**Chapter 3**

**Conception and Implementation**

**3.1Introduction**

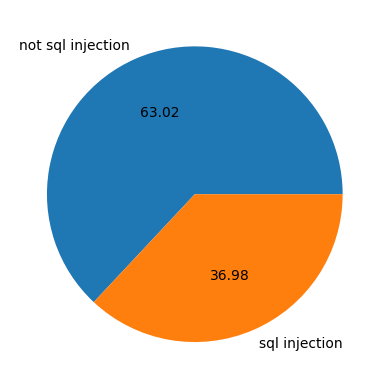
The chapter describes the models' design and implementation for SQL injection detection. Different architectures evaluated in our work include the traditional machine learning, deep learning, and transformer-based approaches. The rationale behind this approach is to assess and compare the models' performances on the same dataset in order to find the one most suitable for accurately detecting SQL injection attempts. The next sections describe the dataset employed, as well as the preprocessing done on it, followed by the design for each of the models and the evaluation strategy considered for training and testing.

**3.2. Dataset**

**3.2.2. Dataset Description**

This study uses two separate datasets for training and testing purposes:

* **Training Dataset (SQLIV3\_cleaned2.csv)**:
  + Contains **30,614 SQL queries**.
  + Each entry is composed of a Sentence (the SQL query) and a Label (0 for benign, 1 for SQL Injection).
  + Label distribution:
    - **Normal (Label = 0)**: 19,268 queries
    - **Malicious (Label = 1)**: 11,346 queries



* **Testing Dataset (sqliv2\_utf8.csv)**:
  + Contains **33,760 SQL queries**.
  + Structured in the same format with Sentence and Label columns.
  + Label distribution:
    - **Normal (Label = 0)**: 22,305 queries
    - **Malicious (Label = 1)**: 11,455 queries

**(SQLIV3\_cleaned2.csv)**

**3.2.3. Data Preprocessing**

Before training, both datasets were cleaned by removing duplicate queries. These queries were then vectorized using appropriate text processing techniques such as **TF-IDF** or **token embeddings**, depending on the model.

**3.3 Models Implemented**

This section on models presents the various models that have been developed for detecting SQL injections. A number of machine learning models and deep learning models were trained and evaluated with varying architecture and hyperparameters. The aim is to see how traditional techniques and modern, state-of-the-art techniques differ in input to achieve results on the same dataset.

For each model, we describe the structure, the main hyperparameters used during training, and the results obtained. These models are as follows:

Support Vector Machine (SVM)

Logistic Regression (LR)

Multilayer Perceptron (MLP)

Simple Neural Network (SN)

Recurrent Neural Network (RNN)

Long Short-Term Memory (LSTM)

BERT Transformer Model (dedicated in Chapter 4)

**3.3.1 Support Vector Machine (SVM)**

Support Vector Machine (SVM) was chosen as one of the traditional machine learning methods to detect SQL injection attacks.

The model was trained using TF-IDF feature vectors generated from the preprocessed SQL queries.

**Hyperparameters used:**

* **Kernel:** Linear
* **Regularization parameter (C):** 1.0
* **Probability estimation:** Enabled
* **Random state:** 42

**Training Results:**

* **Test Accuracy:** 98.78%

The SVM model achieved a test accuracy of **98.78%**, showing strong performance in detecting SQL injection attempts.  
The use of a linear kernel proved effective, suggesting that the TF-IDF-transformed feature space was approximately linearly separable.