IoT-Network-Intrusion-Detection-System-UNSW-NB15

Network Intrusion Detection based on various machine learning and deep learning algorithms using UNSW-NB15 Dataset

Prerequisites

- Sklearn
- Pandas
- Numpy
- Matplotlib
- Pickle

Running the Notebook

The notebook can be run on

- Google Colaboratory
- Jupyter Notebook

Instructions

- To run the code, user must have the required Dataset on their system or programming environment.
- Upload the notebook and dataset on Jupyter Notebook or Google Colaboratory.
- Click on the file with .ipynb extension to open the notebook. To run complete code at once press Ctrl + F9
- To run any specific segment of code, select that code cell and press Shift+Enter or Ctrl+Shift+Enter

> **Caution - The code should be executed in the given order for best results without encountering any errors.**

Datasets

- **UNSW_NB15.csv** - Original Dataset
- **UNSW_NB15_features.csv** - 49 features with the class label. These features are described in UNSW-NB15_freatures.csv file.

- **bin_data.csv** - CSV Dataset file for Binary Classification
- **multi_data.csv** - CSV Dataset file for Multi-class Classification

Machine Learning Models

- Decision Tree Classifier
- K-Nearest-Neighbor Classifier
- Linear Regression Model
- Linear Support Vector Machine
- Logistic Regression Model
- Multi Layer Perceptron Classifier
- Random Forest Classifier

Data Preprocessing

- Dataset had **45 attributes** and **175341 rows**.
- After dropping null values Dataset had **45 attributes** and **81173 rows**.
- Data type of attributes is converted using provided datatype information from features dataset.

One-hot-encoding

- Categorical Columns **'proto'**, **'service'**, **'state'** are one-hot-encoded using **pd.get_dummies()** and these 3 attributes are removed afterwards.
- **data_cat** Dataframe had **19 attributes** after one-hot-encoding.
- **data_cat** is concatenated with the main **data** dataframe.
- Total attributes of **data** dataframe - **61**

Data Normalization

- **58 Numeric Columns** of DataFrame is scaled using **MinMax Scaler**.

Binary Classification

- A copy of DataFrame is created for Binary Classification.
- **'label'** attribute is classified into two categories **'normal'** and **'abnormal'**.
- **'label'** is encoded using **LabelEncoder()**, encoded labels are saved in **'label'**.
- Binary dataset - **81173 rows**, **61 columns**

Multi-class Classification

- A copy of DataFrame is created for Multi-class Classification.

- **'attack_cat'** attribute is classified into **9** categories **'Analysis'**, **'Backdoor'**, **'DoS'**, **'Exploits'**, **'Fuzzers'**, **'Generic'**, **'Normal'**, **'Reconnaissance'**, **'Worms'**
- **attack_cat** is encoded using **LabelEncoder()**, encoded labels are saved in **label**.
- **attack_cat is one-hot-encoded'**.
- Multi-class Dataset - **81173 rows**, **69 columns**

• Feature Extraction

- No. of attributes of **'bin_data'** - **61**
- No. of attributes of **'multi_data'** - **69**
- **Pearson Correlation Coefficient** method is used for feature extraction.
- The attributes with **more than 0.3** correlation coefficient with the target attribute **label** were selected.
- No. of attributes of **'bin_data'** after feature selection - **15**
- **'rate'**, **'sttl'**, **'sload'**, **'dload'**, **'ct_srv_src'**, **'ct_state_ttl'**, **'ct_dst_ltm'**, **'ct_src_dport_ltm'**, **'ct_dst_sport_ltm'**, **'ct_dst_src_ltm'**, **'ct_src_ltm'**, **'ct_srv_dst'**, **'state_CON'**, **'state_INT'**, **'label'**
- No. of attributes of **'multi_data'** after feature selection - **16**
- **'dttl'**, **'swin'**, **'dwin'**, **'tcprtt'**, **'synack'**, **'ackdat'**, **'label'**, **'proto_tcp'**, **'proto_udp'**, **'service_dns'**, **'state_CON'**, **'state_FIN'**, **'attack_cat_Analysis'**, **'attack_cat_DoS'**, **'attack_cat_Exploits'**, **'attack_cat_Normal'**

Splitting Dataset

- Randomly Splitting the **bin_data** in **80% for training** and **20% for testing**
- Randomly Splitting the **multi_data** in **70% for training** and **30% for testing**

Decision Tree Classifier

• Binary Classification

- Accuracy - **98.09054511857099**
- Mean Absolute Error - **0.019094548814290114**
- Mean Squared Error - **0.019094548814290114**
- Root Mean Squared Error - **0.13818302650575473**
- R2 Score - **89.55757103838098**
- `DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini', max_depth=None, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1,`

```
min_samples_split=2, min_weight_fraction_leaf=0.0, presort='deprecated',
random_state=123, splitter='best')
```

- **Multi-class Classification**
  - Accuracy - **97.19940867279895**
  - Mean Absolute Error - **0.06800262812089355**
  - Mean Squared Error - **0.20532194480946123**
  - Root Mean Squared Error - **0.4531246459965086**
  - R2 Score - **86.17743099336013**
  - `DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini', max_depth=None, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, presort='deprecated', random_state=123, splitter='best')`

