Kymbat Zhumagulkyzy Seilkhanova – teacher of the Department of Intelligent Systems and Cybersecurity; Astana IT University, Astana, Republic of Kazakhstan; e-mail: K.Seilkhanova@astanait.edu.kz.
Askhat Mergenbaiuly Zhetkerbay – teacher of the Department of Intelligent Systems and Cybersecurity; Astana IT University, Astana, Republic of Kazakhstan; e-mail: A.Zhetkerbay@astanait.edu.kz.

Редакцияға енуі 17.01.2025
Өңдеуден кейін түсуі 04.04.2025
Жариялауға қабылданды 07.04.2025

https://doi.org/10.53360/2788-7995-2025-2(18)-2

МРНТИ: 81.93.29

С. Адилжанова, М. Кунелбаев, Д. Сыбанова*
Казахский национальный университет имени аль-Фараби,
050040 Республика Казахстан, г. Алматы, пр. аль-Фараби, 71
*e-mail:dsybanovaa@gmail.com

ПРИМЕНЕНИЕ МАШИННОГО ОБУЧЕНИЯ ДЛЯ АНАЛИЗА КИБЕРАТАК: ИССЛЕДОВАНИЕ НА ОСНОВЕ ДАТАСЕТА RT-IOT 2022

Аннотация: *Статья посвящена исследованию применения машинного обучения для анализа кибератак. В исследовании рассматриваются алгоритмы Random Forest, SVM и Logistic Regression, которые успешно справляются с задачами выявления аномалий и минимизации ложных срабатываний. Адаптация моделей к работе с несбалансированными данными, таких как использование LabelEncoder для категориальных признаков и StandardScaler для стандартизации данных, позволила значительно улучшить их производительность. На основе анализа данных из набора «Real-Time Internet of Things (RT-IoT 2022)» представлены результаты точности и устойчивости моделей. Основное внимание уделяется защите от киберугроз, включая утечки информации, DDoS-атаки и другие виды угроз. Анализ различных алгоритмов машинного обучения для исследования кибератак показал значимые результаты. Random Forest продемонстрировала наивысшую точность – 99,86%, обеспечивая высокую стабильность и эффективность в классификации различных видов угроз. SVM показала точность 99,29%, справляясь с большинством сложных классов. Logistic Regression продемонстрировала удовлетворительные результаты с точностью 97,71%, хотя в некоторых редких случаях точность была ниже. Таким образом, Random Forest и SVM продемонстрировали наилучшую эффективность для задач безопасности и анализа кибератак в цифровых системах, обеспечивая высокую точность и надежность. В дальнейшем планируется внедрение более сложных методов, таких как глубокое обучение, для более точного определения и анализа угроз.*

Ключевые слова: *Кибератака, алгоритмы машинного обучения,Random Forest, SVM, Logistic Regression, RT-IoT 2022, обнаружение атак.*

Введение

Данное исследование комбинирует правила обнаружения с методами машинного обучения для снижения Distributed Denial of Service (DDoS) атак в киберфизических производственных системах (CPPS), демонстрируя высокую точность и эффективность в реальном времени при обучении на реальном сетевом трафике [1]. Кроме того, предлагается гибридная глубокая нейронная сеть (DNN) для обнаружения и классификации DDoS атак в Software-Defined IIoT сетях, используя XGBoost для выбора признаков и сочетая CNN и LSTM сети. Этот подход обеспечивает высокую точность и низкую латентность, что критично для среды IIoT [2]. Для детекции аномалий в системах Индустрии 4.0 предлагается гибридная объединенная машина для обучения, включающая модели LOF, One-Class SVM и автоэнкодеры для повышения точности обнаружения. Эффективность системы подтверждается посредством оценок производительности на промышленных машинах [3]. Также представлен человеко-кибер-физический комплекс (HCPS) для реального времени обнаружения аномалий, объединяющий знания человека с киберфизическими системами для улучшения принятия решений, надежности системы и создания рабочих мест в Индустрии 5.0 [4]. Роль глубокого обучения в детекции мошенничества с использованием кредитных карт также анализируется, выделяя различные архитектуры и их влияние на улучшение точности детекции мошенничества, с фокусом на преодоление таких вызовов, как несбалансированность данных и переобучение [5]. Машинное обучение и глубокое обучение

играют критически важную роль в оптимизации детекции аномалий, распределения ресурсов и предиктивных моделей в сетях цифровых двойников (DTN), революционизируя отрасли и усиливая производительность систем [6]. Также исследуются угрозы безопасности в сельскохозяйственных технологиях 4.0 и 5.0, предлагаются стратегии минимизации рисков с использованием искусственного интеллекта, блокчейна и квантовых вычислений для улучшения детекции вредоносных программ и предотвращения атак типа Denial-of-Service (DoS) [7]. Детекция мошенничества с использованием кредитных карт анализируется с использованием алгоритмов машинного и глубокого обучения, подчеркивая эффективность глубокого обучения в решении проблем, таких как несбалансированность данных и высокие ставки ложных срабатываний, достигая точности 99,9% и значительных улучшений в производительности детекции мошенничества [8]. Также представлена инновационная система детекции мошенничества, использующая анти-Бенфорд графы и алгоритмы машинного обучения для повышения безопасности финансовых сетей, достигая точности 94,83% в детекции мошенничества [9]. Для криптовалютных систем предложен метод на основе подхода trimmed k-means для детекции мошенничества в сетях Биткойн, предлагая надежное решение для выявления мошеннических транзакций [10]. Исследуется цифровое доверие в контексте Индустрии 4.0 и 5.0, подчеркивается важность предотвращения мошенничества и защиты данных для поддержания доверия пользователей и обеспечения устойчивости цифровых технологий [11]. SQL инъекции (SQLI) решаются с использованием нового фреймворка под названием DIAVA, который использует анализ сетевого трафика и регулярные выражения для точного обнаружения атак, значительно превосходя традиционные веб-решения защиты [12]. Детекция мошенничества с использованием кредитных карт дополнительно усиливается комбинированным использованием LSTM и GRU нейронных сетей с многослойными перцептронами, достигая высокой точности и минимизации ложных срабатываний [13]. Наконец, обсуждается использование машинного обучения и киберугрозной разведки (CTI) для улучшения безопасности в кибер цепях поставок, прогнозирования угроз и разработки контрольных мер для повышения безопасности [14]. Детекция и защита от DDoS атак в SDN также улучшаются с использованием метода онлайн машинного обучения, обеспечивая более высокую точность обнаружения и надежную защиту от низкоуровневых и нулевых атак [15].

