520

November 28, 2023

[1]:
```
import numpy as np
import pandas as pd
import warnings
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from tensorflow.keras import regularizers
import xgboost as xgb
from sklearn.decomposition import PCA
from sklearn import tree
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import RobustScaler
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn import metrics
pd.set_option('display.max_columns',None)
warnings.filterwarnings('ignore')
%matplotlib inline
```

[4]:
```
from google.colab import files
import pandas as pd

# Read the uploaded file (assuming it's a CSV)
data_train = pd.read_csv("KDDTest-21.txt")
```

[5]:
```
# Check data
data_train.head()
```

[5]:

	13	tcp	telnet	SF	118	2425	0	0.1	0.2	0.3	0.4	1	0.5	0.6	0.7	\
0	0	udp	private	SF	44	0	0	0	0	0	0	0	0	0	0	
1	0	tcp	telnet	S3	0	44	0	0	0	0	0	0	0	0	0	
2	0	udp	private	SF	53	55	0	0	0	0	0	0	0	0	0	
3	0	tcp	private	SH	0	0	0	0	0	0	0	0	0	0	0	

```
4   0  tcp     http  SF  54540  8314  0    0    0    2    0  1    1    0    0

   0.8  0.9  0.10  0.11  0.13  0.14  0.15  1.1  1.2  0.00  0.00.1  0.00.2  \
0    0    0     0     0     0     0     0    4    3   0.0     0.0     0.0
1    0    0     0     0     0     0     0    1    1   1.0     1.0     0.0
2    0    0     0     0     0     0     0  511  511   0.0     0.0     0.0
3    0    0     0     0     0     0     0    1    1   1.0     1.0     0.0
4    0    0     0     0     0     0     0    2    9   0.0     0.0     0.5

   0.00.3  1.00  0.00.4  0.00.5   26   10  0.38  0.12  0.04  0.00.6  0.00.7  \
0    0.00  0.75     0.5    0.00  255  254  1.00  0.01  0.01     0.0    0.00
1    0.00  1.00     0.0    0.00  255   79  0.31  0.61  0.00     0.0    0.21
2    0.00  1.00     0.0    0.00  255  255  1.00  0.00  0.87     0.0    0.00
3    0.00  1.00     0.0    0.00   16    1  0.06  1.00  1.00     0.0    1.00
4    0.11  1.00     0.0    0.22  255  229  0.90  0.01  0.00     0.0    0.00

   0.00.8  0.12.1  0.30  guess_passwd   2
0    0.00    0.00   0.0     snmpguess  12
1    0.68    0.60   0.0  processtable  18
2    0.00    0.00   0.0        normal  17
3    1.00    0.00   0.0          nmap  17
4    0.00    0.01   0.0          back  10
```

[6]:
```
columns =␣
 ↪(['duration','protocol_type','service','flag','src_bytes','dst_bytes','land','wrong_fragmen
,'num_failed_logins','logged_in','num_compromised','root_shell','su_attempted','num_root','num
,'num_shells','num_access_files','num_outbound_cmds','is_host_login','is_guest_login','count',
,'srv_serror_rate','rerror_rate','srv_rerror_rate','same_srv_rate','diff_srv_rate','srv_diff_h
,'dst_host_same_srv_rate','dst_host_diff_srv_rate','dst_host_same_src_port_rate','dst_host_srv
,'dst_host_srv_serror_rate','dst_host_rerror_rate','dst_host_srv_rerror_rate','outcome','level
```

[7]:
```
# Assign name for columns
data_train.columns = columns
```

[8]:
```
data_train.head()
```

[8]:
```
   duration protocol_type  service flag  src_bytes  dst_bytes  land  \
0         0           udp  private   SF         44          0     0
1         0           tcp   telnet   S3          0         44     0
2         0           udp  private   SF         53         55     0
3         0           tcp  private   SH          0          0     0
4         0           tcp     http   SF      54540       8314     0

   wrong_fragment  urgent  hot  num_failed_logins  logged_in  num_compromised  \
0               0       0    0                  0          0                0
1               0       0    0                  0          0                0
2               0       0    0                  0          0                0
```

```
3              0       0    0                 0          0                0
4              0       0    2                 0          1                1

   root_shell  su_attempted  num_root  num_file_creations  num_shells  \
0           0             0         0                   0           0
1           0             0         0                   0           0
2           0             0         0                   0           0
3           0             0         0                   0           0
4           0             0         0                   0           0

   num_access_files  num_outbound_cmds  is_host_login  is_guest_login  count  \
0                 0                  0              0               0      4
1                 0                  0              0               0      1
2                 0                  0              0               0    511
3                 0                  0              0               0      1
4                 0                  0              0               0      2

   srv_count  serror_rate  srv_serror_rate  rerror_rate  srv_rerror_rate  \
0          3          0.0              0.0          0.0             0.00
1          1          1.0              1.0          0.0             0.00
2        511          0.0              0.0          0.0             0.00
3          1          1.0              1.0          0.0             0.00
4          9          0.0              0.0          0.5             0.11

   same_srv_rate  diff_srv_rate  srv_diff_host_rate  dst_host_count  \
0           0.75           0.5                0.00            255
1           1.00           0.0                0.00            255
2           1.00           0.0                0.00            255
3           1.00           0.0                0.00             16
4           1.00           0.0                0.22            255

   dst_host_srv_count  dst_host_same_srv_rate  dst_host_diff_srv_rate  \
0                 254                    1.00                    0.01
1                  79                    0.31                    0.61
2                 255                    1.00                    0.00
3                   1                    0.06                    1.00
4                 229                    0.90                    0.01

   dst_host_same_src_port_rate  dst_host_srv_diff_host_rate  \
0                         0.01                          0.0
1                         0.00                          0.0
2                         0.87                          0.0
3                         1.00                          0.0
4                         0.00                          0.0

   dst_host_serror_rate  dst_host_srv_serror_rate  dst_host_rerror_rate  \
0                  0.00                      0.00                  0.00
```

```
1                  0.21                       0.68                       0.60
2                  0.00                       0.00                       0.00
3                  1.00                       1.00                       0.00
4                  0.00                       0.00                       0.01

   dst_host_srv_rerror_rate       outcome  level
0                       0.0     snmpguess     12
1                       0.0  processtable     18
2                       0.0        normal     17
3                       0.0          nmap     17
4                       0.0          back     10
```

