People's Democratic Republic of Algeria

Ministery of Higher Education and Scientific Research

Ferhat Abbas University of Setif 1



Faculty of Sciences

Computer Science Department

**DISSERTATION**

Presented in fulfillment of the requirements of obtaining the degree

**Master 2 in Computer Science**

Specialty: Data Engineering and Web Technologies

**THEME**

Detecting SQL injections using BERT

*Presented by: Supervised by:*

ATHMANI RAMI Dr. BENZINE MEHDI

BOUHEZILA NASSIM

**2022/2023**

Dedication

*To our parents,*

*To our grandparents,*

*To our brothers and sisters,*

*To our entire family,*

*To all our friends.*

*Athmani Rami,  
Bouhezila Nassim.*

Abstract

Deep learning techniques have improved various domains by using their ability to learn complex patterns from large datasets. In this dissertation, we employed the power of Deep Learning, specifically BERT language model (**B**idirectional **E**ncoder **R**epresentations from **T**ransformers), to resolve the issue of SQL injection attacks on web applications.

The goal of our study is to develop a Deep Learning model using BERT that can accurately identify SQL injections.

Based on the results, our model demonstrated excellent performance; it also indicated that BERT outperforms the compared machine learning models across different evaluation metrics. These results affirm the effectiveness of BERT in detecting SQL injection attacks, underscoring its superior performance in our study.

**Keywords:** Deep learning, Deep learning techniques, Deep Learning model, BERT language model, SQL injection attacks, Web applications, Machine learning models, Evaluation metrics.

Résumé

Les techniques d'apprentissage profond ont amélioré divers domaines en utilisant leur capacité à apprendre des complexes patterns à partir de grands ensembles de données. Dans ce mémoire, nous avons utilisé la puissance de l'apprentissage profond, en particulier le modèle de langage BERT (**B**idirectional **E**ncoder **R**epresentations from **T**ransformers), pour résolu le problème des attaques par injection SQL sur les applications web.

L'objectif de notre étude est de développer un modèle d'apprentissage profond en utilisant BERT qui identifier les injections SQL avec précision.

D'après les résultats, notre modèle a démontré d'excellentes performances ; ils ont également indiqués que BERT a surpassé les autres modèles d'apprentissage automatique comparés à travers différents métriques d'évaluation. Ces résultats confirment l'efficacité de BERT dans la détection des attaques par injection SQL, confirmer sa performance supérieure dans notre étude.

**Mots-clés**: Apprentissage profond, Techniques d'apprentissage profond, Modèle d'apprentissage profond, Modèle de langage BERT, Attaques par injection SQL, Applications web, Modèles d'apprentissage automatique, Métriques d'évaluation.

**ملخص**

حسنت تقنيات التعلم العميق مجالات مختلفة باستخدام قدرتها على تعلم الأنماط المعقدة من مجموعات البيانات الكبيرة. في هذه الأطروحة، استخدمنا قوة التعلم العميق، وتحديداً نموذج اللغة "BERT" (**B**idirectional **E**ncoder **R**epresentations from **T**ransformers) ، لحل مشكلة هجمات حقن SQL على تطبيقات الويب.

الهدف من دراستنا هو تطوير نموذج التعلم العميق باستخدام BERT الذي يمكنه تحديد هجمات حقن SQL بدقة.

بناءً على النتائج المتحصل عليها، أظهر نموذجنا أداءً ممتازًا؛ كما أشار إلى أن BERT يتفوق في الأداء على نماذج التعلم الآلي التي تم المقارنَة بها عبر مقاييس التقييم المختلفة. تؤكد هذه النتائج فعالية BERT في اكتشاف هجمات حقن SQL ، مما يؤكد أدائها المتفوق في دراستنا.

**الكلمات الدالة:** التعلم العميق، تقنيات التعلم العميق، نموذج اللغة BERT، هجمات حقن SQL، تطبيقات الويب، نموذج التعلم العميق، نماذج التعلم الآلي، مقاييس التقييم.