Цель статьи заключается в исследовании возможностей машинного обучения для анализа кибератак в рамках Индустрии 4.0, а также в разработке и применении методов для повышения безопасности цифровых систем. Основное внимание уделяется оценке различных алгоритмов, таких как Random Forest, SVM и Logistic Regression, в контексте защиты от киберугроз и повышения точности их классификации.

Метод

Для решения задачи классификации данных разработана методология, включающая несколько ключевых этапов. На этапе предварительной обработки данные были очищены и подготовлены для использования в обучении моделей. Затем были обучены три алгоритма машинного обучения: случайный лес, опорные векторные машины и логистическая регрессия. Их эффективность оценивалась на основе точности классификации и способности обрабатывать редкие классы. Процесс построения, обучения и оценки моделей представлен на нижеуказанной блок-схеме (рис. 1).

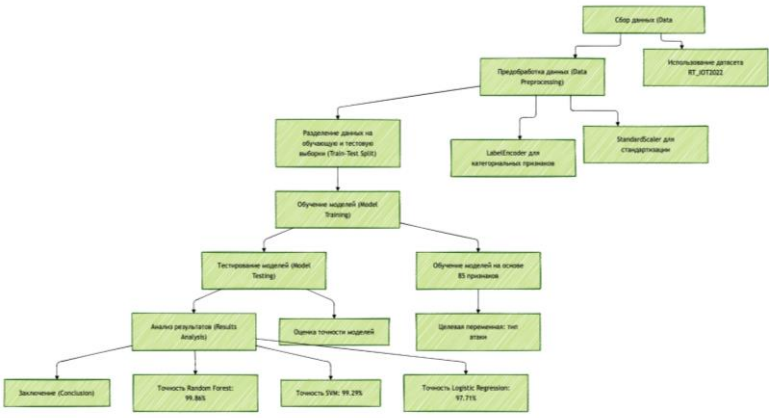

Рисунок 1 – Блок-схема процесса классификации данных

Диаграмма отображает этапы построения моделей машинного обучения, начиная с предварительной обработки данных, обучения трёх различных алгоритмов и их оценки на основе точности. Выбор финальной модели основывался на установленных критериях производительности.

Гипотеза исследования заключается в том, что использование алгоритмов машинного обучения, таких как Random Forest, SVM и Logistic Regression, будет способствовать улучшению анализа кибератак в рамках Индустрии 4.0, обеспечивая высокую точность и устойчивость в распознавании аномалий и угроз.

Научная новизна данного исследования заключается в глубоком анализе применения машинного обучения для классификации редких и сложных кибератак в рамках индустрии 4.0. В работе рассматриваются алгоритмы Random Forest, SVM и Logistic Regression, которые эффективно справляются с задачей обнаружения аномалий и минимизации ложных срабатываний. Адаптация этих моделей для работы с несбалансированными данными, включая методы балансировки с использованием LabelEncoder для категориальных признаков и StandardScaler для стандартизации признаков, позволила улучшить их производительность. Основное внимание уделяется анализу таких трудных для классификации атак, как «Metasploit_Brute_Force_SSH» и «NMAP_FIN_SCAN», где традиционные подходы могли бы привести к низкой точности. Использование современных методов обработки данных и машинного обучения позволило значительно повысить точность и эффективность классификации, что способствует более надежной защите от киберугроз в условиях сложных производственных систем.

Результаты

На рисунке 2 представлены все метрики для всех моделей. Из сравнения можно сделать вывод, что Random Forest доминирует по большинству показателей, а Logistic Regression имеет наименьшие значения. В алгоритме 1 представлен код для общего сравнительного анализа моделей.

```
plt.figure(figsize=(12, 6))
sns.barplot(data=df_melted, x='Metric', y='Value', hue='Model', palette="Set2")
plt.title('Overall Model Performance Comparison', fontsize=16)
plt.xlabel('Metrics', fontsize=12)
plt.ylabel('Values', fontsize=12)
plt.ylim(0.7, 1.02)
plt.legend(title='Models', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

Алгоритм 1 – Код для общего сравнительного анализа моделей

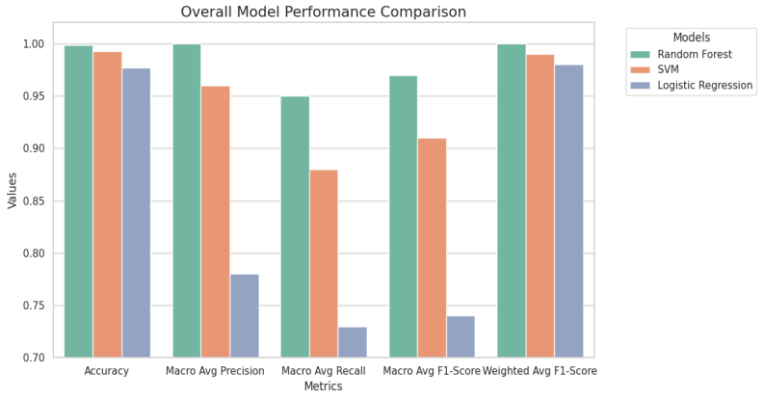

Рисунок 2 – Общий сравнительный анализ производительности моделей

Линейный график демонстрирует изменения значений метрик для каждой модели. В Алгоритме 2 представлен код для создания линейного графика. Основные различия заметны между Random Forest и Logistic Regression, особенно в метриках Recall и Precision (рис. 3).

Код, показанный в алгоритме 3, строит график динамики кибератак по продолжительности потока с использованием библиотек Seaborn и matplotlib. Для каждого типа атаки отображается линия, показывающая зависимость между продолжительностью потока и количеством пакетов. Визуализация результатов представлена на рисунке 4.