[9]:
```
data_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11849 entries, 0 to 11848
Data columns (total 43 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   duration                     11849 non-null  int64
 1   protocol_type                11849 non-null  object
 2   service                      11849 non-null  object
 3   flag                         11849 non-null  object
 4   src_bytes                    11849 non-null  int64
 5   dst_bytes                    11849 non-null  int64
 6   land                         11849 non-null  int64
 7   wrong_fragment               11849 non-null  int64
 8   urgent                       11849 non-null  int64
 9   hot                          11849 non-null  int64
 10  num_failed_logins            11849 non-null  int64
 11  logged_in                    11849 non-null  int64
 12  num_compromised              11849 non-null  int64
 13  root_shell                   11849 non-null  int64
 14  su_attempted                 11849 non-null  int64
 15  num_root                     11849 non-null  int64
 16  num_file_creations           11849 non-null  int64
 17  num_shells                   11849 non-null  int64
 18  num_access_files             11849 non-null  int64
 19  num_outbound_cmds            11849 non-null  int64
 20  is_host_login                11849 non-null  int64
 21  is_guest_login               11849 non-null  int64
 22  count                        11849 non-null  int64
 23  srv_count                    11849 non-null  int64
 24  serror_rate                  11849 non-null  float64
 25  srv_serror_rate              11849 non-null  float64
 26  rerror_rate                  11849 non-null  float64
 27  srv_rerror_rate              11849 non-null  float64
```

```
 28  same_srv_rate                11849 non-null  float64
 29  diff_srv_rate                11849 non-null  float64
 30  srv_diff_host_rate           11849 non-null  float64
 31  dst_host_count               11849 non-null  int64
 32  dst_host_srv_count           11849 non-null  int64
 33  dst_host_same_srv_rate       11849 non-null  float64
 34  dst_host_diff_srv_rate       11849 non-null  float64
 35  dst_host_same_src_port_rate  11849 non-null  float64
 36  dst_host_srv_diff_host_rate  11849 non-null  float64
 37  dst_host_serror_rate         11849 non-null  float64
 38  dst_host_srv_serror_rate     11849 non-null  float64
 39  dst_host_rerror_rate         11849 non-null  float64
 40  dst_host_srv_rerror_rate     11849 non-null  float64
 41  outcome                      11849 non-null  object
 42  level                        11849 non-null  int64
dtypes: float64(15), int64(24), object(4)
memory usage: 3.9+ MB
```

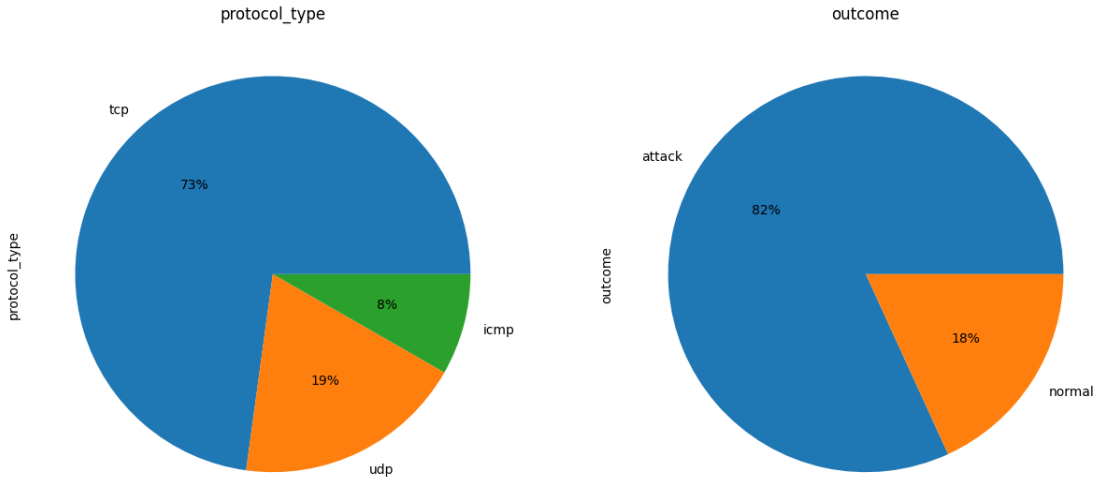
[10]:
```
data_train.describe().style.background_gradient(cmap='Blues').
 ↪set_properties(**{'font-family':'Segoe UI'})
```

[10]:
```
<pandas.io.formats.style.Styler at 0x7d83fe51f7f0>
```

[11]:
```
data_train.loc[data_train['outcome'] == "normal", "outcome"] = 'normal'
data_train.loc[data_train['outcome'] != 'normal', "outcome"] = 'attack'
```

