УДК 004.832:004.056

Вестник УрФО № 4(58) / 2025, с. 5–13

Кушнир Н. В., Власенко А. В., Булгаков В. В., Акулинин Г. А.

10.14529/secur250401

МАШИННОЕ ОБУЧЕНИЕ И ЭВРИСТИЧЕСКИЕ МЕТОДЫ В ЗАДАЧАХ КЛАССИФИКАЦИИ ТЕКСТОВ ДЛЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Статья посвящена сравнительному анализу эвристических и обучаемых методов в задачах классификации текстов, рассматриваемых в контексте информационной безопасности. В качестве прикладного примера выбрана задача фильтрации спам-сообщений, позволяющая продемонстрировать особенности, преимущества и ограничения каждого подхода. Описана реализация двух систем: эвристической, основанной на наборе детерминированных правил, и обучаемой модели, использующей алгоритмы машинного обучения и предварительную обработку текстов на естественном языке.

Далее проведено экспериментальное сравнение качества классификации, скорости обработки и гибкости адаптации обеих систем. Особое внимание уделено критериям точности, полноты, F1-меры, а также возможностям автономного функционирования и расширяемости.

В заключении на основе полученных результатов формулируются выводы о целесообразности применения каждого из подходов в зависимости от условий эксплуатации и требований к безопасности.

Ключевые слова: *классификация текстов, эвристические методы, машинное обучение, информационная безопасность, спам-фильтрация, бинарная классификация, обработка естественного языка.*

MACHINE LEARNING AND HEURISTIC METHODS IN TEXT CLASSIFICATION TASKS FOR INFORMATION SECURITY

The article is devoted to a comparative analysis of heuristic and trainable methods in the tasks of text classification considered in the context of information security. The task of filtering spam messages is chosen as an applied example, which makes it possible to demonstrate the features, advantages and limitations of each approach. The implementation of two systems is described: a heuristic one based on a set of deterministic rules, and a trainable model using machine learning algorithms and preprocessing of natural language texts.

Next, an experimental comparison was made of the classification quality, processing speed, and adaptability of both systems. Special attention is paid to the criteria of accuracy, completeness, F1-measure, as well as the possibilities of autonomous operation and extensibility.

In conclusion, based on the results obtained, conclusions are drawn about the expediency of using each of the approaches, depending on the operating conditions and safety requirements.

Keywords: *text classification, heuristic methods, machine learning, information security, spam filtering, binary classification, natural language processing.*

Введение

Информационная безопасность в современном цифровом мире становится неотъемлемой частью стабильного функционирования частных и государственных информационных систем. Одной из ключевых задач в этой области является эффективная и быстрая обработка и классификация текстовых данных, среди которых могут находиться потенциально опасные сообщения. Тексты, содержащие признаки фишинга, мошенничества или деструктивного контента, могут угрожать конфиденциальности, целостности и доступности данных, особенно при массовом распространении, как это происходит, например, с текстовыми сообщениями. Традиционные эвристические методы фильтрации и классификации текстов, основанные на фиксированных правилах, обладают высокой интерпретируемостью и простотой настройки, но зачастую не справляются с постоянно эволюционирующими типами угроз. В то же время методы машинного обучения демонстрируют высокую адаптивность и точность за счёт способности выявлять скрытые закономерности в больших объёмах текстов. Однако их применение требует качественно размеченных данных, вычислительных ресурсов и дополнительной инфраструктуры. В условиях стремительного роста объёмов цифровых сообщений и увеличения количества сложных атак особенно актуальным становится вопрос выбора подходящего инструмента для автоматической классификации текстов в рамках задач информационной безопасности. Сравнительный анализ эвристических и обучаемых подходов позволяет не только оценить их эффективность в прикладных задачах, но и сформировать понимание о целесообразности их использования в различных контекстах — от автономных систем до крупных защищённых сетей. Таким образом, изучение различий между этими подходами становится важным этапом в построении более устойчивых и интеллектуальных систем защиты информации.

Применение эвристических методов в системах защиты

Эвристические методы классификации текстов, особенно в контексте информационной безопасности, представляют собой совокупность правил и эвристик, разработанных вручную и основанных на анализе признаков, характерных для вредоносных или нежелательных сообщений. Наиболее известным

примером является система SpamAssassin, в основе которой лежит обширный набор регулярных выражений, логических правил и сигнатур, способных обнаруживать шаблоны спама. Подобные подходы широко применяются в почтовых клиентах Outlook, Mozilla Thunderbird и фильтрующих прокси-серверах вроде Squid и WebSense.

Эвристические правила также активно применяются в антифишинговых механизмах: анализируется домен отправителя, наличие в тексте подозрительных фраз, частота появления ссылок и вложений. Специальные правила позволяют выявлять сообщения, содержащие ключевые слова вроде "вы выиграли", "переведите деньги", "действуйте сейчас" и т.д. Преимущество данных методов заключается в их простоте, быстродействии и полной автономности, что делает их пригодными для систем с ограниченными вычислительными ресурсами.

Однако у эвристических решений есть существенные недостатки. В первую очередь, это низкая гибкость: при появлении новых методов обхода фильтров злоумышленниками, правила требуют ручного обновления. Кроме того, они демонстрируют невысокую точность при работе с неоднозначными или грамматически искажёнными текстами.

