GEORGE WASHINGTON UNIVERSITY

ADA UNIVERSITY

COMPUTER SCIENCE AND DATA ANALYTICS

GUIDED RESEARCH I

Midterm Report

Asiman Mammadzada

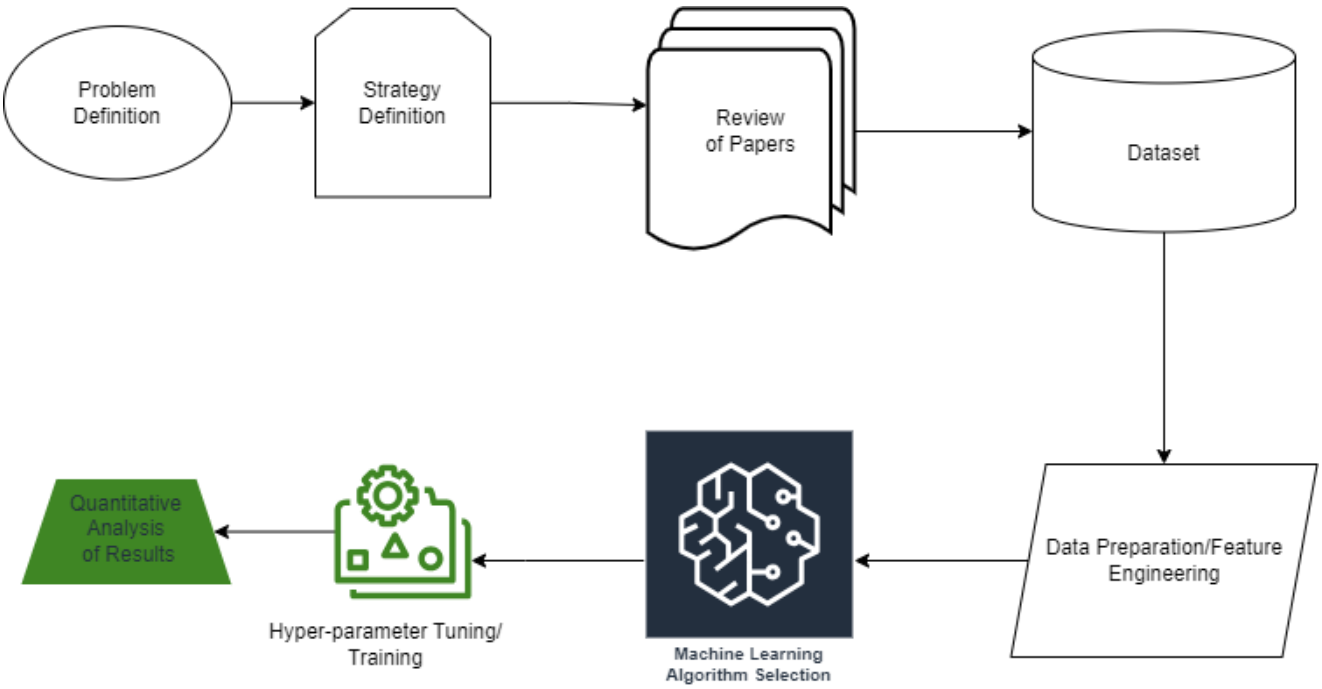
Title: Network Intrusion Detection System using Machine Learning

Instructors: Prof. Dr. Stephen Kaisler, Assoc. Prof Dr. Jamaladdin Hasanov

Problem Description

The problem aims to research possible application of machine learning algorithms in Network Intrusion Detection System (NIDS). NIDS is considerably significant to maintain the security and integrity of computer networks. Currently, the majority of NIDS is implemented on rule-based methods and signatures to identify network intrusions. Considering the evolving cyber-threats and complexity of network traffic, these rules created to detect intrusions become obsolete and less effective against sophisticated attacks. The goal of the project is to develop NIDS for specific sets of cyber-threats using Machine Learning algorithms for effective and accurate detection. The system is expected to be able to process real-time network traffic data based on learning from the historically labeled data. The ML algorithms should be trained on diverse dataset to make sure the robust identification of known network intrusions.