[12]:
```
def pie_plot(df, cols_list, rows, cols):
    fig, axes = plt.subplots(rows, cols)
    for ax, col in zip(axes.ravel(), cols_list):
        df[col].value_counts().plot(ax=ax, kind='pie', figsize=(15, 15),␣
 ↪fontsize=10, autopct='%1.0f%%')
        ax.set_title(str(col), fontsize = 12)
    plt.show()
```

[13]:
```
pie_plot(data_train, ['protocol_type', 'outcome'], 1, 2)
```



[14]:
```
def Scaling(df_num, cols):
    std_scaler = RobustScaler()
    std_scaler_temp = std_scaler.fit_transform(df_num)
    std_df = pd.DataFrame(std_scaler_temp, columns =cols)
    return std_df
```

[15]:
```
cat_cols = ['is_host_login','protocol_type','service','flag','land',␣
 ↪'logged_in','is_guest_login', 'level', 'outcome']
def preprocess(dataframe):
    df_num = dataframe.drop(cat_cols, axis=1)
    num_cols = df_num.columns
    scaled_df = Scaling(df_num, num_cols)

    dataframe.drop(labels=num_cols, axis="columns", inplace=True)
    dataframe[num_cols] = scaled_df[num_cols]

    dataframe.loc[dataframe['outcome'] == "normal", "outcome"] = 0
    dataframe.loc[dataframe['outcome'] != 0, "outcome"] = 1

    dataframe = pd.get_dummies(dataframe, columns = ['protocol_type',␣
 ↪'service', 'flag'])
    return dataframe
```

[16]:
```
scaled_train = preprocess(data_train)
```

[17]:
```
x = scaled_train.drop(['outcome', 'level'] , axis = 1).values
y = scaled_train['outcome'].values
y_reg = scaled_train['level'].values
```

```
pca = PCA(n_components=20)
pca = pca.fit(x)
x_reduced = pca.transform(x)
print("Number of original features is {} and of reduced features is {}".
 ↪format(x.shape[1], x_reduced.shape[1]))

y = y.astype('int')
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,␣
 ↪random_state=42)
x_train_reduced, x_test_reduced, y_train_reduced, y_test_reduced =␣
 ↪train_test_split(x_reduced, y, test_size=0.2, random_state=42)
x_train_reg, x_test_reg, y_train_reg, y_test_reg = train_test_split(x, y_reg,␣
 ↪test_size=0.2, random_state=42)
```

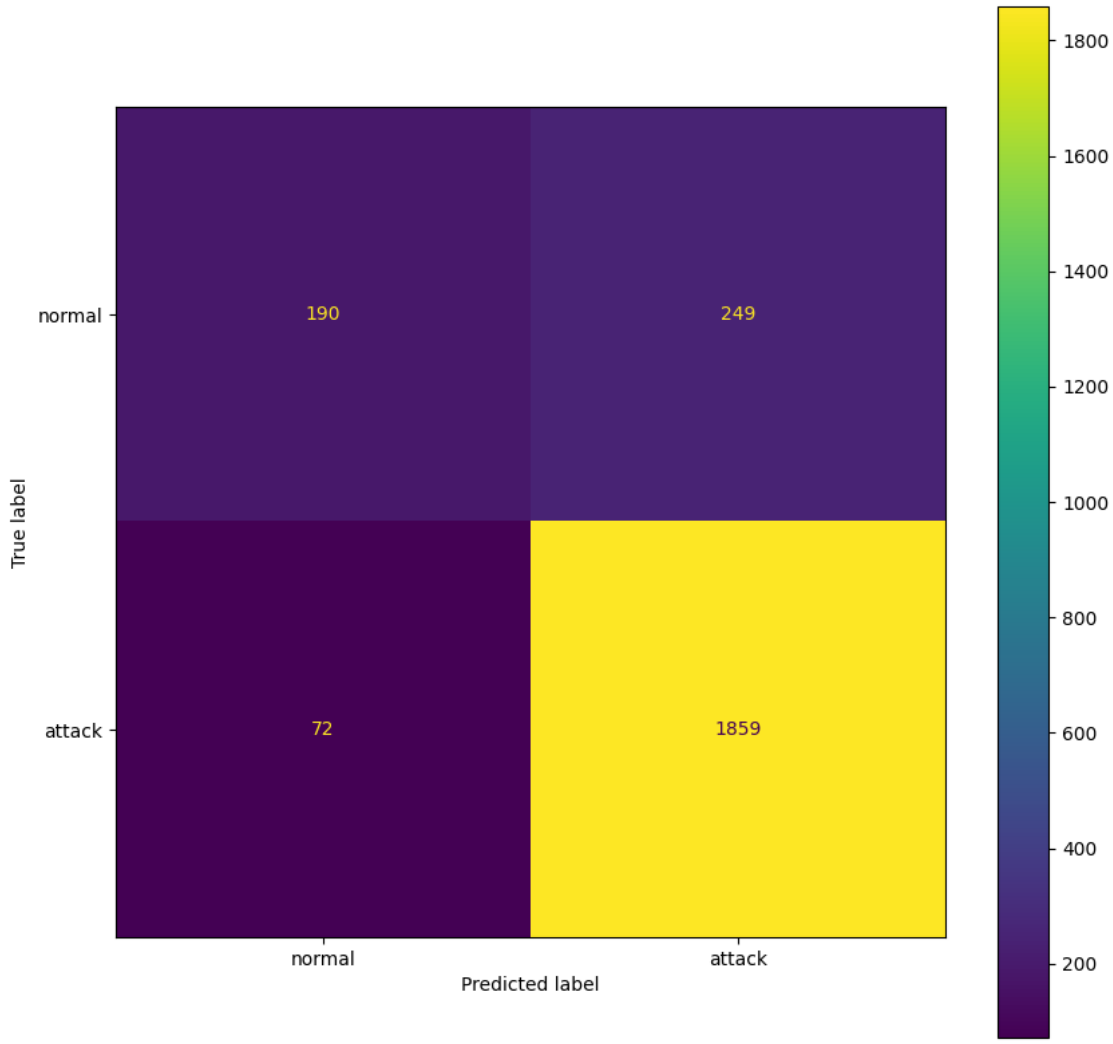
```
Number of original features is 114 and of reduced features is 20
```