```
plt.figure(figsize=(10, 6))
for model in df['Model']:
    plt.plot(metrics, df[df['Model'] == model].iloc[0, 1:], marker='o', label=model)

plt.title('Line Graph of Model Performance', fontsize=16)
plt.xlabel('Metrics', fontsize=12)
plt.ylabel('Values', fontsize=12)
plt.ylim(0.7, 1.02)
plt.legend(title='Models')
plt.grid(True, linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

Алгоритм 2 – Код линейного графика

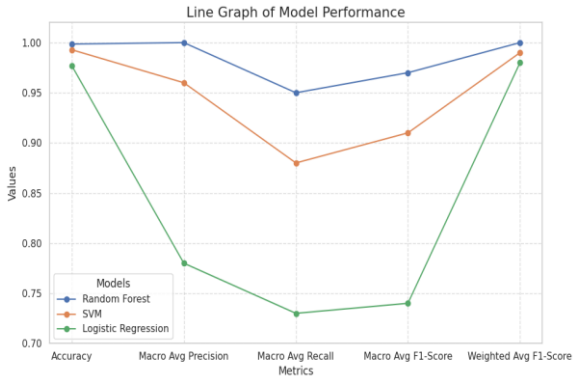

Рисунок 3 – Линейный график для каждой модели

```
import matplotlib.pyplot as plt
import seaborn as sns
# Use Seaborn style for better aesthetics
sns.set(style="whitegrid", palette="muted")
# Unique attack types
attack_types = data['Attack_type'].unique()
# Create a figure with specified size
plt.figure(figsize=(12, 6))
# Loop through all attack types
for attack in attack_types:
    attack_data = data[data['Attack_type'] == attack]
    # Plot for each attack type, set color for each attack
    plt.plot(attack_data['flow_duration'], label=attack, lw=2)
# Set axis labels
plt.xlabel('Flow Duration (seconds)', fontsize=14, fontweight='bold')
plt.ylabel('Packet Count', fontsize=14, fontweight='bold')
# Title of the graph
plt.title('Cyberattack Dynamics by Flow Duration', fontsize=16, fontweight='bold')
# Set up the legend
plt.legend(title='Attack Type', fontsize=12, title_fontsize='13', loc='upper right')
# Add grid for better readability
plt.grid(True, linestyle='--', alpha=0.7)
# Set axis tick label font size
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
# Improve layout
plt.tight_layout()
# Show the plot
plt.show()
```

Алгоритм 3 – Код для отображения динамики кибератак по продолжительности потока

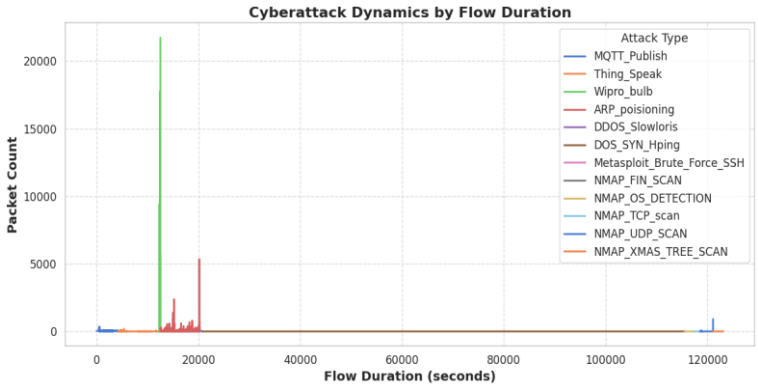

Рисунок 4 – Динамика кибератак по продолжительности потока

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Данные для тепловой карты
data = {
    'RandomForest': [0.9986, 0.99, 0.95, 0.97, 1.0],
    'SVM': [0.9929, 0.96, 0.88, 0.91, 0.99],
    'LogisticRegression': [0.9771, 0.78, 0.73, 0.74, 0.98]
}

df = pd.DataFrame(data, index=['Accuracy', 'Macro Avg Precision', 'Macro Avg Recall', 'Macro Avg F1-Score', 'Weighted Avg F1-Score'])

# Построение тепловой карты
plt.figure(figsize=(8, 6))
sns.heatmap(df, annot=True, cmap="coolwarm", fmt='.4f', cbar=True)
plt.title('Model Comparison Heatmap')
plt.ylabel('Metrics')
plt.xlabel('Models')
plt.tight_layout()
plt.show()
```

Алгоритм 4 – Код тепловой карты

Код, показанный в алгоритме 3, строит тепловую карту с показателями метрик для трех моделей машинного обучения: Random Forest, SVM и Logistic Regression. Каждая модель сравнивается по метрикам: Accuracy, Macro Avg Precision, Macro Avg Recall, Macro Avg F1-Score и Weighted Avg F1-Score (рис. 5).

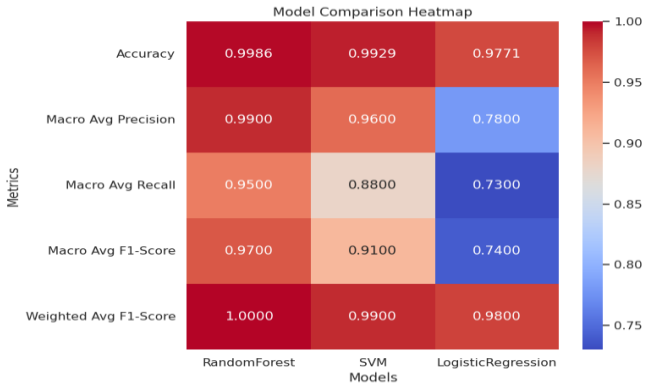

Рисунок 5 – Тепловая карта

Тепловая карта позволяет визуально сравнить эффективность моделей по различным метрикам. Модель Random Forest демонстрирует наивысшие значения во всех метриках, особенно по точности и взвешенному F1-скор. Модель SVM имеет хорошие результаты, но уступает Random Forest в некоторых показателях, таких как точность и макро-усредненная точность. Модель Logistic Regression показывает наименьшие значения по точности и полноте, что ограничивает её использование в более сложных задачах, таких как обнаружение кибератак в реальном времени для IIoT.