Table of contents

[General introduction 12](#_Toc137853425)

[Chapter 1 SQL injection 14](#_Toc137853425)

[1.1 Introduction 14](#_Toc137853426)

[1.2 Understanding how web applications work 15](#_Toc137853427)

[1.3 How SQL injections work 16](#_Toc137853428)

[1.3.1 Definition 16](#_Toc137853429)

[1.4 Techniques of SQL injections 18](#_Toc137853430)

[1.4.1 Tautologies 18](#_Toc137853431)

[1.4.2 Error-based SQL injection 18](#_Toc137853432)

[1.4.2.1 Mysql database errors 19](#_Toc137853433)

[1.4.3 Blind SQL Injection 23](#_Toc137853434)

[1.4.3.1 Content-based 23](#_Toc137853435)

[1.4.3.2 Time-based 24](#_Toc137853436)

[1.4.4 Union-based SQL injections 25](#_Toc137853437)

[1.5 SQL Injection defense techniques 26](#_Toc137853438)

[1.5.1 Escaping 27](#_Toc137853439)

[1.5.2 Input validation 27](#_Toc137853440)

[1.5.3 Parameterized queries 29](#_Toc137853441)

[1.5.4 Web application firewalls WAF 30](#_Toc137853442)

[1.5.5 Detection using machine learning 30](#_Toc137853443)

[1.6 Conclusion 31](#_Toc137853444)

[Chapter 2 Deep Learning 32](#_Toc137854277)

[2.1 Introduction 32](#_Toc137854278)

[2.2 Machine learning 32](#_Toc137854279)

[2.2.1 Types of machine learning 32](#_Toc137854280)

[2.2.1.1 Supervised learning…………………………………………………..33](#_Toc137854281)

[2.2.1.2 Unsupervised learning………………………………………………..33](#_Toc137854282)

[2.2.1.3 Reinforcement………………………………………………………..33](#_Toc137854283)

[2.2.2 Machine learning algorithms 34](#_Toc137854284)

[2.2.2.1 Linear Regression…………………………………………………….34](#_Toc137854285)

[2.2.2.2 Logistic Regression…………………………………………………..35](#_Toc137854286)

[2.2.2.3 Support vector machines…………………………………………......35](#_Toc137854287)

[2.2.2.4 K-Means………………….…………………………………………..36](#_Toc137854288)

[2.2.3 Machine learning applications 37](#_Toc137854289)

[2.3 Deep learning 37](#_Toc137854290)

[2.3.1 Artificial neural networks 38](#_Toc137854291)

[2.3.2 Activation functions 40](#_Toc137854292)

[2.3.3 Deep learning architectures 42](#_Toc137854293)

[2.3.3.1 Recurrent Neural Networks…………………………………………..43](#_Toc137854294)

[2.3.3.2 Long Short-Term Memory Networks ………………………………..44](#_Toc137854295)

[2.3.3.3 Gated Recurrent Units………………………………………………..45](#_Toc137854296)

[2.3.3.4 Transformers ………………………………………………………..46](#_Toc137854297)

[2.3.4 Deep learning applications 49](#_Toc137854298)

[2.4 Conclusion ………………………………………………………………………..49](#_Toc137854299)

[Chapter 3 Conception and Implementation 50](#_Toc137854445)

[3.1 Introduction 50](#_Toc137854446)

[3.2 General conception of the solution 50](#_Toc137854447)

[3.3 Chosen model: BERT 51](#_Toc137854448)

[3.3.1 BERT architecture 52](#_Toc137854449)

[3.3.2 BERT for Text Classification 53](#_Toc137854450)

[3.3.3 Why BERT was chosen 54](#_Toc137854451)

[3.3.4 Fine-tuning BERT for SQL Injection Detection: 54](#_Toc137854452)

[3.4 Presentation of development tools 54](#_Toc137854453)

[3.4.1 Programming language 54](#_Toc137854454)

[3.4.2 Libraries 55](#_Toc137854455)

[3.4.3 Development environment 56](#_Toc137854456)

[3.5 Dataset 57](#_Toc137854457)

[3.6 Code and Implementation 59](#_Toc137854458)

[3.6.1 Split and Preprocess data for the BERT model 59](#_Toc137854459)

[3.6.2 Build the BERT model 60](#_Toc137854460)

[3.6.3 Fine-tuning the BERT model 60](#_Toc137854461)

