832	0.9821
LR	TF-IDF	0.9796	0.9784	0.9588	0.9685



الشكل (6) دقة النموذج المقترح.

الشكل (6) يستعرض النتائج ذاتها بطريقة رسومية بعد تجميعها وفقاً للمصنف المستخدم مع أخذ مقياس Accuracy بعين الاعتبار.

بالاطلاع على النتائج يمكن ملاحظة الآتي:

أ–حقق النموذج المقترح نتائج جيدة، الدقة الأفضل بلغت 99.73% عند استخدام طريقة Fasttext لاستخراج السمات وخوارزمية LR كمصنف.

ب–تفوقت طريقة Fasttext على جميع طرق استخراج السمات وحققت النتائج الأفضل مع خوارزميتي التصنيف المستخدمة وذلك من حيث Accuracy وF1-Score، تفسير ذلك يكمن في اعتمادها على مفهوم N-gram الذي أعطى فرصة لتمثيل كلمات OOV وهي الكلمات الجديدة ضمن مجموعة الاختبار والتي لم يتدرب عليها النموذج خلال مرحلة التدريب (غير موجودة ضمن مجموعة التدريب)، حيث بلغ عددها في مجموعة الاختبار 9513 كلمة.

ج–فعالية وثبات نتائج طريقة Fasttext بالمقارنة مع نتائج طرق استخراج السمات الأخرى يعطي فكرة أن الطريقة التي يتم بها استخراج السمات من النص تُعتبر الخطوة المركزية الأهم والأكثر حساسية في مسائل معالجة اللغة الطبيعية بشكل عام، وهذا مؤشر مهم عند دراسة هجمات أخرى تعتمد على النص في آلية عملها.

د–على الرغم من قدرة خوارزمية GB على إيجاد العلاقات غير الخطية والأنماط المعقدة بين البيانات والتي لا تستطيع الخوارزميات الخطية مثل LR إيجادها، إلا أن أداء الخوارزميتين بدا متقارباً لحد كبير تبعاً لطريقة استخراج السمات المستخدمة.

6- اختبار النموذج على عينات جديدة:

من أجل التأكد من قدرة النموذج المقترح على التعميم والتطبيق العملي، تم اللجوء لمرحلة اختبار إضافية لم تُستخدم من قبل في الأبحاث السابقة، حيث تم اختباره على عينات جديدة أخرى تم جمعها بنفس طريقة جمع وتحضير مجموعة البيانات الأساسية.

لتحقيق ذلك تم الاعتماد على أداة Libinjection للحصول على عينات خبيثة، هذه الأداة معروفة في مجال الأمن السيبراني وهي أداة مفتوحة المصدر تُستخدم على نطاق واسع مع الجدار الناري على الخادم من أجل كشف هجوم حقن SQL وهجوم XSS باستخدام التحليل المعجمي، تحوي على ملفات تضم عدد كبير من العينات الخبيثة لكل من الهجومين [19]، أما العينات الحميدة فتم الحصول عليها من خلال بعض المجموعات النصية المتاحة للاستخدام العام في مجال التعلم الآلي [20].

وبنفس الخطوات التي تم من خلالها تجهيز مجموعة البيانات الأساسية، تم تجهيز المجموعة الجديدة التي بلغ عدد العينات الإجمالي فيها والتي تم اختيارها بشكل عشوائي من المصادر السابقة 52607 عينة، تضم 19790 عينة خبيثة و32817 عينة حميدة، أما عدد كلماتها tokens الإجمالي فقد بلغ 42813 كلمة، أي أنها أكبر حتى من مجموعة البيانات الأساسية.

بعد الانتهاء من تحضير العينات تم اختبار النموذج المقترح المدرب من قبل على هذه المجموعة الجديدة، الجدول (8) يبين نتائج الاختبار، أما الشكل (7) يعرض نفس النتائج بطريقة رسومية وفقاً لمقياس Accuracy.

الجدول (8) نتائج اختبار النموذج المقترح على العينات الجديدة.