[18]:
```
kernal_evals = dict()
def evaluate_classification(model, name, X_train, X_test, y_train, y_test):
    train_accuracy = metrics.accuracy_score(y_train, model.predict(X_train))
    test_accuracy = metrics.accuracy_score(y_test, model.predict(X_test))

    train_precision = metrics.precision_score(y_train, model.predict(X_train))
    test_precision = metrics.precision_score(y_test, model.predict(X_test))

    train_recall = metrics.recall_score(y_train, model.predict(X_train))
    test_recall = metrics.recall_score(y_test, model.predict(X_test))

    train_f1 = metrics.f1_score(y_train, model.predict(X_train))
    test_f1 = metrics.f1_score(y_test, model.predict(X_test))

    kernal_evals[str(name)] = [train_accuracy, test_accuracy, train_precision,␣
 ↪test_precision, train_recall, test_recall, train_f1, test_f1]
    print("Training Accuracy " + str(name) + " {}  Test Accuracy ".
 ↪format(train_accuracy*100) + str(name) + " {}".format(test_accuracy*100))
    print("Training Precesion " + str(name) + " {}  Test Precesion ".
 ↪format(train_precision*100) + str(name) + " {}".format(test_precision*100))
    print("Training Recall " + str(name) + " {}  Test Recall ".
 ↪format(train_recall*100) + str(name) + " {}".format(test_recall*100))
    print("Training F1 Score " + str(name) + " {}  Test F1 Score ".
 ↪format(train_f1*100) + str(name) + " {}".format(test_f1*100))


    actual = y_test
    predicted = model.predict(X_test)
    confusion_matrix = metrics.confusion_matrix(actual, predicted)
```

```
    cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix =␣
↪confusion_matrix, display_labels = ['normal', 'attack'])

    fig, ax = plt.subplots(figsize=(10,10))
    ax.grid(False)
    cm_display.plot(ax=ax)
```

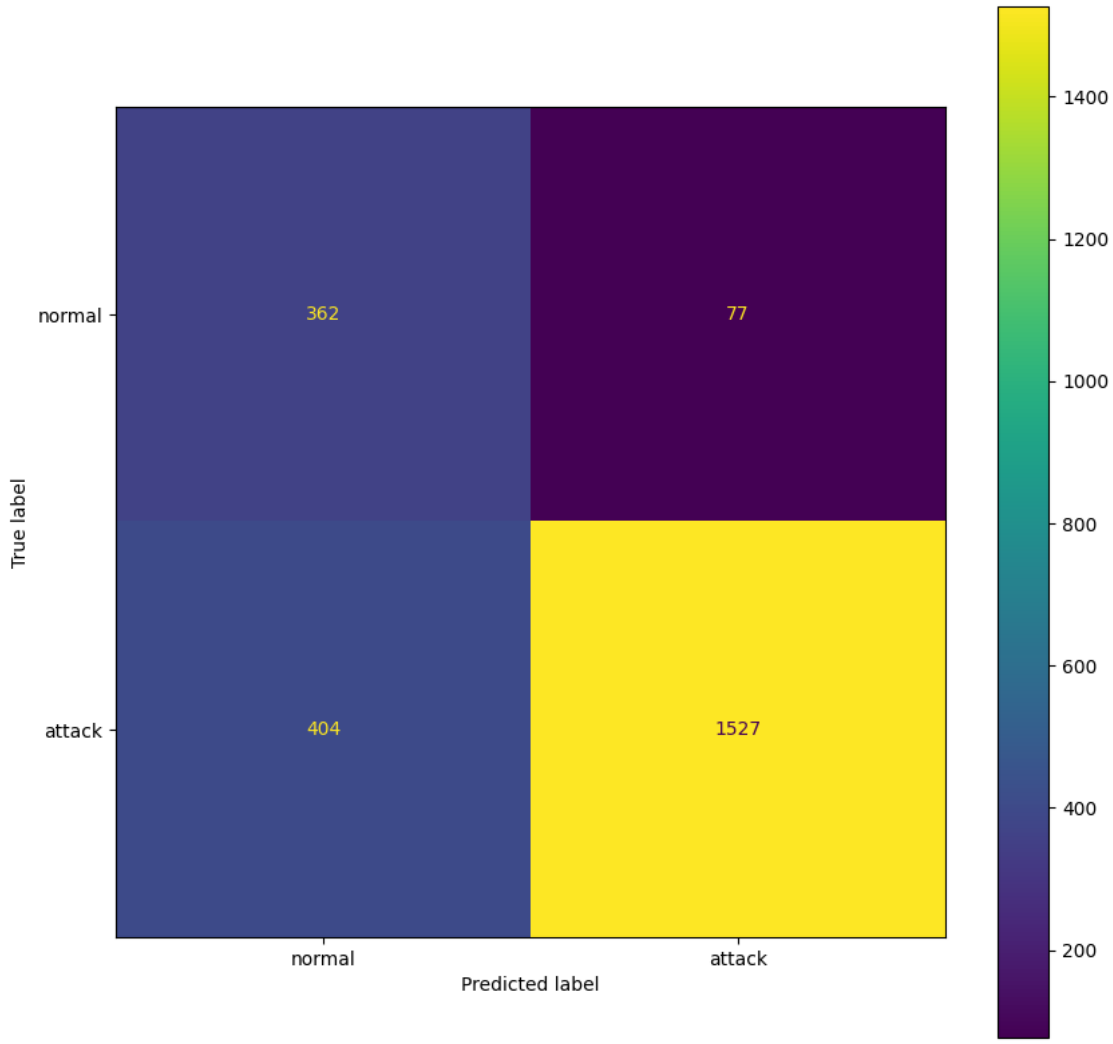
[19]:
```
lr = LogisticRegression().fit(x_train, y_train)
evaluate_classification(lr, "Logistic Regression", x_train, x_test, y_train,␣
↪y_test)
```

```
Training Accuracy Logistic Regression 86.26437387910117  Test Accuracy Logistic
Regression 86.45569620253166
Training Precesion Logistic Regression 88.21234334358005  Test Precesion
Logistic Regression 88.18785578747628
Training Recall Logistic Regression 96.0726242595931  Test Recall Logistic
Regression 96.27136198860694
Training F1 Score Logistic Regression 91.97485207100591  Test F1 Score Logistic
Regression 92.05248823966328
```