```python
import numpy as np
import matplotlib.pyplot as plt
metrics = ['Accuracy', 'Macro Avg Precision', 'Macro Avg Recall', 'Macro Avg F1-Score', 'Weighted Avg F1-Score']
models = ['RandomForest', 'SVM', 'LogisticRegression']
values_rf = [0.9986, 0.99, 0.95, 0.97, 1.0]
values_svm = [0.9929, 0.96, 0.88, 0.91, 0.99]
values_lr = [0.9771, 0.78, 0.73, 0.74, 0.98]
angles = np.linspace(0, 2 * np.pi, len(metrics), endpoint=False).tolist()
values_rf += values_rf[:1]
values_svm += values_svm[:1]
values_lr += values_lr[:1]
angles += angles[:1]
fig, ax = plt.subplots(figsize=(6, 6), dpi=100, subplot_kw=dict(polar=True))
ax.plot(angles, values_rf, color='blue', linewidth=2, linestyle='solid', label='RandomForest')
ax.fill(angles, values_rf, color='blue', alpha=0.25)
ax.plot(angles, values_svm, color='green', linewidth=2, linestyle='solid', label='SVM')
ax.fill(angles, values_svm, color='green', alpha=0.25)
ax.plot(angles, values_lr, color='red', linewidth=2, linestyle='solid', label='LogisticRegression')
ax.fill(angles, values_lr, color='red', alpha=0.25)
ax.set_yticklabels([])
ax.set_xticks(angles[:-1])
ax.set_xticklabels(metrics)
plt.title('Model Comparison (Radar Chart)')
ax.legend(loc='upper right', bbox_to_anchor=(1.2, 1.2))
plt.tight_layout()
plt.show()
```

Алгоритм 5 – Код радарной графики

На радарном графике, который построен с помощью кода, указанного в алгоритме 5, представлены метрики для трех моделей машинного обучения: RandomForest, SVM и LogisticRegression. Каждая модель оценивается по пяти метрикам: Accuracy, Macro Avg Precision, Macro Avg Recall, Macro Avg F1-Score и Weighted Avg F1-Score (рис. 6).

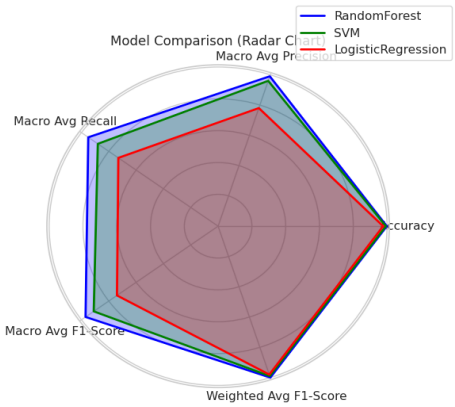

Рисунок 6 – Радарный график

Радарный график помогает наглядно сравнить производительность моделей по различным метрикам. Модель RandomForest имеет наивысшие значения по всем метрикам, особенно по точности и взвешенному F1-скор. SVMпоказывает хорошие результаты, но немного уступает RandomForest, особенно по макро-средней точности. LogisticRegression имеет наименьшие показатели, что ограничивает её применимость в задачах с высокими требованиями к точности, например, в области информационной безопасности для IIoT.Радарный график наглядно демонстрирует, что RandomForest является наиболее подходящей моделью для задач анализа кибератак и мошенничества в системе IIoT, так как она достигает наивысших показателей по ключевым метрикам. Модели SVM и LogisticRegression могут быть полезны, но их использование ограничено менее высокими результатами по точности и полноте.

```python
from sklearn.model_selection import learning_curve

def plot_learning_curve(estimator, X, y, title):
    train_sizes, train_scores, test_scores = learning_curve(
        estimator, X, y, cv=5, scoring='accuracy', n_jobs=-1, train_sizes=np.linspace(0.1, 1.0, 10)
    )
    train_mean = np.mean(train_scores, axis=1)
    test_mean = np.mean(test_scores, axis=1)

    plt.figure(figsize=(10, 6))
    plt.plot(train_sizes, train_mean, label='Точность на обучающей выборке', color='blue')
    plt.plot(train_sizes, test_mean, label='Точность на тестовой выборке', color='green')
    plt.title(title, fontsize=16)
    plt.xlabel('Размер обучающей выборки', fontsize=12)
    plt.ylabel('Точность', fontsize=12)
    plt.legend(loc='best')
    plt.grid()
    plt.show()