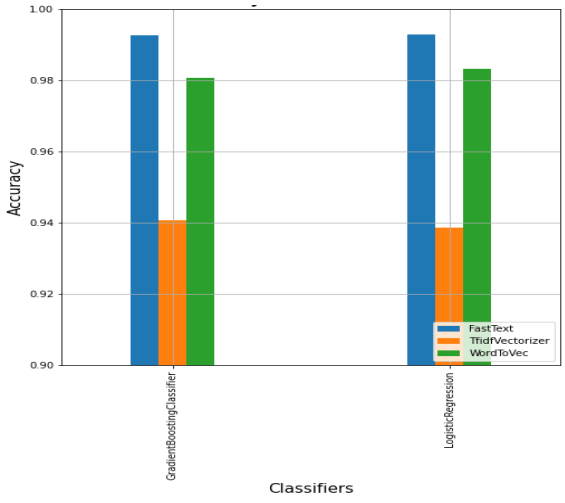
Classifier	Features Extraction Tech	Accuracy
LR	Fasttext	0.9928
GB	Fasttext	0.9926
LR	Word2Vec	0.9832
GB	Word2Vec	0.9806
GB	TF-IDF	0.9407
LR	TF-IDF	0.9384

من خلال استعراض نتائج اختبار النموذج المقترح على العينات الجديدة يمكن ملاحظة الآتي:

أ-تماشياً مع النمط الذي تم رصده في تحليل نتائج مجموعة الاختبار الأصلية في الفقرة السابقة، حافظت طريقة Fasttext على تفوقها واستقرار نتائجها بشكل واضح مع جميع المصنفات، الأمر الذي يجعل منها الطريقة الأنسب للتعميم والتطبيق على أرض الواقع، وذلك على الرغم من أن عدد كلمات OOV ضمن المجموعة الجديدة بلغ 30041 كلمة.

ب-مع ازدياد عدد كلمات OOV انخفض أداء طريقتي Word2Vec وTF_IDF بشكل واضح وخاصة TF-IDF، وهذا أمر متوقع نتيجة أنهما لا تستطيعان تمثيل كلمات OVV، هذا يؤشر إلى أنه كلما استطاع المخترق استخدام كلمات جديدة OVV في الهجوم، فإن فرصه في الاختراق سوف ترتفع على الأرجح في حال كان النموذج معتمداً على هاتين الطريقتين.

ج-بدا أداء خوارزميات التصنيف معتمداً بشكل واضح على الطريقة التي تم استخراج السمات من خلالها، وهذا يؤكد مرة أخرى وتماشياً مع النمط الملاحظ على أهمية مرحلة استخراج السمات.



الشكل (7) دقة النموذج المقترح على العينات الجديدة.

7-الخاتمة والآفاق المستقبلية:

في هذه الورقة تم تطوير نموذج لاكتشاف هجوم حقن SQL بالاعتماد على طريقة Fasttext لاستخراج السمات ومقارنة أدائها مع طريقتي TF-IDF وWord2Vec وذلك باستخدام خوارزميتين تعلم آلي للتصنيف LR وGB.

الهدف الرئيسي من هذا النموذج هو اكتشاف هجوم حقن SQL الذي يشكل وفقاً لإحصائيات مراكز أمن المعلومات والشبكات الدولية واحداً من أخطر الهجمات الإلكترونية، يُستخدم هذا الهجوم على نطاق واسع من قبل المتسللين لاختراق تطبيقات الويب على الإنترنت والوصول غير الشرعي إلى قواعد بياناتها واستخراج المعلومات الحساسة منها وتعديلها أو حذفها وتجاوز المصادقة وأخيراً اختراق الخادم بشكل تام.

أظهرت النتائج التي توصل إليها النموذج المقترح الأداء الأفضل مقارنة بنتائج نماذج الأبحاث السابقة ذات الصلة، بنسبة تصل إلى 99.73٪ بالنسبة لمقياس Accuracy و99.59% بالنسبة لمقياس F1-Score، كما أكدت مرحلة الاختبار الإضافية التي تم تنفيذها على النموذج المقترح قدرته على التعميم والتطبيق العملي.