K-Nearest-Neighbor

- **Binary Classification**
  - Accuracy - **98.3061287342162**
  - Mean Absolute Error - **0.016938712657838004**
  - Mean Squared Error - **0.016938712657838004**
  - Root Mean Squared Error - **0.13014880966738807**
  - R2 Score - **90.74435871039374**
  - `KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski', metric_params=None, n_jobs=None, n_neighbors=5, p=2, weights='uniform')`
- **Multi-class Classification**
  - Accuracy - **97.36777266754271**
  - Mean Absolute Error - **0.06508705650459921**
  - Mean Squared Error - **0.19411136662286466**
  - Root Mean Squared Error - **0.44058071521897624**
  - R2 Score - **86.92848100772136**
  - `KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski', metric_params=None, n_jobs=None, n_neighbors=5, p=2, weights='uniform')`

Linear Regression

- ## Binary Classification
    - Accuracy - **97.80720665229443**
    - Mean Absolute Error - **0.021927933477055742**
    - Mean Squared Error - **0.021927933477055742**
    - Root Mean Squared Error - **0.1480808342664767**
    - R2 Score - **88.20923868071647**
    - `LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)`
- ## Multi-class Classification
    - Accuracy - **95.12976346911958**
    - Mean Absolute Error - **0.06824901445466491**
    - Mean Squared Error - **0.12146846254927726**
    - Root Mean Squared Error - **0.3485232596962178**
    - R2 Score - **91.82055676180129**
    - `LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)`

Linear Support Vector Machine

- ## Binary Classification
    - Accuracy - **97.85032337542347**
    - Mean Absolute Error - **0.021496766245765322**
    - Mean Squared Error - **0.021496766245765322**
    - Root Mean Squared Error - **0.1466177555610688**
    - R2 Score - **88.45167193436498**
    - `SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='auto', kernel='linear', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False)`
- ## Multi-class Classification
    - Accuracy - **97.59362680683311**
    - Mean Absolute Error - **0.059912943495400786**
    - Mean Squared Error - **0.17941031537450722**
    - Root Mean Squared Error - **0.42356854861345317**

- R2 Score - **87.93449282205455**
- `SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='auto', kernel='linear', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False)`

Logistic Regression

- ## Binary Classification
  - Accuracy - **97.80104712041884**
  - Mean Absolute Error - **0.02198952879581152**
  - Mean Squared Error - **0.02198952879581152**
  - Root Mean Squared Error - **0.1482886671186019**
  - R2 Score - **88.17947258428785**
  - `LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, l1_ratio=None, max_iter=5000, multi_class='auto', n_jobs=None, penalty='l2', random_state=123, solver='lbfgs', tol=0.0001, verbose=0, warm_start=False)`
- ## Multi-class Classification
  - Accuracy - **97.58952036793693**
  - Mean Absolute Error - **0.060077201051248356**
  - Mean Squared Error - **0.18056011826544022**
  - Root Mean Squared Error - **0.42492366169165047**
  - R2 Score - **87.87674567880146**
  - `LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, l1_ratio=None, max_iter=5000, multi_class='multinomial', n_jobs=None, penalty='l2', random_state=123, solver='newton-cg', tol=0.0001, verbose=0, warm_start=False)`

Multi Layer Perceptron

- ## Binary Classification
  - Accuracy - **98.36772405297197**
  - Mean Absolute Error - **0.01632275947028026**
  - Mean Squared Error - **0.01632275947028026**
  - Root Mean Squared Error - **0.12776055522061674**

- R2 Score - **91.10646238100463**
- `MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08, hidden_layer_sizes=(100,), learning_rate='constant', learning_rate_init=0.001, max_fun=15000, max_iter=8000, momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5, random_state=123, shuffle=True, solver='adam', tol=0.0001, validation_fraction=0.1, verbose=False, warm_start=False)`

Multi-class Classification

- Accuracy - **97.54434954007884**
- Mean Absolute Error - **0.06065210249671485**
- Mean Squared Error - **0.17858902759526937**
- Root Mean Squared Error - **0.4225979502970517**
- R2 Score - **87.97913543550516**
- `MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08, hidden_layer_sizes=(100,), learning_rate='constant', learning_rate_init=0.001, max_fun=15000, max_iter=8000, momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5, random_state=123, shuffle=True, solver='adam', tol=0.0001, validation_fraction=0.1, verbose=False, warm_start=False)`

Random Forest Classifier

Binary Classification

- Accuracy - **98.64490298737296**
- Mean Absolute Error - **0.013550970126270403**
- Mean Squared Error - **0.013550970126270403**
- Root Mean Squared Error - **0.1164086342427846**
- R2 Score - **92.59509512345335**
- `RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, max_samples=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=None, oob_score=False, random_state=123, verbose=0, warm_start=False)`

Multi-class Classification

- Accuracy - **97.31849540078844**

- Mean Absolute Error - **0.06611366622864652**
- Mean Squared Error - **0.1985052562417871**
- Root Mean Squared Error - **0.4455392869790352**
- R2 Score - **86.6379909424011**
- `RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, max_samples=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=None, oob_score=False, random_state=50, verbose=0, warm_start=False)`

Citations

- N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," 2015 Military Communications and Information Systems Conference (MilCIS), 2015, pp. 1–6, DOI: 10.1109/MilCIS.2015.7348942.
- Nour Moustafa & Jill Slay (2016) The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set, Information Security Journal: A Global Perspective, 25:1–3, 18–31, DOI: 10.1080/19393555.2015.1125974