# Пример вызова
plot_learning_curve(RandomForestClassifier(), X_train, y_train, 'Кривая обучения для Random Forest')
```

Алгоритм 6 – Код кривой обучения для Random Forest

График кривой обучения иллюстрирует изменение точности модели в зависимости от размера обучающей выборки (рис. 7). Код кривой обучения в алгоритме 6 указан. На горизонтальной оси представлено количество данных в обучающей выборке, а на вертикальной оси – точность модели. Синяя линия показывает точность модели на обучающей выборке, а зеленая линия – на тестовой выборке. Увеличение размера обучающей выборки обычно приводит к улучшению точности модели, однако важно учитывать, насколько хорошо модель обобщается на новых данных (тестовых). Если две линии близки друг к другу, это свидетельствует о хорошем обобщении модели, минимизирующем риск переобучения. В противном случае значительное различие между обучающей и тестовой точностью может указывать на переобучение или недообучение.

Рисунок 7 – Кривая обучения

График распределения типов атак отображает количество каждой категории атак в набранных данных. Код указан в алгоритме 7. Горизонтальная ось указывает на количество записей для каждого типа атаки, а вертикальная ось показывает различные типы атак в порядке убывания их частоты. Использованная палитра 'viridis' помогает визуально выделить различия в количестве каждой категории атак. Такой график предоставляет понимание распределения типов атак и может использоваться для дальнейшего анализа, выявления редких или доминирующих угроз (рис. 8).

```
plt.figure(figsize=(10, 6))
sns.countplot(y=data['Attack_type'], order=data['Attack_type'].value_counts().index, palette='viridis')
plt.title('Распределение типов атак', fontsize=16)
plt.xlabel('Количество', fontsize=12)
plt.ylabel('Тип атаки', fontsize=12)
plt.show()
```

Алгоритм 7 – Код для распределение типов атак

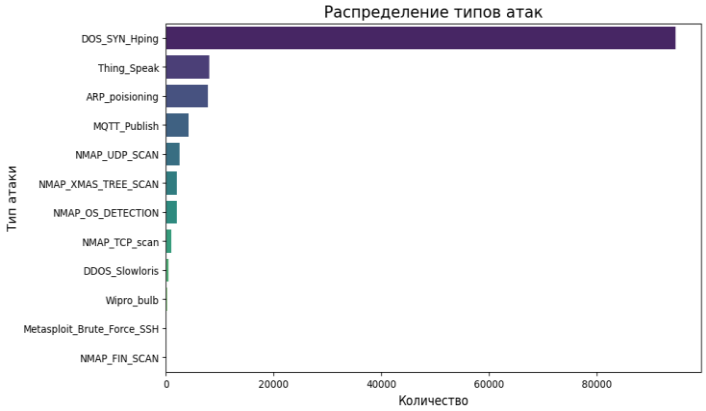

Рисунок 8 – Распределение типов атак

В алгоритмах 8, 9 и 10 указаны коды различных моделей, таких как Random Forest, SVM и Logistic Regression. Каждая модель настроена с различными параметрами для тренировки и оценки её качества. На рисунках 9, 10 11 представлены отчеты классификации соответствующих моделей. В каждом отчете указаны метрики, такие как Accuracy, Precision, Recall, F1 Score и Support. Эти метрики помогают анализировать результаты классификации и определить, какая модель более эффективна для конкретной задачи.Сравнивая отчеты, можно увидеть, какая модель демонстрирует наилучшие результаты в различных метриках и предоставляет наиболее точное предсказание.

```
from sklearn.ensemble import RandomForestClassifier
# Модель RandomForest
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)
rf_predictions = rf_model.predict(X_test)
print("\nОбучение модели: RandomForest")
print(f"Точность модели RandomForest: {accuracy_score(y_test, rf_predictions):.4f}")
print("Отчет о классификации:")
print(classification_report(y_test, rf_predictions))
```

Алгоритм 8 – Код модели Random Forest

```
Обучение модели: RandomForest
Точность модели RandomForest: 0.9986
Отчет о классификации:
                         precision    recall  f1-score   support

          ARP_poisioning       0.99      1.00      0.99      1578
           DDOS_Slowloris      0.99      0.99      0.99       100
            DOS_SYN_Hping      1.00      1.00      1.00     18897
             MQTT_Publish      1.00      1.00      1.00       871
Metasploit_Brute_Force_SSH     1.00      0.83      0.91         6
          NMAP_FIN_SCAN      1.00      0.67      0.80         3
       NMAP_OS_DETECTION     1.00      1.00      1.00       393
          NMAP_TCP_scan      1.00      1.00      1.00       220
          NMAP_UDP_SCAN      0.99      0.99      0.99       489
     NMAP_XMAS_TREE_SCAN     1.00      0.99      1.00       384
           Thing_Speak       1.00      0.99      0.99      1625
            Wipro_bulb       1.00      0.95      0.97        58

              accuracy                           1.00     24624
             macro avg       1.00      0.95      0.97     24624
          weighted avg       1.00      1.00      1.00     24624
```

Рисунок 9 – Отчет классификации модели Random Forest

```
# Модель SVM
svm_model = SVC(random_state=42)
svm_model.fit(X_train, y_train)
svm_predictions = svm_model.predict(X_test)
print("\nОбучение модели: SVM")
print(f"Точность модели SVM: {accuracy_score(y_test, svm_predictions):.4f}")
print("Отчет о классификации:")
print(classification_report(y_test, svm_predictions))
```

Алгоритм 9 – Код модели SVM

```
Обучение модели: SVM
Точность модели SVM: 0.9929
Отчет о классификации:
                          precision    recall  f1-score   support

            ARP_poisioning       0.93      0.98      0.96      1578
              DDOS_Slowloris       1.00      0.71      0.83       100
                 DOS_SYN_Hping       1.00      1.00      1.00     18897
                  MQTT_Publish       1.00      0.99      1.00       871
Metasploit_Brute_Force_SSH       1.00      0.67      0.80         6
                NMAP_FIN_SCAN       0.67      0.67      0.67         3
          NMAP_OS_DETECTION       1.00      1.00      1.00       393
              NMAP_TCP_scan       1.00      1.00      1.00       220
              NMAP_UDP_SCAN       0.95      0.97      0.96       489
         NMAP_XMAS_TREE_SCAN       1.00      0.99      1.00       384
                Thing_Speak       0.98      0.95      0.97      1625
                 Wipro_bulb       1.00      0.62      0.77        58

                   accuracy                           0.99     24624
                  macro avg       0.96      0.88      0.91     24624
               weighted avg       0.99      0.99      0.99     24624
```

Рисунок 10 – Отчет классификации модели SVM