Roadmap



Strategy Definition:

In this project, both qualitative and quantitative approach will be utilized during the research.

The variety of ML multi-classification models will be targeted to be trained on the dataset. Having applied the hyper-parameter tuning to attain the highest accuracy, the model will be final tested on testing dataset. Qualitative approach will help to understand the contexts, nuances of network data which in fact cannot be adequately captured by quantitative approach solely. Examination of network packet payloads and their semantic content will be one aspect of qualitative research in ML application in NIDS.

In addition, quantitative research plays an important role in advancing the field of ML for NIDS. Based on the using aggregated numerical dataset, statistical analysis and modelling relying on mathematical formulas, the quantitative approach will provide an objective to deeply understand NIDS and its combination with ML. Quantitative methods in ML-NIDS research sometimes entail gathering and analyzing vast amounts of network traffic information. Packet headers, flow logs, connection logs, and other network-related data are examples of this data. Utilizing quantitative methods, researchers can find statistical trends, patterns, and anomalies in the data, providing insightful information on network intrusions.

Dataset Selection:

This implementation will be carried out using KDD Cup dataset. The KDD Cup dataset is open-source dataset which actually has 42 features on variety of intrusions. It has around 494K records of intrusions.

Dataset Preparation:

- The datasets are labeled and structured dataset. But, considering the extensive number of features, some feature engineering techniques will be applied to get the most predictive factors among them. Label Encoder technique will be applied in feature engineering and using the RandomForest the best predictors will be selected and trained in different models. One hot encoding will not be utilized in this case as when it is applied, the number of the columns is getting extensively big. Thus, label encoding is thought to be appropriate for the specific dataset.

```
label_encoder = LabelEncoder()
def labelEncode(df):
    for col in df.columns:
        if df[col].dtype == 'object':
                df[col] = label_encoder.fit_transform(df[col])

labelEncode(df)
```

- Train – Test split: in order to keep the equal distribution of train-test data with respect to each cybercrime, in other words, to keep the same portion of different crimes in test data, the following algorithm is done.
    - 0.2 portion of the whole dataset is found
    - Appended to test data
    - Indexed resetted
    - Subtraction is done.

```
test = []
train = []

for i in df['label'].unique():
    label_size = int(len(df[df['label']==i])*0.2)
    test_data = df[df['label']==i].sample(label_size)
    test.append(test_data)

df_test = pd.concat(test)
df_test.reset_index(inplace=True)
df.reset_index(inplace=True)
df_train = df[~(df['index'].isin((df_test['index'].unique())))]

df_test.drop('index', axis=1, inplace=True)
df_train.drop('index', axis=1, inplace=True)
```

- The best features selection:

```
model = RandomForestClassifier()

rfe = RFE(model, n_features_to_select=15)
rfe = rfe.fit(X_train, y_train)

feature_map = [(i, v) for i, v in itertools.zip_longest(rfe.get_support(), X_train.columns)]
selected_features = [v for i, v in feature_map if i==True]

selected_features
```

Here, using Random Forest ML algorithm method, the best features are attained which are 15 in general. Below, it is mentioned the name of the features:

```
['protocol_type',
 'service',
 'src_bytes',
 'dst_bytes',
 'wrong_fragment',
 'hot',
 'logged_in',
 'lnum_compromised',
 'count',
 'rerror_rate',
 'same_srv_rate',
 'dst_host_count',
 'dst_host_same_srv_rate',
 'dst_host_same_src_port_rate',
 'dst_host_srv_diff_host_rate']
```

