**4.1 Introduction**

In this chapter, we present our detection of SQL injection attacks using the BERT language model. BERT is distinct from machine learning and deep learning approaches since it is able to both understand context and syntactic patterns of input queries. Therefore, it is more capable of detecting fine-grained differences and underlying malicious patterns which are often embedded in SQL statements. We begin by laying down BERT's architecture and the reasons behind choosing it for this task. We then move on to explaining the preprocessing that must be done in order to make our dataset compatible with BERT's input requirements, and then move on to explaining the fine-tuning process employed in order to train the model for binary classification.

**4.2 Why BERT for SQL Injection Detection**

Machine learning approaches that existed before deep learning systems along with traditional deep learning models frequently encounter difficulties in detecting SQL injection attacks. The models depend on human-engineered features and produce superficial data representations which prevent them from grasping the full meaning of SQL statements. These models find it difficult to recognize intricate malicious patterns which developers purposely hide inside SQL queries.

BERT utilizes its bidirectional transformer architecture to solve these challenges by processing token context from both directions. The system reaches a superior level of semantic analysis which enables BERT to understand the complete syntax and context of SQL queries. BERT demonstrates a substantial enhancement in detection performance for complex SQL attack patterns which earlier models were unable to detect.

**4.3 BERT Architecture Overview**

BERT model architecture is a multilayer bidirectional transformer encoder closely mimicking the base Transformer model. The Transformer architecture consists of an encoder-decoder network with self-attention on the encoder and attention mechanisms on the decoder.

BERTBASE has 12 encoder stack layers and BERTLARGE has 24 encoder layers. Both models outclass the original Transformer model described in the seminal paper, which had only 6 encoder layers. The base Transformer model includes 512 hidden units and 8 attention heads.

Compared to the base Transformer, the BERTBASE and BERTLARGE architectures have greater feedforward networks with 768 and 1024 hidden units, respectively, as well as more attention heads, 12 and 16, respectively.

The BERTBASE architecture comprises approximately 110 million parameters, whereas the BERTLARGE architecture contains approximately 340 million parameters.

The input to the model begins with a classification token, [CLS], followed by a sequence of words. The [CLS] token is exclusively utilized for classification. The input is then fed into the sequence of encoder layers, wherein each is executing self-attention mechanisms followed by feedforward neural networks, and then to the subsequent encoder layer.

The output of the model is a 768-dimensional hidden vector for BERTBASE. In classification, the [CLS] token's output is typically used as the representation of the input sequence and passed through a classifier layer. helps generate more detailed language representations which specifically benefit tasks dealing with syntax-sensitive input such as SQL queries.

**4.4 Data Preprocessing for BERT**

**4.5 Fine-tuning BERT for SQL Injection Detection**

**4.6 Implementation Details**

**4.7 Evaluation Metrics and Results**