CSC 215-01 Artificial Intelligence (Spring 2019)

Mini-Project 3: Network Intrusion Detection

Due at 4 pm, Monday, March 11, 2019

Demo Session: class time, Monday, March 11, 2019

1. Problem Formulation

Software to detect network intrusions protects a computer network from unauthorized users, including perhaps insiders. This project aims to build a network intrusion detector, a predictive model capable of distinguishing between bad connections, called intrusions or attacks, and good normal connections.

Model this problem as a BINARY classification problem. Use the following models to detect bad connections (intrusions). Compare the accuracy, recall, precision and F1-score of ALL the models. PRINT and PLOT the confusion matrix for each model. Show the ROC curve of each model.

- Logistic Regression
- Nearest Neighbor
- Support Vector Machine
- Gaussian Naive Bayes
- Fully-Connected Neural Networks
- Convolutional Neural Networks (CNN)

In all the models, encode good connections as "0" and bad connections as "1". To achieve this, you may want to apply some operations/functions to the label column. Check this out if you need hints:

https://chrisalbon.com/python/data_wrangling/pandas_apply_operations_to_dataframes/

Hint: For CNN, find a way to view each sample data as an image. Please refer to our lab tutorial on using CNN to handle data other than images. You may use either Conv2D or Conv1D.

2. Dataset

Download link:
https://drive.google.com/open?id=1J3Fgo1DCwuQpljJo8Yspe_PQad7oJb52

This database contains a wide variety of intrusions simulated in a military network environment.

3. Requirements

- Split data for training and testing. Use training data to train your models and evaluate the model quality using test data
- Drop any rows with missing values.
- Drop all the redundant records. This data set has **a big number of redundant records**. Redundant records in the train set will cause learning algorithms to be biased towards the more frequent records.
- Encode categorical features and normalize numeric features.
- You must use EarlyStopping and ModelCheckpoint when training neural networks using Tensorflow.
- Tuning the following hyperparameters when training neural networks using Tensorflow to see how they affect performance
    - **Activation:** relu, sigmoid, tanh
    - **Layers and neuron counts**
    - **Optimizer:** adam and sgd
    - **Kernel number and kernel size** (for CNN only)

4. Grading breakdown

You may feel this project is described with some certain degree of vagueness, which is left on purpose. In other words, **creativity is strongly encouraged**. Your grade for this project will be based on the soundness of your design, the novelty of your work, and the effort you put into the project.

Use the evaluation form on Canvas as a checklist to make sure your work meet all the requirements.

Implementation	**70 pts**
Your report	**15 pts**
In-class defense	**10 pts**
Additional features (novelty)	**5 pts**

5. Teaming

Students must work in teams of 2 people. Think clearly about who will do what on the project. Normally people in the same group will receive the same grade. However, the instructor reserve the right to assign different grades to team members depending on their contributions. So you should choose partner carefully!

6. Deliverables

(1) All your source code in Python Jupyter notebook.

(2) Your report in PDF format, with your name, your id, course title, assignment id, and due date on the first page. As for length, I would expect a report with more than one page. Your report should include the following sections (but not limited to):

- **Problem Statement**
- **Methodology**
- **Experimental Results and Analysis**
- **Task Division and Project Reflection**
- **Additional Features.**

In the section "**Task Division and Project Reflection**", describe the following:

- who is responsible for which part,
- challenges your group encountered and how you solved them
- and what you have learned from the project as a team.

10 pts will be deducted for missing the section of task division and project reflection.

All the files must be submitted **by team leader** on Canvas before

4 pm, Monday, March 11, 2019

NO late submissions will be accepted.

7. In-class Demo:

Each team member must demo your work during the scheduled demo session. Each team should create **a video of 4-5 minutes** to demo your work in class. One free tool to create video is here:

https://screencast-o-matic.com/screen-recorder

The following is how you should allocate your time in your video:

- Model design (1 minute)
- Findings/results (1 minute)
- Additional features (1 minute)
- Task division (1 minute)
- Challenges encountered and what you have learned from the project (1 minutes)

If you implement **additional features (novelty)**, please do mention them to receive credit for novelty.

Failure to show up in defense session will result in **zero** point for the project.

8. Additional Features

Some possible ways to explore for additional features:

(1) Can we do feature selection based on **feature importance analysis** to improve the model quality? Can you use a plot to visualize the importance of each feature?

(2) Can you model this intrusion detection problem as **a multi-class classification problem** so that we can detect the type of each intrusion? How good the predictive model can be in this case?

(3) A grand challenge:

https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/

9. Hints

- The CSV file has no column headers. So you may want to add them using the following code after you load data into dataframe using pd.read_csv():

```
df.columns = [
    'duration',
    'protocol_type',
    'service',
    'flag',
```

```
    'src_bytes',
    'dst_bytes',
    'land',
    'wrong_fragment',
    'urgent',
    'hot',
    'num_failed_logins',
    'logged_in',
    'num_compromised',
    'root_shell',
    'su_attempted',
    'num_root',
    'num_file_creations',
    'num_shells',
    'num_access_files',
    'num_outbound_cmds',
    'is_host_login',
    'is_guest_login',
    'count',
    'srv_count',
    'serror_rate',
    'srv_serror_rate',
    'rerror_rate',
    'srv_rerror_rate',
    'same_srv_rate',
    'diff_srv_rate',
    'srv_diff_host_rate',
    'dst_host_count',
    'dst_host_srv_count',
    'dst_host_same_srv_rate',
    'dst_host_diff_srv_rate',
    'dst_host_same_src_port_rate',
    'dst_host_srv_diff_host_rate',
    'dst_host_serror_rate',
    'dst_host_srv_serror_rate',
    'dst_host_rerror_rate',
    'dst_host_srv_rerror_rate',
    'outcome'
]
```