[20]:
```
gnb = GaussianNB().fit(x_train, y_train)
evaluate_classification(gnb, "GaussianNB", x_train, x_test, y_train, y_test)
```

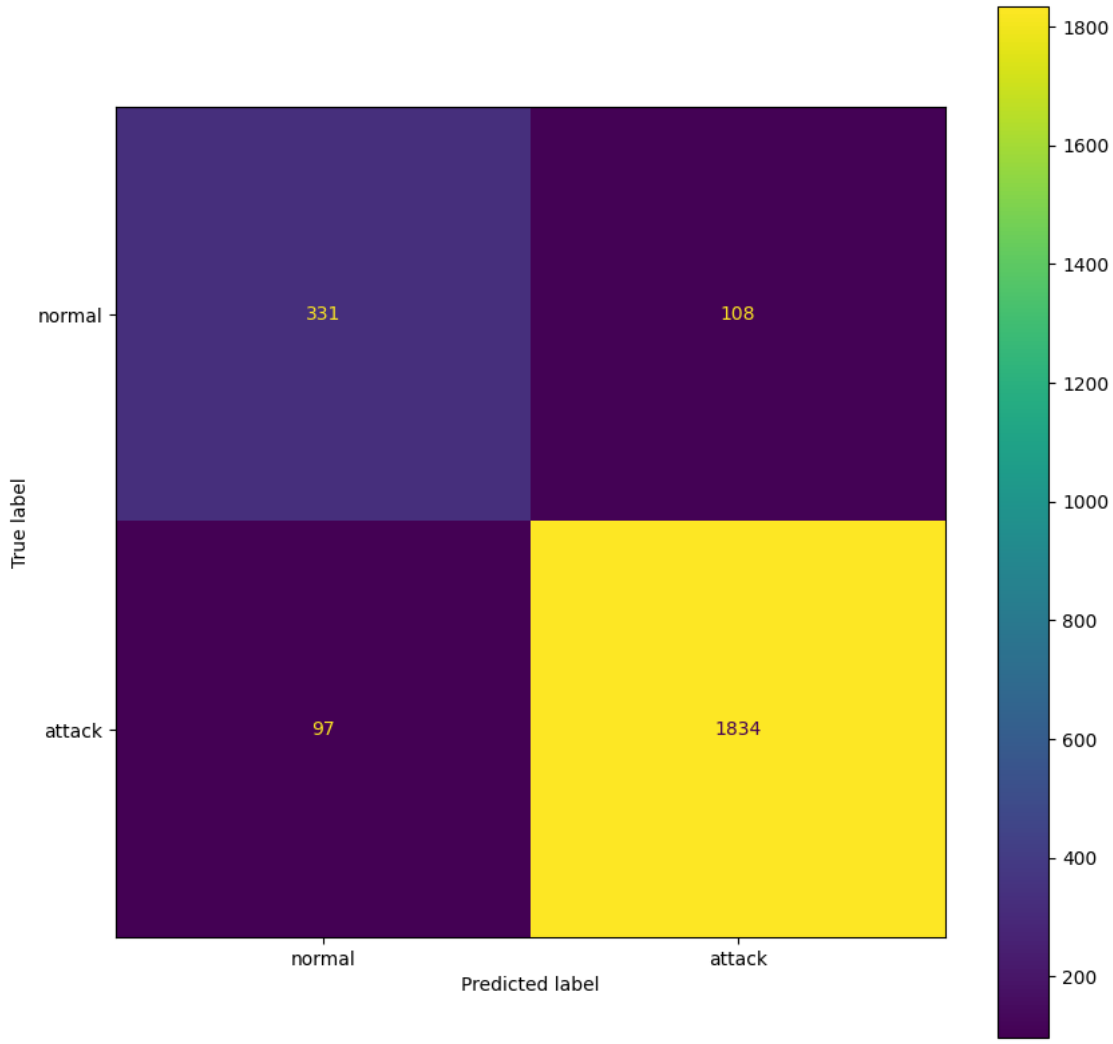
```
Training Accuracy GaussianNB 79.71304989977845  Test Accuracy GaussianNB
79.70464135021096
Training Precesion GaussianNB 95.82745098039216  Test Precesion GaussianNB
95.19950124688279
Training Recall GaussianNB 78.66340458408447  Test Recall GaussianNB
79.07819782496117
Training F1 Score GaussianNB 86.4012446078778  Test F1 Score GaussianNB
86.3932107496464
```



