In this project, **we contributed** to the field of web application security by developing and thoroughly evaluating various machine learning and deep learning approaches for SQL injection detection, with a special focus on the BERT model. Throughout our work, **we explored and implemented** both classical models (Logistic Regression, SVM) and deep learning architectures (MLP, RNN, LSTM), carefully analyzing their structure, tuning different hyperparameters, and assessing their performance using multiple evaluation metrics. All models were trained on a labeled dataset and validated on an unseen test set to ensure generalization.

**We invested considerable effort** in experimenting with different configurations, preprocessing techniques, and model parameters to improve detection accuracy. Among all tested models, BERT clearly outperformed the others, achieving **the best results** across all performance indicators. Thanks to its deep contextual and semantic understanding, BERT demonstrated strong capabilities in accurately identifying SQL injection attempts. These findings confirm the relevance of transformer-based models in cybersecurity, especially in detecting language-driven threats.

Despite these promising outcomes, some limitations must be acknowledged. The datasets used, although appropriate for experimentation, may not fully represent the complexity and variety of real-world SQL injection attacks. Moreover, our evaluation was conducted in an offline environment, and the models have yet to be deployed or tested in real-time systems.

Based on our findings and experience, we propose several directions for future work:

* **Enrich and diversify the dataset**, integrating real-world and more complex SQL injection examples;
* **Deploy the model in real-time environments** to better assess its behavior and performance in operational conditions;
* **Continuously update the model** with newly emerging attack patterns to preserve its detection accuracy over time;
* **Expand the detection scope** to include other web-based attacks such as XSS and CSRF, using multi-class classification or ensemble learning techniques.

In conclusion, this study demonstrates the effectiveness of BERT in SQL injection detection and highlights its potential as a powerful tool in modern cybersecurity systems. **Our work lays the groundwork** for future research in building intelligent, adaptive, and scalable threat detection frameworks based on advanced NLP technologies.

(10/10 ) 100 ai

In this project, we contributed to the field of web application security by developing and testing thoroughly numerous machine learning and deep learning techniques for SQL injection with a specific focus on the BERT model. During our project, we experimented and used both classic models (Logistic Regression, SVM) and deep neural architectures (MLP, RNN, LSTM), studying in depth their architecture, hyperparameter optimization of several parameters, and estimation of their performance through a range of evaluation metrics. All models were trained on a labeled set and cross-validated on an unseen test set to test their ability to generalize.

We spent significant time trying out various configurations, preprocessing methods, and model parameters to enhance detection quality. Out of all the models experimented with, BERT surpassed all the others and produced the optimal results on all the performance metrics. Due to its deep contextual and semantic comprehension, BERT showed excellent potential in precisely detecting SQL injection attacks. These results validate the applicability of transformer-based models in cybersecurity, particularly in language-based threat detection.

Despite these promising findings, some of the limitations must be acknowledged. The datasets used, although adequate for experimentation, were not able to reflect the richness and diversity of real SQL injection attacks. In addition, we tested in a simulated environment and have not yet tested or deployed the models on live systems.

Based on our results and experience, the following directions are suggested for future research:

•\tEnrich and diversify the dataset, including real-world and more advanced SQL injection examples;

•\tApply the model in real-time settings to further evaluate its behavior and performance under operational scenarios;

•\tRegularly update the model with recently emerging attack patterns to maintain its detection accuracy over time;

•\tGeneralize the detection scope to other web-based attacks like XSS and CSRF using multi-class classification or ensemble learning methods.

In conclusion, the research confirms the effectiveness of BERT for SQL injection detection and indicates its application as a valuable tool in modern cybersecurity frameworks. This research lays the ground for subsequent research in creating smart, adaptive, and extensible threat detection systems using cutting-edge NLP technologies.

36 %ia (8/10)

In this project, we contributed to web application security by implementing and comparing various machine learning and deep learning approaches to SQL injection detection, with a focus on the BERT model. In our task, we learned and used both classical models (Logistic Regression, SVM) and deep learning models (MLP, RNN, LSTM), analyzing their architecture, tuning multiple hyperparameters, and comparing their performance on multiple evaluation metrics. All models were trained on a labeled dataset and evaluated on an unseen test set for generalization.

We invested much effort in experimenting with different configurations, preprocessing techniques, and model hyperparameters to improve detection accuracy. Among all models experimented with, BERT consistently outperformed the others with the best scores for all performance metrics. Due to its deep contextual and semantic understanding, BERT achieved high competence at accurately detecting SQL injection attempts. The findings confirm the feasibility of transformer-based models for cybersecurity, and specifically for language-based threat detection.

Despite these positive outcomes, several limitations must be mentioned. Datasets used, although appropriate for experimentation, may not ideally represent the diversity and complexity of real-world SQL injection attacks. Also, our tests were conducted in an offline environment, and the models have yet to be deployed or tested on real-time systems.

Based on our findings and experience, we propose several directions for future work:

• Improve and enrich the dataset with real-world and more complex SQL injection examples;

• Implement the model in real environments to continue testing its behavior and performance in operational conditions;

• Periodically update the model with new arising attack patterns to preserve its detection accuracy over time;

• Enlarge the detection to other web-oriented attacks such as XSS and CSRF, using multi-class classification or ensemble learning techniques.

In conclusion, this study demonstrates the feasibility of BERT for SQL injection detection and its potential as a valuable component of modern cybersecurity systems. Our study paves the way for follow-up research on the development of intelligent, adaptive, and scalable threat detection systems with advanced NLP technologies

40 %ia (9/10)