

Лабораторная работа 3

Анализ DDoS атак моделями глубокого обучения

Дан датасет DDoS атак

<https://www.kaggle.com/code/chitrakhsingh/ddos-sdn>

Разработать и протестировать модели нейронных сетей для классификации DDoS-атак на основе датасета DDoS-SDN. Провести сравнительный анализ различных архитектур нейронных сетей.

Этапы выполнения

1. Подготовка данных

1. Загрузка данных

- Скачать датасет с Kaggle и загрузить в среду разработки (Google Colab, Jupyter Notebook, PyCharm).
- Использовать pandas и numpy для обработки данных.

2. Анализ данных

- Определить целевую переменную и признаки.
- Исследовать баланс классов (value_counts()).
- Проверить наличие пропущенных значений и удалить/заменить их.

3. Предобработка данных

- Кодирование категориальных признаков (LabelEncoder, OneHotEncoder).
- Масштабирование числовых признаков (StandardScaler, MinMaxScaler).
- Разделение на обучающую и тестовую выборки (train_test_split).

2. Обучение нейронных сетей

Реализовать и обучить три типа нейронных сетей:

2.1 Полносвязная нейронная сеть (Dense Neural Network, DNN)

Архитектура:

- Входной слой (Input Layer).
- Несколько скрытых слоев с Dense и ReLU.
- Dropout (для предотвращения переобучения).
- Выходной слой с sigmoid (если бинарная классификация) или softmax (если многоклассовая).

Библиотеки:

- TensorFlow/Keras
- Dense из tf.keras.layers

Гиперпараметры для настройки:

- Количество слоев и нейронов.
- learning_rate (оптимизатор Adam).
- batch_size, epochs.

2.2 Сверточная нейронная сеть (Convolutional Neural Network, CNN)

Архитектура:

- Входной слой (Input Layer), преобразующий данные в 2D-матрицу.
- Conv1D слои для обработки временных последовательностей.
- BatchNormalization и ReLU для улучшения сходимости.
- MaxPooling1D для снижения размерности.
- Flatten и Dense для классификации.

Библиотеки:

- Conv1D, MaxPooling1D, Flatten, Dense из tf.keras.layers.

Гиперпараметры:

- Количество фильтров и размер ядра в Conv1D.
- Размерность MaxPooling1D.
- Количество Dense-слоев.

2.3 Рекуррентная нейронная сеть (Recurrent Neural Network, RNN)

Архитектура:

- LSTM или GRU для работы с временными рядами.
- Dropout и BatchNormalization для регуляризации.
- Dense слой для классификации.

Библиотеки:

- LSTM, GRU из tf.keras.layers.

Гиперпараметры:

- Количество LSTM-нейронов.
- Количество слоев LSTM.
- Размер batch_size.

3. Оценка моделей

- 1. Метрики качества**
 - accuracy
 - precision

- recall
 - F1-score
 - ROC-AUC
2. **Кросс-валидация**
 - Использование KFold или StratifiedKFold.
 3. **Визуализация обучения**
 - Построение loss и accuracy по эпохам.

4. Анализ и выводы

1. Сравнить результаты всех моделей:
 - Какая архитектура работает лучше?
 - Время обучения каждой модели.
 - Как обработка входных данных влияет на результат?
2. Сделать выводы о применимости нейросетевых моделей к задаче DDoS-классификации.

5. Требования к отчету

1. Код с комментариями.
2. Графики и таблицы с результатами.
3. Описание результатов и выводы.