```python
# Обучение модели LogisticRegression
logistic_model = LogisticRegression(max_iter=500, solver='saga', random_state=42)
logistic_model.fit(X_train, y_train)
# Предсказания
y_pred_logistic = logistic_model.predict(X_test)

# Оценка модели
accuracy_logistic = accuracy_score(y_test, y_pred_logistic)
print(f"Точность модели LogisticRegression: {accuracy_logistic:.4f}")
print("Отчет о классификации:")
print(classification_report(y_test, y_pred_logistic, zero_division=0))
```

Алгоритм 10 – Код модели Logistic Regression

```
Точность модели LogisticRegression: 0.9771
Отчет о классификации:
                          precision    recall  f1-score   support

            ARP_poisioning       0.87      0.80      0.83      1578
              DDOS_Slowloris       0.72      0.71      0.71       100
                 DOS_SYN_Hping       1.00      1.00      1.00     18897
                  MQTT_Publish       0.99      1.00      1.00       871
Metasploit_Brute_Force_SSH       0.00      0.00      0.00         6
                NMAP_FIN_SCAN       0.00      0.00      0.00         3
          NMAP_OS_DETECTION       0.97      1.00      0.98       393
              NMAP_TCP_scan       1.00      1.00      1.00       220
              NMAP_UDP_SCAN       0.93      0.91      0.92       489
         NMAP_XMAS_TREE_SCAN       1.00      0.99      1.00       384
                Thing_Speak       0.84      0.92      0.88      1625
                 Wipro_bulb       1.00      0.43      0.60        58

