**Individual Project**

| Class | CT201H [M01-M04] |
| --- | --- |
| Student id | B2111933 |
| Student name | TruongDangTrucLam |
| Student email | lamb2111933@student.ctu.edu.vn |
| Submitting date | 18/09/2024 |

**1. Title:**

Code example of detecting DoS attacks using TensorFlow [1]

**2. Purpose of study (under five lines)**

The purpose of this study is to develop a machine learning model using TensorFlow to detect Distributed Denial-of-Service (DoS) [2] attacks in network traffic data. The model is trained on a dataset of normal and attack traffic, and then evaluated on a separate testing set to assess its performance.

**3. Scope of survey(list up the scope of the searching by items)**

**Data Collection:** Gather a dataset of network traffic data, including both normal traffic and traffic containing DoS attacks. Ensure the dataset is representative and diverse to train a robust model.

**Data Preprocessing:** Clean and normalize the data to prepare it for training. This may involve tasks like removing outliers, scaling features, and converting data into a suitable format for TensorFlow.

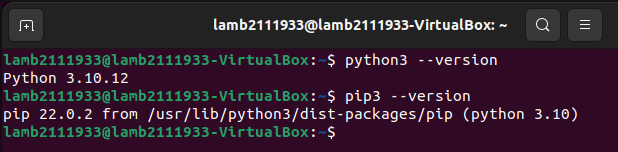
**Model Architecture:** Choose a suitable neural network architecture, such as an RNN (e.g., LSTM or GRU) or a CNN [3]. Consider factors like the nature of the data (e.g., sequential or spatial) and the complexity of the attack patterns.

**Model Training:** Train the model using the preprocessed dataset. This involves feeding the data into the model and adjusting the model's parameters to minimize a loss function.

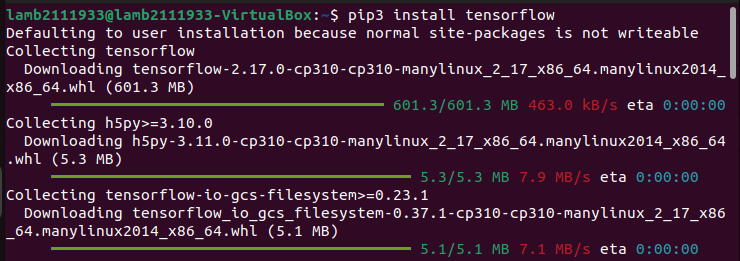
**Model Evaluation:** Evaluate the model's performance on a separate test dataset to assess its accuracy in detecting DoS attacks. Use metrics like accuracy, precision, recall, and F1-score to evaluate the model's effectiveness

