**Machine learning for network threat detection**

**Block 1**

1. Describe the main types of Internet threats that can be detected using machine learning methods.
2. Describe stages classifications DDoS, Man in the Middle, Malware, Phishing, SQL injection threats. Describe classification algorithms Support Vector Machine and k - nearest neighbors method.
3. Describe the classification methods of DDoS, Man in the Middle, Malware, Phishing, SQL injection threats. Describe Decision classification algorithms Tree and Random Forest.
4. Describe stages classifications DDoS, Man in the Middle, Malware, Phishing, SQL injection threats. Describe the XgBoost classification algorithms and CatBoost.
5. Describe what types of neural networks are commonly used for DDoS classification, Man in the Middle, Malware, Phishing, SQL injection threats​

**Block 2**

1. Describe the application of convolutional neural networks (CNN) for DDoS classification, Man in the Middle, Malware, Phishing, SQL injection threats​
2. Explain how recurrent neural networks (RNN) can be used for DDoS classification, Man in the Middle, Malware, Phishing, SQL injection threats​
3. Describe the LSTM (Long Short-Term Memory) neural network and why they are effective in DDoS classification tasks, Man in the Middle, Malware, Phishing, SQL injection threats​
4. Describe BERT (Bidirectional Encoder Representations from Transformers) model for classifications DDoS, Man in the Middle, Malware, Phishing, SQL injection threats.
5. Explain what methods can be used to evaluate the quality of a model trained on a DDoS detection task. Man in the Middle, Malware, Phishing, SQL injection threats.

**Block 3**

1. Create a small data classification script using the Scikit - learn library.
2. A small script is given. Complete it with a command for classification using Decision Tree

import tensorflow as tf

from tensorflow.keras .preprocessing.text import Tokenizer

from tensorflow.keras .preprocessing.sequence import pad\_sequences

from tensorflow.keras .models import Sequential

from tensorflow.keras .layers import Dense, Embedding, GlobalAveragePooling1D

# Example data

reviews = [

"Great movie! Must watch.",

"Worst movie ever."

"It was a fantastic performance!",

"Not a good movie, boring plot.",

"The cinematography is pretty great.",

 "I hated the storyline ."

]

labels = [1, 0, 1, 0, 1, 0 ] # 1 - positive, 0 - negative

1. A small script is given. Complete it with a classification command using Random Forest.

import tensorflow as tf

from tensorflow.keras .preprocessing.text import Tokenizer

from tensorflow.keras .preprocessing.sequence import pad\_sequences

from tensorflow.keras .models import Sequential

from tensorflow.keras .layers import Dense, Embedding, GlobalAveragePooling1D

# Example data

reviews = [

"Great movie! Must watch.",

"Worst movie ever."

"It was a fantastic performance!",

"Not a good movie, boring plot.",

"The cinematography is pretty great.",

 "I hated the storyline ."

]

labels = [1, 0, 1, 0, 1, 0 ] # 1 - positive, 0 - negative

1. A small script is given. Complete it with a command for classification using Dense neural network

import tensorflow as tf

from tensorflow.keras .preprocessing.text import Tokenizer

from tensorflow.keras .preprocessing.sequence import pad\_sequences

from tensorflow.keras .models import Sequential

from tensorflow.keras .layers import Dense, Embedding, GlobalAveragePooling1D

# Example data

reviews = [

"Great movie! Must watch.",

"Worst movie ever."

"It was a fantastic performance!",

"Not a good movie, boring plot.",

"The cinematography is pretty great.",

 "I hated the storyline ."

]

labels = [1, 0, 1, 0, 1, 0 ] # 1 - positive, 0 - negative

1. A small script is given. Complete it with a command for classification using Long short-term memory neural network

import tensorflow as tf

from tensorflow.keras .preprocessing.text import Tokenizer

from tensorflow.keras .preprocessing.sequence import pad\_sequences

from tensorflow.keras .models import Sequential

from tensorflow.keras .layers import Dense, Embedding, GlobalAveragePooling1D

# Example data

reviews = [

"Great movie! Must watch.",

"Worst movie ever."

"It was a fantastic performance!",

"Not a good movie, boring plot.",

"The cinematography is pretty great.",

 "I hated the storyline ."

]

labels = [1, 0, 1, 0, 1, 0 ] # 1 - positive, 0 - negative