                   accuracy                           0.98     24624
                  macro avg       0.78      0.73      0.74     24624
               weighted avg       0.98      0.98      0.98     24624
```

Рисунок 11 – Отчет классификации модели Logistic Regression

Заключение

Для анализа кибератак использовались три модели машинного обучения: Logistic Regression, SVM и Random Forest. Каждая из них продемонстрировала высокую точность в классификации атак, но с определенными особенностями и ограничениями.

1. Logistic Regression показала точность 97,71%, однако точность для отдельных классов, таких как «Metasploit_Brute_Force_SSH» и «NMAP_FIN_SCAN», была ниже из-за трудностей с классификацией редких случаев. Модель показала удовлетворительные результаты в большинстве категорий, но ее точность может снижаться при работе с малоизученными аномалиями.

2. SVM продемонстрировала значительно высокую точность – 99,29%. Эта модель хорошо справилась с большинством классов, включая сложные случаи, такие как «ARP_poisioning» и «DOS_SYN_Hping». Высокие значения метрик свидетельствуют о надежности SVM в обнаружении аномалий и уязвимостей.

3. Random Forest продемонстрировала наивысшую точность – 99,86%. Модель достигла превосходных результатов по всем показателям, включая precision, recall и F1-score. Высокая стабильность Random Forest позволила эффективно классифицировать атаки, даже в случаях с ограниченными выборками, такими как «Metasploit_Brute_Force_SSH».

Таким образом, Random Forest показала наилучшие результаты по точности и устойчивости.

В будущем предполагается дальнейшее развитие моделей машинного обучения, включая применение глубокого обучения и усовершенствование методов для улучшения классификации и обнаружения киберугроз в системах Industry 4.0.

Список литературы

1. Rule-Based With Machine Learning IDS for DDoS Attack Detection in Cyber-Physical Production Systems (CPPS) / A. Hussain et al // IEEE Access. – 2024. – vol. 12. – P. 3445261. https://doi.org/10.1109/ACCESS.2024.3445261.

2. An Efficient Hybrid-DNN for DDoS Detection and Classification in Software-Defined IIoT Networks / A. Zainudin et al // IEEE Internet of Things Journal. – 2023. – vol. 10, № 10.
3. A Hybrid Machine-Learning Ensemble for Anomaly Detection in Real-Time Industry 4.0 Systems / D. Velásquez et al // IEEE Access. – 2022. – vol. 10. – P. 3188102. https://doi.org/10.1109/ACCESS.2022.3188102.
4. Toward a Human-Cyber-Physical System for Real-Time Anomaly Detection / B. Bajic et al // IEEE Systems Journal. – 2024. – vol. 18, № 2.
5. Mienye I.D. Deep Learning for Credit Card Fraud Detection: A Review of Algorithms, Challenges, and Solutions / I.D. Mienye, N. Jere // IEEE Access. – 2024. – vol. 12. – P. 3426955. https://doi.org/10.1109/ACCESS.2024.3426955.
6. Machine and Deep Learning for Digital Twin Networks: A Survey / B. Qin et al // IEEE Internet of Things Journal. – 2024. – vol. 11, № 21.
7. Cybersecurity Threats and Mitigation Measures in Agriculture 4.0 and 5.0 / C. Maraveas et al // Smart Agricultural Technology. – 2024. – vol. 9. https://doi.org/10.1016/j.atech.2024.100616.
8. Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms / F.K. Alarfaj et al // IEEE Access. – 2022. – vol. 10. – P. 3166891. https://doi.org/10.1109/ACCESS.2022.3166891.
9. A Fraud Detection System in Financial Networks Using AntiBenford Subgraphs and Machine Learning Algorithms / R.K. Somkunwar et al // in 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE). – 2023. https://doi.org/10.1109/AIKIIE60097.2023.1039032.
10. Monamo P. Unsupervised Learning for Robust Bitcoin Fraud Detection / P. Monamo, V. Marivate, B. Twala // in 2016 IEEE Conference. – 2016. https://doi.org/10.1109/XXXX.2016.XXXXXXX.
11. Chatterjee J. Digital Trust in Industry 4.0 & 5.0: Impact of Frauds / J. Chatterjee, M. Damle, A. Aslekar // in Proceedings of the 7th International Conference on Intelligent Computing and Control Systems (ICICCS-2023). – 2023. https://doi.org/10.1109/ICICCS56967.2023.10142925.
12. DIAVA: A Traffic-Based Framework for Detection of SQL Injection Attacks and Vulnerability Analysis of Leaked Data / H. Gu et al // IEEE Transactions on Reliability. – 2020. – vol. 69, № 1.
13. Mienye I.D. A Deep Learning Ensemble With Data Resampling for Credit Card Fraud Detection / I.D. Mienye, Y. Sun, // IEEE Access. – 2023. – vol. 11. – P. 3262020. https://doi.org/10.1109/ACCESS.2023.3262020.
14. Cyber Threat Predictive Analytics for Improving Cyber Supply Chain Security / Yeboah-Ofori A.Y.O.F.et al // IEEE Access. – 2021. – vol. 9. – P. 3087109. https://doi.org/10.1109/ACCESS.2021.3087109.
15. Enhancing DDoS Attack Detection and Mitigation in SDN Using an Ensemble Online Machine Learning Model / A.A. Alashhab et al // IEEE Access. – 2024. – vol. 12. P. 3384398. https://doi.org/10.1109/ACCESS.2024.3384398.

С. Адилжанова, М. Кунелбаев, Д. Сыбанова*
әл-Фараби атындағы Қазақ ұлттық университеті,
050040 Қазақстан Республикасы, Алматы қ., әл-Фараби даңғылы, 71
*e-mail:dsybanovaa@gmail.com

КИБЕРШАБУЫЛДАРДЫ ТАЛДАУ ҮШІН МАШИНАЛЫҚ ОҚЫТУДЫ ҚОЛДАНУ: RT-IGOR DATASET НЕГІЗІНДЕГІ ЗЕРТТЕУ 2022

Мақала кибершабуылдарды талдау үшін машиналық оқытудың қолданылуын зерттеуге арналған. Зерттеу аномалияларды анықтау және жалған позитивтерді азайту тапсырмаларын сәтті орындайтын Random Forest, SVM және Logistic Regression алгоритмдерін қарастырады. Модельдерді категориялық белгілер үшін LabelEncoder және деректерді стандарттау үшін StandardScaler пайдалану сияқты теңгерімсіз деректермен жұмыс істеуге бейімдеу олардың өнімділігін айтарлықтай жақсартуға мүмкіндік берді. «Real-time Internet of Things (RT-IoT 2022)» жиынтығындағы деректерді талдау негізінде модельдердің дәлдігі мен тұрақтылығының нәтижелері ұсынылған. Негізгі назар киберқауіптерден, соның ішінде ақпараттың ағып кетуінен, DDoS шабуылдарынан және басқа қауіп түрлерінен қорғауға бағытталған. Кибершабуылдарды зерттеуге арналған әртүрлі Машиналық оқыту алгоритмдерін талдау айтарлықтай нәтиже көрсетті. Random Forest қауіптердің әртүрлі түрлерін жіктеуде жоғары тұрақтылық пен тиімділікті қамтамасыз ете отырып, ең жоғары дәлдікті – 99,86% көрсетті. SVM көптеген күрделі

сыныптарды басқара отырып, 99,29% дәлдікті көрсетті. Logistic Regression қанағаттанарлық нәтижелерді 97,71% дәлдікпен көрсетті, дегенмен кейбір сирек жағдайларда дәлдік төмен болды. Осылайша, Random Forest және SVM жоғары дәлдік пен сенімділікті қамтамасыз ете отырып, Цифрлық жүйелердегі қауіпсіздік тапсырмалары мен кибершабуылдарды талдау үшін ең жақсы тиімділікті көрсетті. Болашақта қауіптерді дәлірек анықтау және талдау үшін терең оқыту сияқты күрделі әдістерді енгізу жоспарлануда.

Түйін сөздер: *Кибершабуыл,машиналық оқыту алгоритмдері,кездейсоқ орман, SVM, логистикалық регрессия, RT-IoT 2022, шабуылдарды анықтау.*

S. Adilzhanova, M. Kunelbayev, D. Sybanova*
Al-Farabi Kazakh National University,
050040 Republic of Kazakhstan, Almaty, al-Farabi Ave., 71
*e-mail:dsybanovaa@gmail.com

THE USE OF MACHINE LEARNING TO ANALYZE CYBER ATTACKS: A STUDY BASED ON THE RT-IGOR 2022 DATASET

