Task 1 (10 points)

Describe the Feed-forward neural network. The answer should be comprehensive. (5 points)

The practical example of application in cybersecurity with data and the python code are mandatory. (5 points)

Place your answer into the task1.pdf file. The source code and the data should be included in the resulting pdf file. No additional files are required.

Feed-Forward Neural Networks (FNNs)

Feed-forward neural networks (FNNs) are a type of artificial neural network where the information flow is unidirectional, moving from the input layer through the hidden layers to the output layer, without any cycles or loops in the network. These networks are called "feed-forward" because the data feeds forward through the network, with no backtracking or feedback connections.

The architecture of an FNN typically consists of the following components:

Input Layer: This layer receives the initial input data or features.

Hidden Layers: These are the intermediate layers between the input and output layers. They perform computations and transformations on the input data. There can be one or multiple hidden layers, depending on the complexity of the problem.

Output Layer: This layer produces the final output or prediction of the network, based on the computations performed by the hidden layers.

Each layer is composed of multiple nodes or neurons, and each node in a layer is connected to every node in the subsequent layer through weighted connections. These connection weights are adjusted during the training process to minimize the error between the predicted output and the actual target output.

The computations within each node involve taking the weighted sum of the inputs and applying an activation function, such as the sigmoid, tanh, or ReLU function, to introduce non-linearity and enable the network to model complex relationships in the data.

Cybersecurity Application: Network Intrusion Detection

One practical application of FNNs in cybersecurity is network intrusion detection. In this scenario, the network traffic data, including features like source and destination IP addresses, ports, protocols, and packet payloads, can be used as input to an FNN for classifying whether the traffic is normal or an attack.

Sample Data:

Assume we have a dataset containing network traffic data with the following features:

Source IP

Destination IP

Source Port

Destination Port

Protocol

Payload Size

We can generate a small random sample dataset:

```
import numpy as np
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Generate toy data
data = np.random.randint(0, 256, size=(1000, 6))
labels = np.random.randint(0, 2, size=(1000, 1))

# Split into train/test
X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2)

# Build FNN model
model = Sequential()
```

```
model.add(Dense(64, activation='relu', input_shape=(6,)))  # Input layer
model.add(Dense(32, activation='relu'))  # Hidden layer
model.add(Dense(16, activation='relu'))  # Hidden layer
model.add(Dense(1, activation='sigmoid'))  # Output layer

# Compile and train
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(X_train, y_train, epochs=20, batch_size=32, validation_data=(X_test, y_test))
```

This code builds an FNN with 1 input layer, 2 hidden layers, and 1 output layer to classify network traffic as normal or an attack based on the 6 input features. The model is trained on the generated toy data.