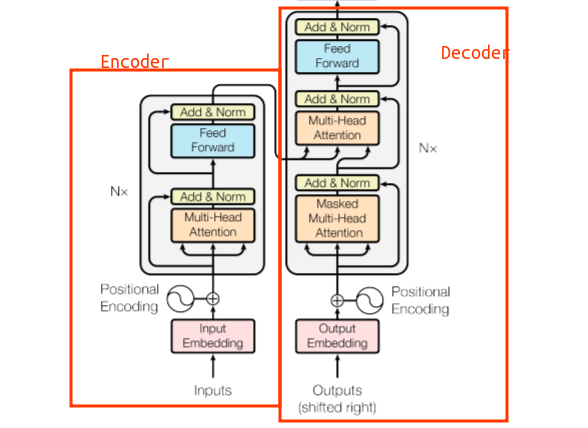
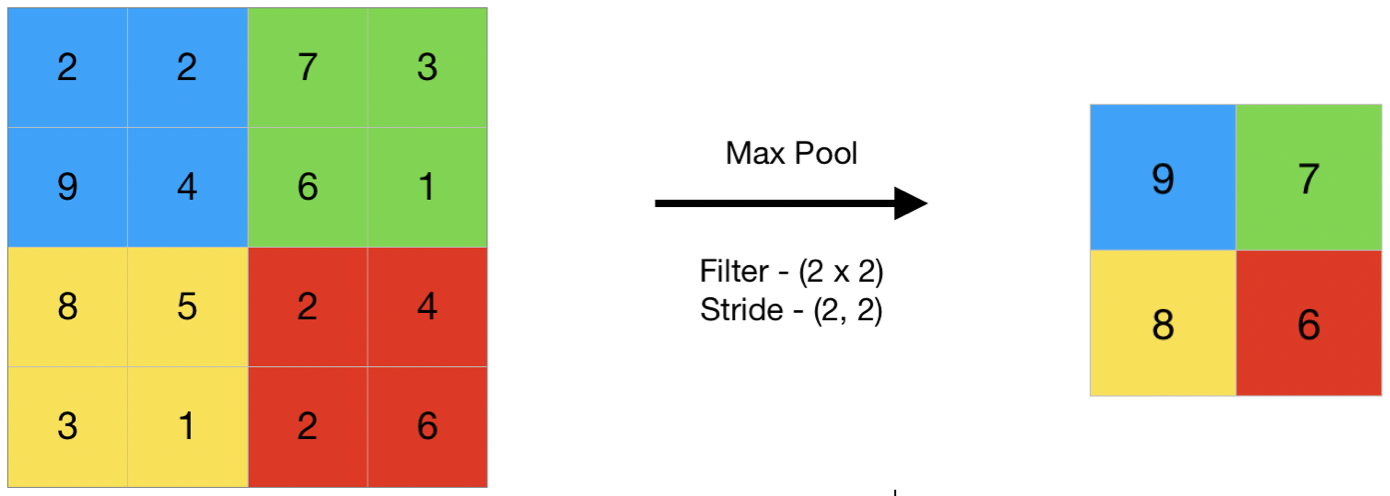
Proposal of EIE4117(Capstone Project)- DGA detection (23016082d Ye Lim Kit Michael)

1. Objectives:  
   The objective of this project is to detect the domain generated by the DGA algorithm through the deep learning method and compare the performance with machine learning.
2. Introduction:  
   DGA domain is used in computer viruses when a computer gets infected, the attacker may want to have the personal credential of the victim or sometimes get control of their devices. To achieve this, sometimes a virus will install a DGA algorithm to the infected device and keep sending out DNS requests with a computer-generated domain name. And within these DNS requests, only a few of them can be connected to the attacker. Then they can get the credentials of the user or command their computer to perform what they want.  
   So in this project, I will try to develop a machine learning model to classify whether the domain is generated by the DGA algorithm. We will try a deep learning model first, which is a mixed layer model with N-gram, transformer (encoder only) and a pooling layer to process the encoded vector from the transformer. After we build a basic model for DGA detection. There are several aspects we would like to further investigate:  
   1. Performance between machine learning and our deep learning model.  
   2. The resistance to the adversarial attack.  
   3. Optimize the model.
3. Methodology:  
   Our model consists of different layers and parts. I will combine them in one and train the model with all the layers together.  
   First, we will do some basic cleaning on the domain name, as we are only interested in the second-level domain of the site. Some cleaning for the domain will need to be performed first. To achieve this, I will use a library “tldextract” to clear the domain first with only the second-level domain being left. For example, a domain name “www.fb01cc736a91a324bedd71db0a8e2a3c.info” will be converted to “fb01cc736a91a324bedd71db0a8e2a3c” as we are only interested in the second-level domain. The “www.” and “.info” will be removed.   
   Furthermore, after the domain name is clean, we will use an N-gram model to separate a word into a small piece to input it into the model in the back. For example, if we are using 1-gram, then the word “google”, will be separated into multiple “grams” and each gram with N words. In this case, the word “google” will be converted to “g”, “o”, “o”, “g”, “l”, “e”. And it will be inputted to the model.   
   After the word is converted into N-gram. It will be inputted into an encoder of a transformer first. Which contains a word embedding layer, a multi-head attention mechanism, a normalization layer and a feed word layer.  
   The workflow of an encoder will be: first the word will get input to the word embedding layer which converts the word into a number for the model to analyse, then it will use the positional encoding algorithm to give each text a positional index which will be added to the embedded word. After that, the model will use the multi-head attention to calculate the Query, Key and Value vector for each word. Which can analyse the relationship between each word of a sentence. Then it will output a vector of attention value. Each represents different parts of the data. Finally, it will go through a Feed Forward network, which is a simple neural network, which connects the vector as an input with some neurons inside. It can weigh the output vector of attention and decide which is more related to the task. And it will produce a vector as the output of the encoder.  
     
     
   fig1. For a transformer model, we will only focus on the encoder in this project. Please ignore the decoder part.[1]  
   After the encoder, it will get to the pooling layer, which is a simple layer that calculates the max/average of an area of a matrix. Depending on our choice, it may either be calculating the average of the area or the maximum number of that area. Then it will get through a fully connected layer again and end with a neuron with a sigmoid as an activation function to decide whether it is a malicious domain.  
     
   fig2. A demonstration of the pooling layer. [2]  
   And the following is the structure of my model: 一張含有 文字, 螢幕擷取畫面 的圖片

   自動產生的描述
4. Project schedule:  
   build the basic model- before the 1st of Oct 2024  
   refine the model- before the 1st of November 2024  
   develop several traditional machine learning model for compression- before the 1st of December 2024  
   test other classification method and compare the performance(traditional machine learning)- before 1st of January 2025  
   try to make the model able to resist the adversarial attack- before the 1st of February 2025
5. Reference:  
   1. Encoder-Only vs Decoder-Only vs Encoder-Decoder Transformer. Vaclav Kosar(29 Oct 2023): <https://vaclavkosar.com/ml/Encoder-only-Decoder-only-vs-Encoder-Decoder-Transfomer>  
   2. CNN | Introduction to Pooling Layer. savyakhosla (04 Sep 2024): <https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>