[3.6.4 Make predictions with the BERT model 61](#_Toc137854462)

[3.7 Choice of hyperparameters 62](#_Toc137854463)

[3.7.1 Preprocessing hyperparameters 62](#_Toc137854464)

[3.7.2 Data splitting hyperparameters 62](#_Toc137854465)

[3.7.3 Model training hyperparameters 62](#_Toc137854466)

[3.8 Conclusion 63](#_Toc137854467)

[Chapter 4 Test and Evaluation 64](#_Toc137854626)

[4.1 Introduction 64](#_Toc137854627)

[4.2 Confusion matrix 64](#_Toc137854628)

[4.3 Evaluation metrics for assessing model performance 65](#_Toc137854629)

[4.3.1 Accuracy 65](#_Toc137854630)

[4.3.2 Precision 65](#_Toc137854631)

[4.3.3 Recall 66](#_Toc137854632)

[4.3.4 F1 Score 66](#_Toc137854633)

[4.4 Model performance analysis 66](#_Toc137854634)

[4.5 Evaluate the presence of overfitting 68](#_Toc137854635)

[4.6 Comparative analysis with other approaches 69](#_Toc137854636)

[4.7 Model Performance evaluation on new Data 70](#_Toc137854637)

[4.8 Conclusion 71](#_Toc137854638)

[General Conclusion 72](#_Toc137854639)

[References 74](#_Toc137854639)

**List of figures**

[Figure ‎1.1 Three-tier architecture. 15](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-01-SQL-Injection.docx#_Toc137853446)

[Figure 1.‎1.2 Example of a SQL injection attack [3]. 17](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-01-SQL-Injection.docx#_Toc137853447)

[Figure ‎1.3 How Information flows during an SQL injection error [4]. 18](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-01-SQL-Injection.docx#_Toc137853448)

[Figure ‎1.4 SQL injection vulnerability in PHP code. 20](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-01-SQL-Injection.docx#_Toc137853449)

[Figure ‎1.5 handle query error with mysqli library in PHP. 21](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-01-SQL-Injection.docx#_Toc137853450)

[Figure ‎1.6 Character-escaping in PHP code example. 27](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-01-SQL-Injection.docx#_Toc137853451)

[Figure ‎1.7 Input validation of a String example in PHP code. 28](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-01-SQL-Injection.docx#_Toc137853452)

[Figure ‎1.8 Input validation of an integer example in PHP code. 28](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-01-SQL-Injection.docx#_Toc137853453)

[Figure ‎1.9 Parameterized queries using PDO in PHP code example. 29](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-01-SQL-Injection.docx#_Toc137853454)

[Figure ‎2.1 Graphical representation of linear regression. 34](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-02-Deep-Learning.docx#_Toc137854301)

[Figure ‎2.2 Graphical representation of logistic regression. 35](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-02-Deep-Learning.docx#_Toc137854302)

[Figure ‎2.3 Graphical representation of Support vector machines. 36](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-02-Deep-Learning.docx#_Toc137854303)

[Figure ‎2.4 Graphical representation of k means. 36](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-02-Deep-Learning.docx#_Toc137854304)

[Figure ‎2.5 Typical biological-inspired neuron. 38](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-02-Deep-Learning.docx#_Toc137854305)

[Figure ‎2.6 Schematic representation of a neural network. 39](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-02-Deep-Learning.docx#_Toc137854306)

[Figure ‎2.7 Sigmoid activation function. 41](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-02-Deep-Learning.docx#_Toc137854307)

[Figure ‎2.8 ReLU activation function. 41](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-02-Deep-Learning.docx#_Toc137854308)

[Figure ‎2.9 Tanh activation function. 42](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-02-Deep-Learning.docx#_Toc137854309)

[Figure ‎2.10 Diagram of simple recurrent network. 43](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-02-Deep-Learning.docx#_Toc137854310)

[Figure ‎2.11 Long Short-term Memory Neural Network. 44](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-02-Deep-Learning.docx#_Toc137854311)

[Figure ‎2.12 Gated Recurrent Unit. 45](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-02-Deep-Learning.docx#_Toc137854312)