[21]:
```
lin_svc = svm.LinearSVC().fit(x_train, y_train)
```

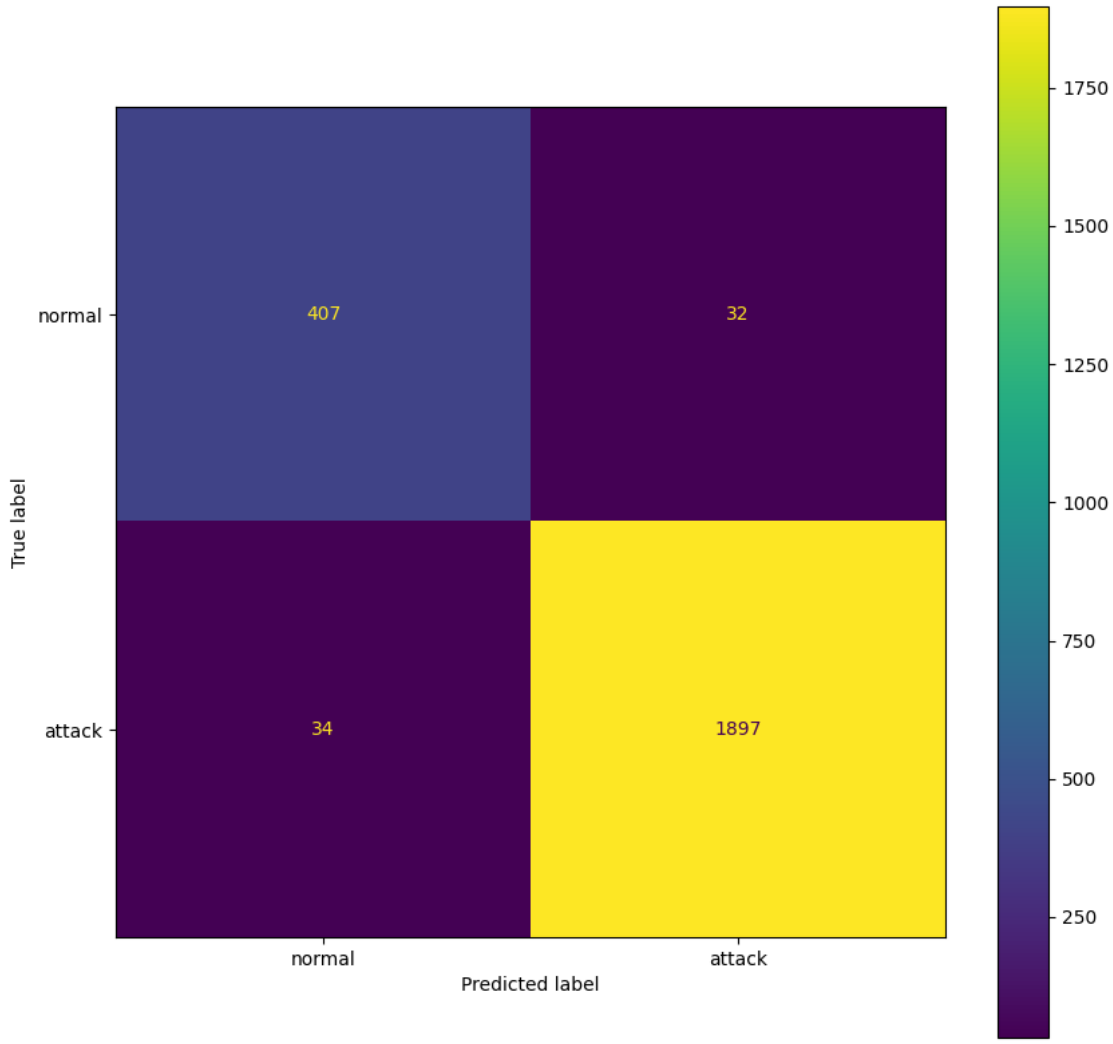
[22]:
```
evaluate_classification(lin_svc, "Linear SVC(LBasedImpl)", x_train, x_test,␣
↪y_train, y_test)
```

```
Training Accuracy Linear SVC(LBasedImpl) 91.39149699335373  Test Accuracy Linear
SVC(LBasedImpl) 91.35021097046413
Training Precesion Linear SVC(LBasedImpl) 94.0319310694374  Test Precesion
Linear SVC(LBasedImpl) 94.43872296601442
Training Recall Linear SVC(LBasedImpl) 95.55755858872006  Test Recall Linear
SVC(LBasedImpl) 94.97669601242879
Training F1 Score Linear SVC(LBasedImpl) 94.78860646314983  Test F1 Score Linear
SVC(LBasedImpl) 94.70694552026853
```