يمكن أن يركز الجهد المستقبلي لتطوير البحث على مسارين:

أ-اختبار تقنيات تعلم آلي أخرى مستخدمة في تضمين الجمل واكتشاف التشابه فيما بينها، ومقارنة نتائجها مع نتائج هذه الورقة وخاصة مع نتائج طريقة Fasttext، وذلك من أجل الحصول على الأداء الأفضل للتصدي لهذا النوع من الهجمات الإلكترونية.

ب-توسيع نطاق الهجمات ضمن مجموعة البيانات ليشمل هجمات أخرى مثل هجومXSS وغيره من هجمات الحقن بشكل عام.

التمويل: هذا البحث ممول من جامعة دمشق وفق رقم التمويل(501100020595).

METHODS FOR TEXT CLASSIFICATION". Scientific Journal of Impact Factor (SJIF), 5(04).

11) Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). “Efficient estimation of word representations in vector space”. arXiv: 1301.3781.

12) Rong, X. (2016). "word2 vec Parameter Learning Explained". arXiv: 1411.2738v4.

13) Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). "Enriching Word Vectors with Subword Information". arXiv: 1607.04606v2.

14) Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). "Bag of Tricks for Efficient Text Classification". arXiv: 1607.01759v3.

15) Friedman, J. H. (2001). "Greedy function approximation: a gradient boosting machine". Annals of statistics, pages 1189–1232.

16) Friedman, J. H. (2002). "Stochastic gradient boosting". Computational Statistics & Data Analysis 38, 367–378.

17) Zhang, Z., Mayer, G., Dauvilliers, Y., Plazzi, G., Pizza, F., Fronczek, R., & others. (2018). "Exploring the clinical features of narcolepsy type 1 versus narcolepsy type 2 from European Narcolepsy Network database with machine learning". SCIENTIFIC REPORTS, DOI: 10.1038/s41598-018-28840-w.

18) Olson, R., Cavay, W., Mustahsan, Z., Varik, A., & Moorey, J. (2019). "Data-driven advice for applying machine learning to bioinformatics problems". Institute for Biomedical Informatics, University of Pennsylvania, Philadelphia USA.

19) Libinjection, Retrieved November 3, 2022 from https://github.com/client9/libinjection/tree/master/data.

20) Datasets, Retrieved November 3, 2022 from http://mlg.ucd.ie/datasets/bbc.html.

References:

1) OWASP Top Ten Web Application Security Risks 2021, Retrieved November 3, 2022 from https://owasp.org/www-project-top-ten/.

2) ENISA Threat Landscape Report 2020, Retrieved November 3, 2022 from https://www.enisa.europa.eu/publications/enisa-threat-landscape-2020-list-of-top-15-threats.

3) Chen, Zhuang, Guo, Min., &zhou, Lin. (2018). “Research on SQL injection detection technology based on SVM”. MATEC Web of Conferences 173, (2018).

4) Hoang, Dau (2020). "Detecting Common Web Attacks based on Machine Learning Using Web Log". ICERA 2020. Lecture Notes in Networks and Systems, vol 178. Springer, Cham.

5) Chen, Ding, Yan, Qiseng, Wu, Chunwang, &Zhao, Jun. (2020). "SQL Injection Attack Detection and Prevention Techniques Using Deep Learning". Journal of Physics: Conference Series.

6) SQL Injection Dataset, Retrieved November 3, 2022 from https://www.kaggle.com/datasets/syedsaqlainhussain/sql-injection-dataset.Farooq, U. (2021). "Ensemble Machine Learning Approaches for Detection of SQL Injection Attack". TECHNICAL JOURNAL, pp 112-120.

7) SQL Injection Detection, Retrieved November 3, 2022 from https://github.com/shreekanthsenthil/SQL-InjectionDetection/tree/master/Data.

8) SQL Injection Payload List, Retrieved November 3, 2022 from https://github.com/payloadbox/sql-injection-payload-list.

9) Machine Learning Web Application Firewall and Dataset, Retrieved November 3, 2022 from https://github.com/grananqvist/Machine-Learning-Web-Application-Firewall-and-Dataset/tree/master/data.

SQL Injection Overview, Retrieved November 3, 2022 from https://owasp.org/www-community/attacks/SQL_Injection.

10) Waykole, R. N., & Thakare, A. D. (2018). "A REVIEW OF FEATURE EXTRACTION