Machine Learning

- Random Forest Classifier:

```
[[440   0   0   0   0   0]
 [  0 249   0   0   0   0]
 [  0   0 208   0   0   0]
 [  0   0   0 316   0   0]
 [  0   0   0   0 195   0]
 [  0   0   0   1   0 204]]
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       440
           1       1.00      1.00      1.00       249
           2       1.00      1.00      1.00       208
           3       1.00      1.00      1.00       316
           4       1.00      1.00      1.00       195
           5       1.00      1.00      1.00       205

    accuracy                           1.00      1613
   macro avg       1.00      1.00      1.00      1613
weighted avg       1.00      1.00      1.00      1613
```

```
Accuracy: 0.999
```

- Decision Tree Classifier:

```
array([[440,   0,   0,   0,   0,   2],
       [  0, 249,   0,   0,   0,   0],
       [  0,   0, 208,   1,   0,   0],
       [  0,   0,   0, 316,   0,   0],
       [  0,   0,   0,   0, 195,   0],
       [  0,   0,   0,   0,   0, 202]], dtype=int64)
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	442
1	1.00	1.00	1.00	249
2	1.00	1.00	1.00	209
3	1.00	1.00	1.00	316
4	1.00	1.00	1.00	195
5	0.99	1.00	1.00	202
accuracy			1.00	1613
macro avg	1.00	1.00	1.00	1613
weighted avg	1.00	1.00	1.00	1613

```
Accuracy: 0.999
```

- Logistic Regression

```
array( [[440   1   0   1   0  61]
       [  0   0   0   0   0   0]
       [  0 227   0   2  26 143]
       [  0  21 208 314 169   0]
       [  0   0   0   0   0   0]
       [  0   0   0   0   0   0]], dtype=int64)
```

	precision	recall	f1-score	support
0	1.00	0.87	0.93	503
1	0.00	0.00	0.00	0
2	0.00	0.00	0.00	398
3	0.99	0.44	0.61	712
4	0.00	0.00	0.00	0
5	0.00	0.00	0.00	0
accuracy			0.47	1613
macro avg	0.33	0.22	0.26	1613
weighted avg	0.75	0.47	0.56	1613

```
Accuracy: 0.467
```

The following steps to be done in the project are as followings:

- These attained results back, satan, ipsweep, portsweep, warezclient, teardrop crime types. Additional models will be built to detect smurf, Neptune, pod and nmap intrusions.
- The models will be tested provided 42 independent features as well to measure the impro vement.

In conclusion, the application of Machine Learning (ML) in Network Intrusion Detection Systems (NIDS) has shown promise in enhancing cybersecurity by effectively detecting various types of cyber intrusions. The results attained from the current stage of the study demonstrate the potential of ML algorithms, specifically the Random Forest and Decision Tree models, in achieving high accuracy rates for detecting intrusions, particularly back, satan, ipsweep, portsweep, warezclient, and teardrop crime types. These findings highlight the importance of ML as a powerful tool for bolstering the security of digital infrastructures. The obvious contrast between the accuracy rates of the Random Forest model (99.9%) and the Logistic Regression model (46.7%) underscores the importance of choosing appropriate algorithms in NIDS development. It further emphasizes that complex and non-linear relationships in network traffic data can be better captured by ensemble methods like Random Forest, resulting in superior intrusion detection performance. As the study primarily focused on detecting back, satan, ipsweep, portsweep, warezclient, and teardrop intrusions, there is a clear scope for expanding the NIDS's capabilities. Future work should include the development of additional models to detect other types of cyber intrusions, such as smurf, Neptune, pod, and nmap. This expansion will create a more comprehensive and robust NIDS, capable of addressing a broader spectrum of threats prevalent in the evolving cybersecurity landscape.

In conclusion, the successful application of ML in NIDS, as evidenced by the high accuracy rates achieved, is a testament to the technology's potential in fortifying cybersecurity defenses. The findings of this study serve as a stepping stone for further research and development in the field of NIDS, guiding future efforts to build more sophisticated and proactive defense systems. As cyber threats continue to evolve, the fusion of ML with cybersecurity will remain vital in safeguarding digital assets and ensuring a secure digital landscape for individuals and organizations alike.