[23]:
```
dt = DecisionTreeClassifier(max_depth=3).fit(x_train, y_train)
tdt = DecisionTreeClassifier().fit(x_train, y_train)
evaluate_classification(tdt, "DecisionTreeClassifier", x_train, x_test,␣
↪y_train, y_test)
```

```
Training Accuracy DecisionTreeClassifier 99.59911383057285  Test Accuracy
DecisionTreeClassifier 97.21518987341771
Training Precesion DecisionTreeClassifier 100.0  Test Precesion
DecisionTreeClassifier 98.34110938310006
Training Recall DecisionTreeClassifier 99.51068761267061  Test Recall
DecisionTreeClassifier 98.23925427239773
Training F1 Score DecisionTreeClassifier 99.75474377178263  Test F1 Score
DecisionTreeClassifier 98.29015544041452
```

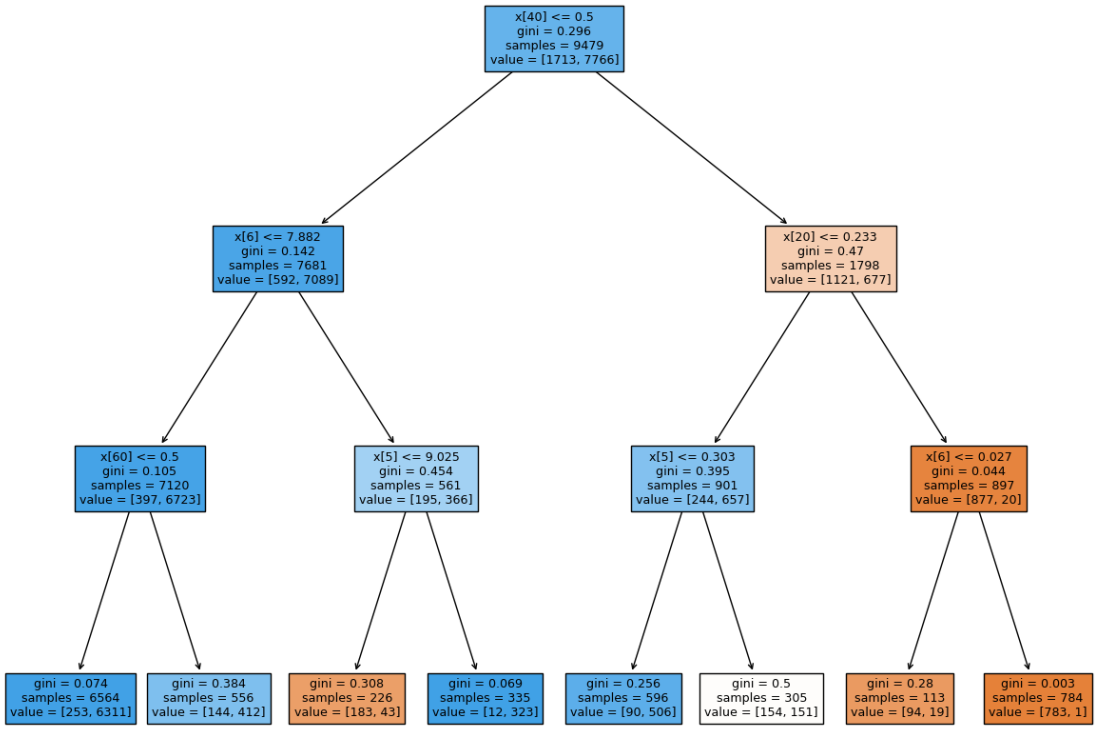


[24]:
```
def f_importances(coef, names, top=-1):
    imp = coef
    imp, names = zip(*sorted(list(zip(imp, names))))

    # Show all features
    if top == -1:
        top = len(names)

    plt.figure(figsize=(10,10))
    plt.barh(range(top), imp[::-1][0:top], align='center')
    plt.yticks(range(top), names[::-1][0:top])
    plt.title('feature importances for Decision Tree')
    plt.show()

    features_names = data_train.drop(['outcome', 'level'] , axis = 1)
```

```
    f_importances(abs(tdt.feature_importances_), features_names, top=18)
```