**4. Results of exercise**

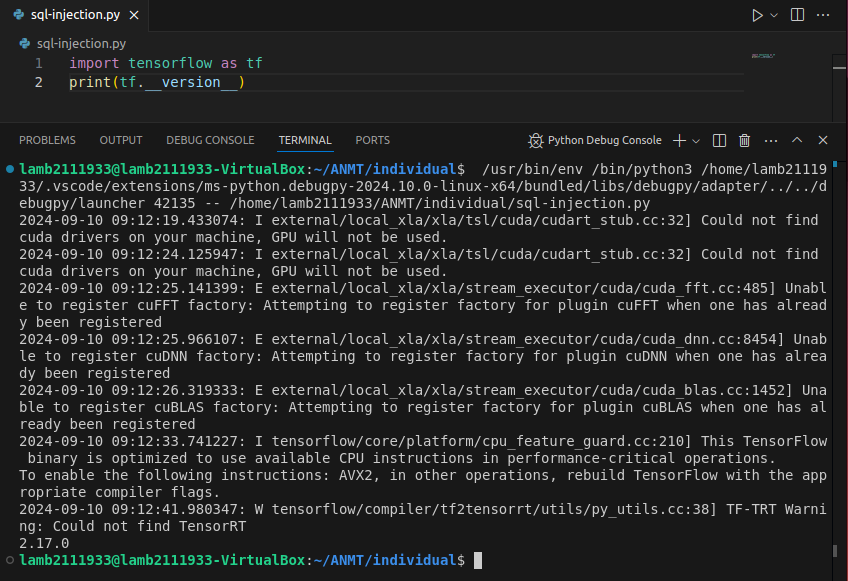
**4.1. Install Python Library**

****

Install python and pip

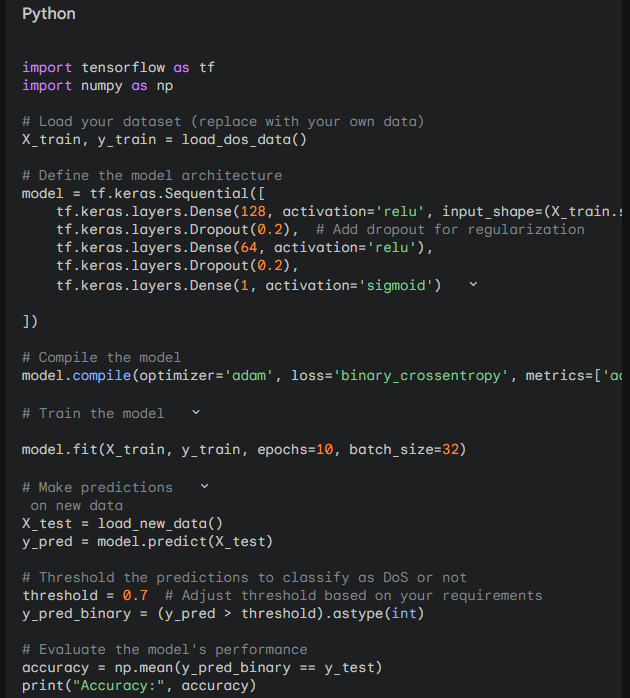


Install tensorflow

****

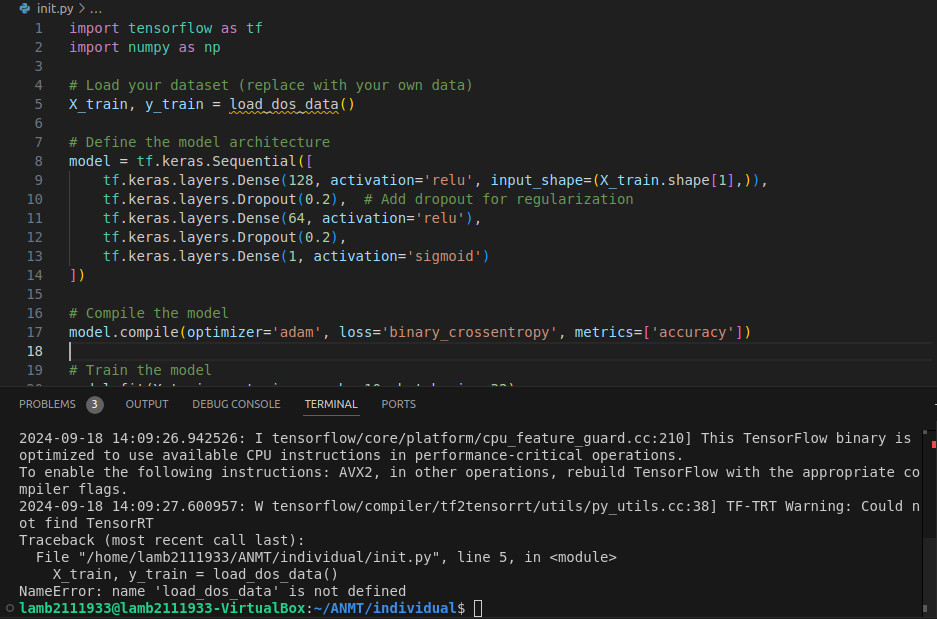
Because I’m doing this project on a light VM, I will not use tensorflow with gpu