The article is devoted to the study of the use of machine learning for the analysis of cyber attacks. The study examines Random Forest, SVM and Logistic Regression algorithms, which successfully cope with the tasks of detecting anomalies and minimizing false positives. Adapting models to work with unbalanced data, such as using LabelEncoder for categorical features and StandardScaler for data standardization, has significantly improved their performance. Based on the analysis of data from the «Real-Time Internet of Things (RT-IoT 2022)» set, the results of the accuracy and stability of the models are presented. The main focus is on protecting against cyber threats, including information leaks, DDoS attacks, and other types of threats. An analysis of various machine learning algorithms for cyberattack research has shown significant results. Random Forest has demonstrated the highest accuracy – 99,86%, providing high stability and efficiency in classifying various types of threats. SVM showed an accuracy of 99,29%, coping with most complex classes. Logistic Regression showed satisfactory results with an accuracy of 97,71%, although in some rare cases the accuracy was lower. Thus, Random Forest and SVM have demonstrated the best performance for security and cyberattack analysis tasks in digital systems, providing high accuracy and reliability. In the future, it is planned to introduce more sophisticated methods, such as deep learning, to more accurately identify and analyze threats.

Key words: *Cyberattack, machine learning algorithms, Random Forest, SVM, Logistic Regression, RT-IoT 2022, attack detection.*

Сведения об авторах

Салтанат Альмуханбетовна Адилжанова – доктор технических наук, преподаватель кафедры криптологии и кибербезопасности факультета информационных технологий; Казахский национальный университет имени аль-Фараби, Республика Казахстан; e-mail: asaltanat81@gmail.com. ORCID: https://orcid.org/0000-0003-1768-064X.

Мурат Кунелбаев – старший научный сотрудник Института информационных и вычислительных технологий Министерства науки и высшего образования Республики Казахстан; e-mail: murat7508@yandex.kz. ORCID: https://orcid.org/0000-0002-5648-4476.

Дана Даулетовна Сыбанова* – аналитик в сфере информационной безопасности, магистрантка кафедры криптологии и кибербезопасности факультета информационных технологий; Казахский национальный университет имени аль-Фараби, Республика Казахстан; e-mail: dsybanovaa@gmail.com. ORCID: https://orcid.org/0009-0003-4684-5586.

Авторлар туралы мәліметтер

Салтанат Альмуханбетовна Адилжанова – техника ғылымдарының докторы, ақпараттық технологиялар факультетінің криптология және киберқауіпсіздік кафедрасының оқытушысы; Әл-Фараби атындағы Қазақ ұлттық университеті, Қазақстан Республикасы; e-mail: asaltanat81@gmail.com. ORCID: https://orcid.org/0000-0003-1768-064X.

Мурат Кунелбаев – Қазақстан Республикасы Ғылым және жоғары білім министрлігіне қарасты Ақпараттық және есептеу технологиялары институтының аға ғылыми қызметкері; e-mail: murat7508@yandex.kz. ORCID: https://orcid.org/0000-0002-5648-4476.

Дана Даулетовна Сыбанова* – ақпараттық қауіпсіздік саласының аналитигі, ақпараттық технологиялар факультетінің криптология және киберқауіпсіздік кафедрасының магистранты; Әл-Фараби атындағы Қазақ ұлттық университеті, Қазақстан Республикасы; e-mail: dsybanovaa@gmail.com. ORCID: https://orcid.org/0009-0003-4684-558

Information about the authors

Saltanat Adilzhanova – doctor of technical sciences, lecturer at the department of Cryptology and Cybersecurity of the Faculty of Information Technology; Al-Farabi Kazakh National University, Republic of Kazakhstan; e-mail: asaltanat81@gmail.com. ORCID: https://orcid.org/0000-0003-1768-064X.

Murat Kunelbayev – senior research fellow at the Institute of Information and Computational Technologies of the Ministry of Science and Higher Education of the Republic of Kazakhstan; e-mail: murat7508@yandex.kz. ORCID: https://orcid.org/0000-0002-5648-4476.

Dana Sybanova* – cybersecurity analyst, master's student at the department of Cryptology and Cybersecurity of the Faculty of Information Technology; Al-Farabi Kazakh National University, Republic of Kazakhstan; e-mail: dsybanovaa@gmail.com. ORCID: https://orcid.org/0009-0003-4684-5586.

Поступила в редакцию 04.02.2025
Поступила после доработки 05.04.2025
Принята к публикации 09.04.2025

https://doi.org/10.53360/2788-7995-2025-2(18)-3

IRSTI: 20.19.27

A.M. Amantay*, Zh.M. Makhambetali
Kazakh-British Technical University,
050000, Republic of Kazakhstan, Almaty, Tole bi street, 59
*e-mail: ai_amantay@kbtu.kz

CLUSTERING AND CLASSIFICATION OF DISEASES USING STOCHASTIC DYNAMIC OPTIMIZATION

Abstract: *This study presents a new approach to the optimization of Natural Language Processing (NLP) techniques for medical entity recognition and disease classification. By leveraging patient queries and PubMed article abstracts, the research uses advanced extraction methods to identify biomedical entities and diseases from medical texts. Diseases are grouped using a combination of TF-IDF and K-means clustering, and classification models are then applied to predict disease clusters based on known entities. A key innovation of this work is the use of Stochastic Dynamic Optimization to fine-tune parameters, significantly enhancing clustering and classification performance.*

Experimental results demonstrate that the proposed method improves the accuracy of extraction and classification, outperforming traditional methods in terms of precision and scalability. This scalable and efficient approach to biomedical data analysis has the potential to support future clinical decision-making, enable personalized medicine, and provide valuable healthcare insights, ultimately contributing to improved patient outcomes and more effective research workflows.

Key words: *Machine Learning, Stochastic Dynamic Optimization, Disease clustering, PubMed abstracts, Medical Entity Recognition, Data Extraction, Healthcare data optimization.*

Introduction

The rapid expansion of biomedical literature and the increasing volume of patient-generated data present significant challenges for healthcare systems. Understanding unstructured medical texts from the growing medical information becomes an inefficient task for traditional data analysis methods, which struggle to keep up with expansion rates [1, 2]. The fundamental technology Named Entity Recognition (NER) encounters difficulties when processing medical entities because medical language proves complex along with its domain-specific jargon and information presentation irregularities across medical sources. Medical institutions require modern techniques that produce scalable evaluations using efficient methods with high accuracy to analyze their extensive healthcare databases. Current methods fail to build interconnected frameworks, which unite patient-dependent inquiries with medical contents to provide meaningful results while being easy to interpret. The existing gap prevents healthcare providers from using personalized and efficient decision-making approaches during clinical situations.

Our investigation establishes a new medical entity detection system with disease classification capabilities through the integration of modern extraction tools alongside clustering algorithms and operation optimization. The alignment between patient queries and disease clusters creates more precise disease predictions that professionals can easily understand through