Machine learning cybersecurity

**Fraud detection**

# LAB 9: Writing a classifier for CREDIT CARD FRAUD Solution

### **Lab Exercise 1**

In this exercise, WEKA will be used for the classification.

1. Open WEKA and select the Explorer in the applications section

A screenshot of a cell phone

Description automatically generated

1. Open file and read the data. There are 284,807 records have been read from the credit card fraud dataset.

A screenshot of a cell phone

Description automatically generated

1. Convert the type of class value from numeric to nominal

A screenshot of a cell phone

Description automatically generated

1. In classify section, select J48 as the classifier and set the configuration

A screenshot of a cell phone

Description automatically generated

1. For the test option, 10-folds cross-validation has been chosen. Click the start button to perform the classification

A picture containing text, receipt

Description automatically generated

### **Lab Exercise 2**

The task of lab is to implement different classifiers with sklearn package

1. Import the required libraries

A screenshot of a cell phone

Description automatically generated

1. Encode the labels for class values, standard scale the feature values and split the dataset.

A screenshot of a cell phone

Description automatically generated

1. Define the function to print statistics metrics

A screenshot of a cell phone

Description automatically generated

1. Specify the type of classifier, then fit it with the training set and perform the prediction with testing set.

Finally, print out the statistics.

A screenshot of a cell phone

Description automatically generated

1. Results

A close up of text on a white background

Description automatically generated

A screenshot of text

Description automatically generated

### **Lab Exercise 3**

In Exercise 3, tensorflow is applied for the deep-learning-based classifier.

1. Import the required libraries

import tensorflow as tf
import numpy as np
import pandas as pd
from numpy import genfromtxt
from sklearn import datasets
from sklearn.datasets import fetch_mldata
from sklearn.model_selection import train_test_split
import sklearn
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score, f1_score
import pandas as pd

1. Define the parameters

A screenshot of a cell phone

Description automatically generated

1. Read the features and class values from dataset

featuer_col = np.arange(30)
feature = pd.read_csv('creditcard.csv',delimiter=',',usecols=(featuer_col),dtype=float,skiprows=1).values
target=pd.read_csv('creditcard.csv',delimiter=',',usecols=([30]),dtype=str,skiprows=1).values


1. Encode the labels for class values, standard scale the feature values and split the dataset as well as the one hot encoding for class values

A screenshot of a cell phone

Description automatically generated

A close up of a knife

Description automatically generated with low confidence

1. Define the architecture for deep neural network

Since the number of input features is 30, so the number of neurons in each layer should no more than 30 or less than 30.

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generated

1. Define the function to print statistics metrics

A screenshot of a cell phone

Description automatically generated

1. Define the placeholder and perform the deep learning for credit card fruad dataset

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generated

1. Results

The 33th epoch:

A screenshot of text

Description automatically generated

The last epoch:

A screenshot of a computer

Description automatically generated

As we can see, all the statistics metrics have been improved after 10,000 epochs.