[Figure ‎2.13 Architecture of transformers [19]. 46](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-02-Deep-Learning.docx#_Toc137854313)

[Figure ‎3.1 Sql injection Detection Tool conception and architecture. 51](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-03-conception-implementation.docx#_Toc137854469)

[Figure ‎3.2 BERT model size. 52](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-03-conception-implementation.docx#_Toc137854470)

[Figure ‎3.3 BERT model architecture. 53](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-03-conception-implementation.docx#_Toc137854471)

[Figure ‎3.4 Dataset query classes distribution. 58](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-03-conception-implementation.docx#_Toc137854472)

[Figure ‎3.5 Split data into training and testing sets and preprocess data for BERT model. 59](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-03-conception-implementation.docx#_Toc137854473)

[Figure ‎3.6 Build BERT model Python code. 60](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-03-conception-implementation.docx#_Toc137854474)

[Figure ‎3.7 Fine-tuning BERT model Python code. 60](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-03-conception-implementation.docx#_Toc137854475)

[Figure ‎3.8 Make predictions with the trained model. 61](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-03-conception-implementation.docx#_Toc137854476)

Figure 4.1 Training and validation loss ………………………………………………….68

**List of tables**

[Table ‎1.1 Results of a SELECT query without UNION. 25](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-01-SQL-Injection.docx#_Toc137853446)

[Table ‎1.2 Result of user query after a UNION based SQL injection. 26](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-01-SQL-Injection.docx#_Toc137853446)

[Table ‎4.1 Confusion Matrix. 64](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-01-SQL-Injection.docx#_Toc137853446)

[Table ‎4.2 Confusion Matrix (Classification Results). 67](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-01-SQL-Injection.docx#_Toc137853446)

[Table ‎4.3 Comparing the model performances using various metrics (%).. 69](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-01-SQL-Injection.docx#_Toc137853446)

[Table ‎4.4 Confusion Matrix (Test Classification Results). 70](file:///D:\New-G-Drive\Master-2\Soutenance\detecting-sql-injections-using-bert\Memoire\V3.00\Chapter-01-SQL-Injection.docx#_Toc137853446)

**General Introduction**

The rapid growth of web applications has revolutionized the way we interact and conduct various activities online. From e-commerce platforms and social networks to financial systems and government portals, web applications have become an integral part of our daily lives. However, with greater dependence on online applications comes an increased danger of cyber attacks with SQL injections being one of the most common and dangerous vulnerabilities.

To mitigate the growing threat of SQL injections, traditional approaches such as input validation and query parameterization have been widely adopted. While these methods provide some level of protection, they often struggle to keep pace with the evolving attack techniques employed by adversaries. Thus, there is a pressing need for more advanced and proactive defense mechanisms to detect and prevent SQL injection attacks.

In recent years, deep learning approaches have emerged as a promising solution in various domains, leveraging their ability to automatically learn complex patterns from large datasets. One such powerful deep learning model is BERT (Bidirectional Encoder Representations from Transformers), originally developed for natural language processing tasks. BERT has proven to be highly effective in capturing the semantic and contextual understanding of text, leading to remarkable performance in tasks such as text classification.

In this research, we propose using the power of BERT-based deep learning models to address the critical issue of SQL injection attacks. Our objective is to develop a reliable and efficient detection model capable of accurately identifying SQL injection attempts in real-time. By using BERT's contextual understanding and semantic representation capabilities, we aim to create a model that can effectively distinguish between normal and SQL malicious queries.

Our thesis is organized as follows:

In Chapter 1, we exploreSQL injection attacks, their definitions, types and their detecting techniques, then we head on machine learning and Deep Learning, we present popular algorithmic approaches in machine learning and explore deep learning architectures in Chapter 2. Chapter 3 is dedicated to the general conception of our work and the materials used, including the dataset and the type of deep learning architecture employed. Furthermore, we cover the preprocessing steps taken to ensure the accuracy and efficiency of our system. In Chapter 4, we discuss the test and evaluation of our model for detecting SQL injection attacks. We use a variety of evaluation metrics, including accuracy, precision, recall, and F1 score. We also compare the performance of our model to other machine learning algorithms and related works.