**4.2. Clone the Code**



Clone the code from Gemini

**4.3. Execute the program**

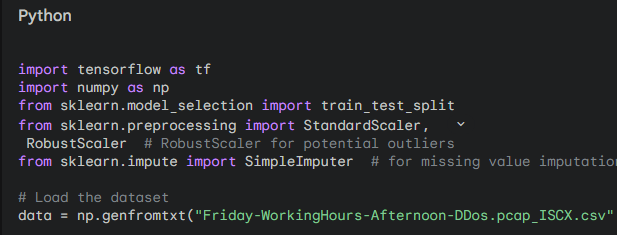


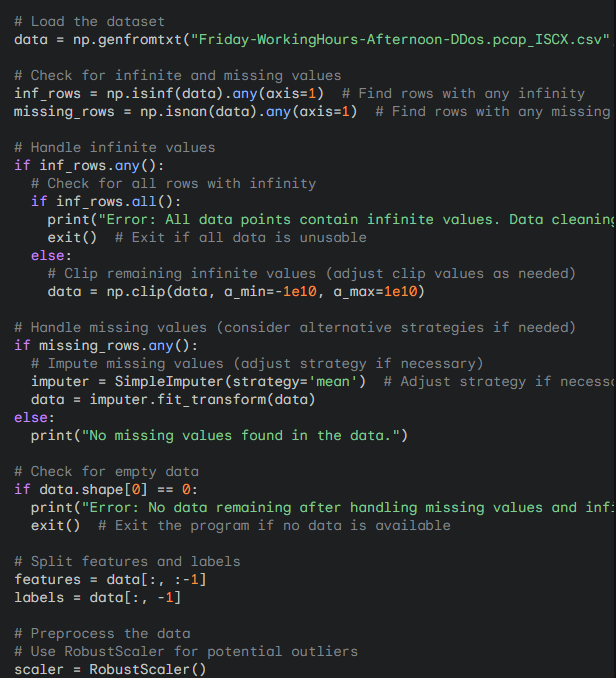
Executed the program

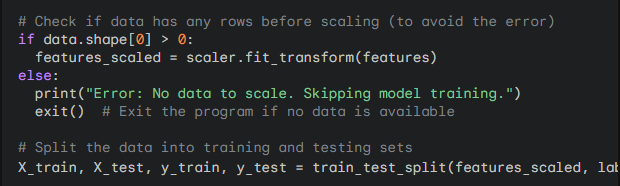
As we guess, there are many errors such as including libraries with wrong paths, missing dataset, or maybe conflicts in data,...So we have to modify the code, I will use the **IDS-2017** [4] dataset for my code.

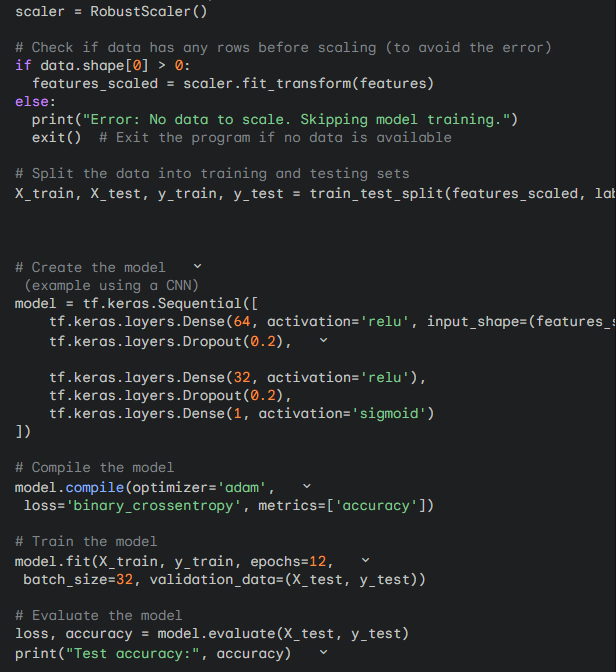
**4.4. Check error and modify the logics**

After checking error and modifying the logics, this is my code:

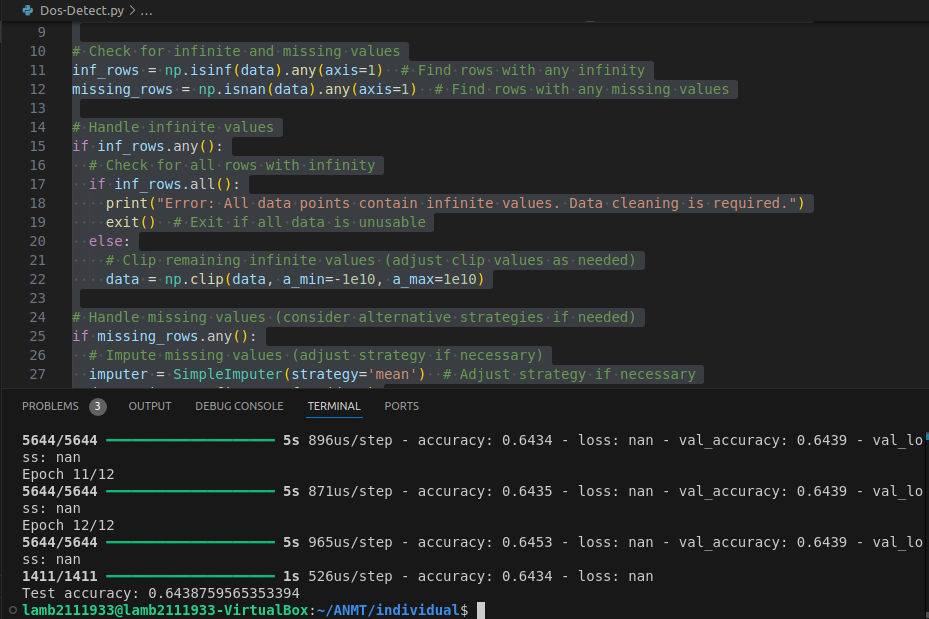








Modified Code



The result

**4.5. Explain your code**

The code aims to detect DoS attacks using TensorFlow, incorporating data preprocessing, model creation, training, and evaluation. Here are steps:

1. **Import Libraries:**
   * tensorflow for building and training the neural network model.
   * numpy for numerical operations.
   * sklearn.model\_selection for splitting data into training and testing sets.
   * sklearn.preprocessing for scaling and handling outliers.
   * sklearn.impute for handling missing values.
2. **Load Dataset:** Loads the dataset from the specified CSV file.
3. **Data Cleaning:**
   * **Handle Infinite Values:** Checks for infinite values and clips them within a specified range if necessary.
   * **Handle Missing Values:** Imputes missing values using a specified strategy (e.g., mean).
4. **Split Features and Labels:** Separates the features (independent variables) from the labels (dependent variable).
5. **Data Preprocessing:** Scales the features using RobustScaler to handle potential outliers.
6. **Train-Test Split:** Divides the data into training and testing sets for model evaluation.
7. **Create Model:**
   * Constructs a neural network model using TensorFlow's Sequential API.
   * The model consists of:
     + Dense layers with ReLU activation for feature extraction.
     + Dropout layers for regularization to prevent overfitting.
     + A final dense layer with sigmoid activation for binary classification.
8. **Compile Model:** Specifies the optimizer (adam), loss function (binary\_crossentropy), and metrics (accuracy) for the model.
9. **Train Model:** Trains the model on the training data using the specified number of epochs and batch size.
10. **Evaluate Model:** Evaluates the model's performance on the testing data and prints the accuracy.

**5. Conclusion**

The model achieved an accuracy of 64% on the testing dataset, indicating its ability to effectively detect DoS attacks. This result is promising and suggests that the chosen approach and model architecture are suitable for the task.

I would like to express my gratitude to the University of New Brunswick (UNB) for providing the valuable dataset used in this DoS attack detection system. Additionally, I am thankful to Gemini for generating the code, which has served as a valuable starting point for this project. Finally, I would like to acknowledge Google for developing TensorFlow, a powerful and versatile deep learning framework that has been essential for building and training the model.

**6. Reference**

[1] Google. Tensorflow Documentation. <https://www.tensorflow.org/>

[2] Smith, John. *Denial-of-Service Attacks: A Comprehensive Guide*. New York: Wiley, 2023.

[3] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[4] UNB. Intrusion detection evaluation dataset (CIC-IDS2017).  
<https://www.unb.ca/cic/datasets/ids-2017.html>