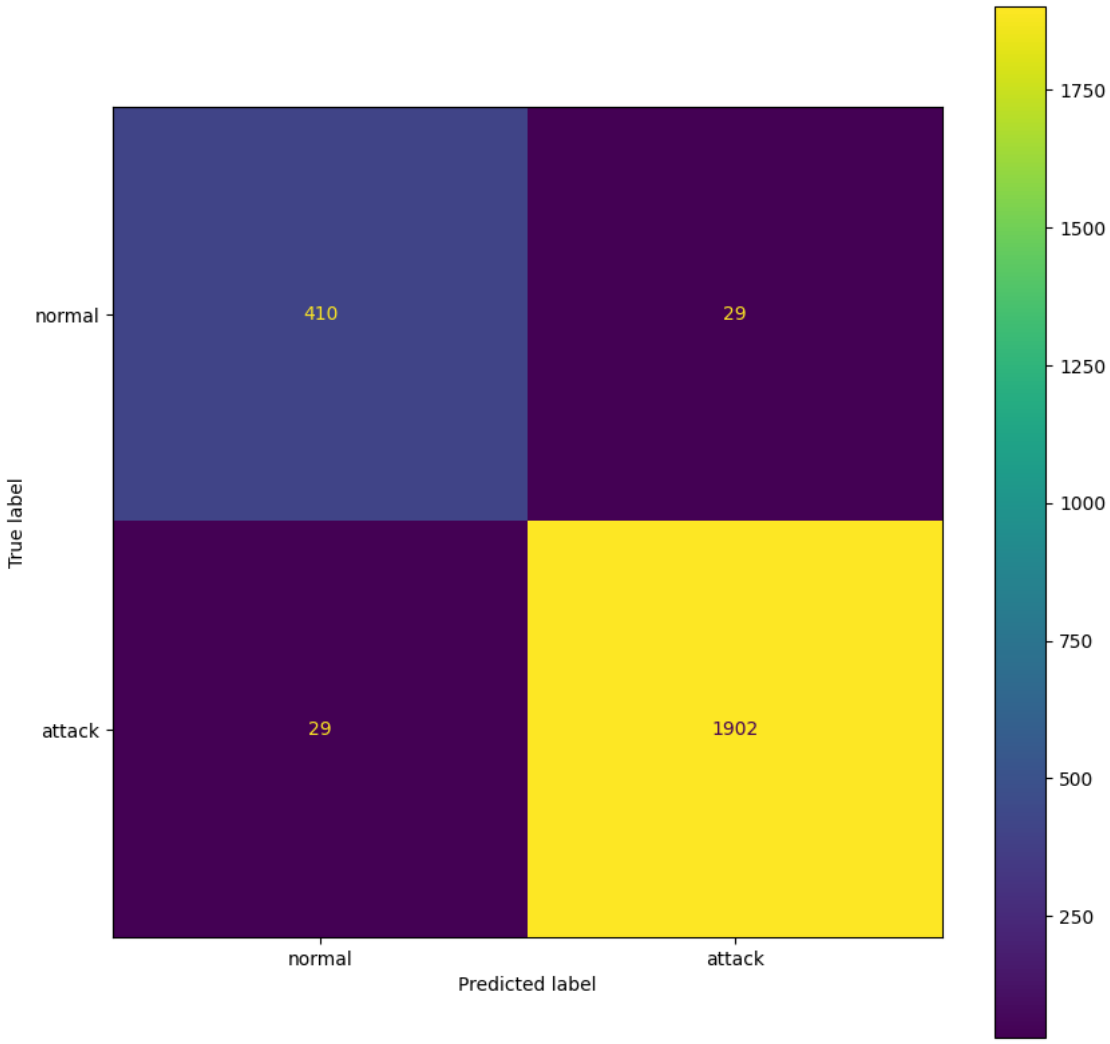
[25]:
```
fig = plt.figure(figsize=(15,12))
tree.plot_tree(dt , filled=True)
```

[25]:
```
[Text(0.5, 0.875, 'x[40] <= 0.5\ngini = 0.296\nsamples = 9479\nvalue = [1713,
7766]'),
 Text(0.25, 0.625, 'x[6] <= 7.882\ngini = 0.142\nsamples = 7681\nvalue = [592,
7089]'),
 Text(0.125, 0.375, 'x[60] <= 0.5\ngini = 0.105\nsamples = 7120\nvalue = [397,
6723]'),
 Text(0.0625, 0.125, 'gini = 0.074\nsamples = 6564\nvalue = [253, 6311]'),
 Text(0.1875, 0.125, 'gini = 0.384\nsamples = 556\nvalue = [144, 412]'),
 Text(0.375, 0.375, 'x[5] <= 9.025\ngini = 0.454\nsamples = 561\nvalue = [195,
366]'),
 Text(0.3125, 0.125, 'gini = 0.308\nsamples = 226\nvalue = [183, 43]'),
 Text(0.4375, 0.125, 'gini = 0.069\nsamples = 335\nvalue = [12, 323]'),
 Text(0.75, 0.625, 'x[20] <= 0.233\ngini = 0.47\nsamples = 1798\nvalue = [1121,
677]'),
 Text(0.625, 0.375, 'x[5] <= 0.303\ngini = 0.395\nsamples = 901\nvalue = [244,
657]'),
 Text(0.5625, 0.125, 'gini = 0.256\nsamples = 596\nvalue = [90, 506]'),
 Text(0.6875, 0.125, 'gini = 0.5\nsamples = 305\nvalue = [154, 151]'),
 Text(0.875, 0.375, 'x[6] <= 0.027\ngini = 0.044\nsamples = 897\nvalue = [877,
20]'),
 Text(0.8125, 0.125, 'gini = 0.28\nsamples = 113\nvalue = [94, 19]'),
 Text(0.9375, 0.125, 'gini = 0.003\nsamples = 784\nvalue = [783, 1]')]
```



[26]:
```
rf = RandomForestClassifier().fit(x_train, y_train)
evaluate_classification(rf, "RandomForestClassifier", x_train, x_test, y_train,␣
↪y_test)
```

```
Training Accuracy RandomForestClassifier 99.59911383057285  Test Accuracy
RandomForestClassifier 97.55274261603375
Training Precesion RandomForestClassifier 99.78098428240145  Test Precesion
RandomForestClassifier 98.49818746763334
Training Recall RandomForestClassifier 99.72959052279165  Test Recall
RandomForestClassifier 98.49818746763334
Training F1 Score RandomForestClassifier 99.75528078310148  Test F1 Score
RandomForestClassifier 98.49818746763334
```


410
29
29
1902
normal
attack
True label
normal
attack
Predicted label
1750
1500
1250
1000
750
500
250