Software Supply Chain Vulnerabilities Detection in Source Code: Performance Comparison between Traditional and Quantum Machine Learning Algorithms

Mst Shapna Akter

Software Supply Chain Vulnerabilities Detection in Source Code: Performance Comparison between Traditional and Quantum Machine Learning Algorithms

Mst Shapna Akter*, Nafisa Anjum†, Hossain Shahriar†

*Department of Computer Science, Kennesaw State University, USA
†Department of Information Technology, Kennesaw State University, USA
{makter2, nanjum}@students.kennesaw.edu | {hshahria}@kennesaw.edu

Abstract: The software supply chain (SSC) attack has become one of the crucial issues that are being increased rapidly with the advancement of the software development domain. In general, SSC attacks execute during the software development processes lead to vulnerabilities in software products targeting downstream customers and even involved stakeholders. Machine Learning approaches are proven in detecting and preventing software security vulnerabilities. Besides, emerging quantum machine learning can be promising in addressing SSC attacks. Considering the distinction between traditional and quantum machine learning, performance could be varies based on the proportions of the experimenting dataset. In this paper, we conduct a comparative analysis between quantum neural networks (QNN) and conventional neural networks (NN) with a software supply chain attack dataset known as ClaMP. Our goal is to distinguish the performance between QNN and NN and to conduct the experiment, we develop two different models for QNN and NN by utilizing Pennylane for quantum and TensorFlow and Keras for traditional respectively. We evaluated the performance of both models with different proportions of the ClaMP dataset to identify the f1 score, recall, precision, and accuracy. We also measure the execution time to check the efficiency of both models. The demonstration result indicates that execution time for QNN is slower than NN with a higher percentage of datasets. Due to recent advancements in QNN, a large level of experiments shall be carried out to understand both models accurately in our future research.

Keywords— Software supply chain Security, Quantum machine learning, Quantum neural network (QNN), Neural Network (NN), ClaMP, TensorFlow, Pennylane

References:

[1] M. Mohammad *et al.*, "Quantum Machine Learning for Software Supply Chain Attacks: How Far Can We Go?," 2022, [Online]. Available: https://arxiv.org/abs/2204.02784.

Abstract— The software supply chain (SSC) attack has become one of the crucial issues that are being increased rapidly with the advancement of the software development domain. In general, SSC attacks execute during the software development processes lead to vulnerabilities in software products targeting downstream customers and even involved stakeholders. Machine Learning approaches are proven in detecting and preventing software security vulnerabilities. Besides, emerging quantum machine learning can be promising in addressing SSC attacks. Considering the distinction between traditional and quantum machine learning, performance could be varies based on the proportions of the experimenting dataset. In this paper, we conduct a comparative analysis between quantum neural networks (QNN) and conventional neural networks (NN) with a software supply chain attack dataset known as ClaMP. Our goal is to distinguish the performance between QNN and NN and to conduct the experiment, we develop two different models for QNN and NN by utilizing Pennylane for quantum and TensorFlow and Keras for traditional respectively. We evaluated the performance of both models with different proportions of the ClaMP dataset to identify the f1 score, recall, precision, and accuracy. We also measure the execution time to check the efficiency of both models. The demonstration result indicates that execution time for QNN is slower than NN with a higher percentage of datasets. Due to recent advancements in QNN, a large level of experiments shall be carried out to understand both models accurately in our future research.