Лабораторная работа 4

Анализ Man in the Middle атак моделями машинного обучения

Дан датасет MitM атак

https://www.kaggle.com/datasets/ymirsky/network-attack-dataset-kitsune

Разработать и протестировать модели машинного обучения для классификации атак типа «Man in the Middle» (MitM). Провести сравнительный анализ различных алгоритмов.

Этапы выполнения

1. Подготовка данных

1. **Загрузка данных**
   - Скачать датасет с Kaggle и загрузить в среду разработки (Jupyter Notebook, PyCharm или Google Colab).
   - Использовать библиотеки pandas и numpy для обработки данных.
2. **Анализ данных**
   - Определить целевую переменную (метка атаки).
   - Исследовать распределение классов (количество атак MitM vs нормальный трафик).
   - Проверить наличие пропущенных значений (df.isnull().sum()).
3. **Предобработка данных**
   - Обработать пропущенные значения (если есть).
   - Масштабировать числовые признаки (StandardScaler или MinMaxScaler).
   - Разделить данные на обучающую и тестовую выборки (train_test_split).

2. Обучение моделей машинного обучения

Обучить и протестировать следующие модели:

1. **Наивный Байесовский классификатор (Naïve Bayes)**
   - GaussianNB или MultinomialNB из sklearn.naive_bayes.
   - Проверить, подходит ли модель к данному набору данных.
2. **Логистическая регрессия (Logistic Regression)**
   - LogisticRegression из sklearn.linear_model.
   - Проверить влияние параметров (penalty, C).
3. **Метод опорных векторов (Support Vector Machine, SVM)**
   - SVC из sklearn.svm с разными ядрами (linear, rbf).
   - Оптимизировать C и gamma.
4. **Метод k-ближайших соседей (k-Nearest Neighbors, k-NN)**
   - KNeighborsClassifier из sklearn.neighbors.
   - Оптимизация k с помощью кросс-валидации.
5. **Дерево решений (Decision Tree)**
   - DecisionTreeClassifier из sklearn.tree.
   - Настроить max_depth, criterion.
6. **Случайный лес (Random Forest)**
   - RandomForestClassifier из sklearn.ensemble.
   - Оптимизация n_estimators, max_depth.
7. **Градиентный бустинг (XGBoost)**
   - XGBClassifier из xgboost.

- Оптимизация learning_rate, n_estimators, max_depth.

8. **CatBoost**
   - CatBoostClassifier из catboost.
   - Оптимизация iterations, depth, learning_rate.
9. **AdaBoost**
   - AdaBoostClassifier из sklearn.ensemble.
   - Подбор n_estimators, learning_rate.

3. Оценка моделей

1. **Метрики качества**
   - accuracy
   - precision
   - recall
   - F1-score
   - ROC-AUC
2. **Кросс-валидация**
   - KFold или StratifiedKFold для проверки устойчивости моделей.
3. **Матрица ошибок (Confusion Matrix)**
   - confusion_matrix для анализа ошибок классификации.

4. Визуализация результатов

1. **Графики:**
   - Сравнение accuracy моделей.
   - ROC-кривые для нескольких моделей.
   - Матрицы ошибок лучших моделей.

5. Анализ и выводы

1. Сравнить модели:
   - Какая модель показала наилучший результат?
   - Время обучения моделей.
   - Какие модели лучше справляются с данной задачей?
   - Влияют ли признаки на качество классификации?
2. Сделать вывод о применимости машинного обучения для выявления атак MitM.