Машинное обучение в задачах классификации и защиты

Методы машинного обучения (МО) предлагают альтернативу ручному проектированию правил. Они способны выявлять скрытые закономерности в больших объёмах данных, основываясь на примерах помеченных сообщений. Такие подходы стали особенно популярны в задачах классификации спама, фишинга и обнаружения вредоносных вложений [1].

Один из наиболее известных примеров — фильтрация спама в Gmail, где используются модели, обученные на огромном количестве пользовательских писем. Подходы включают классические алгоритмы, такие как наивный байесовский классификатор, который определяет спам на основе вероятностей слов; логистическая регрессия, метод статистической классификации для разделения спама и не спама; и SVM, или машина опорных векторов, которая находит оптимальную границу для разделения данных. Также применяются современные нейросетевые модели, включая BERT, модель на основе трансформеров, глубоко понимающая контекст текста, и DistilBERT, более легкая и быстрая версия BERT, оптимизированная для задач обработки естественного языка [2] [3].

Применение МО также эффективно в системах обнаружения фишинга: модели анализируют как текст письма, так и характеристики ссылок и вложений. В других случаях — при анализе логов систем безопасности, журналов событий, сетевого трафика — машинное обучение позволяет выявлять аномалии и потенциальные угрозы без необходимости заранее формализовать правила [4] [5] [6].

Отдельно стоит отметить тренд на гибридные подходы, в которых эвристические фильтры используются как предварительный барьер, а более ресурсоёмкие МО-модели подключаются на следующих этапах анализа.

Постановка задач исследования

Цель настоящего исследования — провести сравнительный анализ эффективности эвристического и обучаемого подходов к классификации текстов в задачах информационной безопасности. В качестве примера рассматривается проблема автоматической фильтрации текстовых сообщений.

Для достижения цели поставлены следующие задачи:

1. Разработать систему фильтрации сообщений на основе набора эвристических правил;
2. Реализовать систему машинного обучения для классификации сообщений на основе размеченного корпуса данных;
3. Провести экспериментальное сравнение решений по метрикам точности, полноты, F1-меры и скорости обработки;
4. Оценить возможности масштабирования, автономности и адаптивности обоих подходов.

Система на основе эвристических правил

Фильтрация реализована в виде Python-скрипта, обрабатывающего текстовые SMS-сообщения. Для каждого сообщения применяются эвристические правила, где каждое правило добавляет по одному баллу к итоговой оценке.

Примеры правил:

- Наличие ключевых слов: "бесплатно", "подарок", "срочно";
- Более 2 ссылок в сообщении;
- Высокая доля заглавных букв;
- Подозрительная длина сообщения (менее 20 или более 160 символов);
- Высокая частота специальных символов.

Сообщение считается спамом при достижении порогового значения (например, три балла). Порог можно настроить вручную. Для каждого сообщения формируется отчёт с указанием, какие именно правила сработали.

Листинг 1. Эвристический классификатор сообщений

```
import re
from typing import List, Dict

# Настраиваемые параметры
SPAM_KEYWORDS = [
    'бесплатно', 'выиграть', 'срочно', 'акция', 'купи',
    'free', 'win', 'urgent', 'act now', 'buy now'
]
MAX_LINKS = 2
UPPERCASE_RATIO_THRESHOLD = 0.5
SPECIAL_CHAR_THRESHOLD = 10
SPAM_SCORE_THRESHOLD = 3

def extract_links(text: str) -> List[str]:
    return re.findall(r'https?://\S+', text)

def get_uppercase_ratio(text: str) -> float:
    if not text:
        return 0
    uppercase_chars = sum(1 for c in text if c.isupper())
    return uppercase_chars / len(text)

def get_special_char_count(text: str) -> int:
    return sum(1 for c in text if not c.isalnum() and not c.isspace())

def classify_sms(text: str) -> Dict:
    score = 0
    reasons = []

    text_lower = text.lower()

    # 1. Ключевые слова
    for keyword in SPAM_KEYWORDS:
        if keyword in text_lower:
            score += 1
            reasons.append(f"Ключевое слово: '{keyword}'")
            break

    # 2. Ссылки
    links = extract_links(text)
    if len(links) > MAX_LINKS:
        score += 1
        reasons.append(f"Слишком много ссылок: {len(links)}")

    # 3. Заглавные буквы
    uppercase_ratio = get_uppercase_ratio(text)
    if uppercase_ratio > UPPERCASE_RATIO_THRESHOLD:
        score += 1
        reasons.append(f"Высокая доля ЗАГЛАВНЫХ букв: {uppercase_
ratio:.2f}")

    # 4. Спецсимволы
    special_chars = get_special_char_count(text)
    if special_chars > SPECIAL_CHAR_THRESHOLD:
```

```
            score += 1
            reasons.append(f"Много специальных символов: {special_chars}")

        # 5. Длина сообщения
        if len(text) < 20 or len(text) > 160:
            score += 1
            reasons.append(f"Подозрительная длина сообщения: {len(text)}
символов")

        is_spam = score >= SPAM_SCORE_THRESHOLD

        return {
            'result': 'SPAM' if is_spam else 'NOT SPAM',
            'score': score,
            'reasons': reasons
        }

    # Пример использования
    if __name__ == "__main__":
        test_sms = "АКЦИЯ! Выиграй миллион! https://spam.link"
        result = classify_sms(test_sms)
        print("Результат:", result['result'])
        print("Сработавшие правила:")
        for reason in result['reasons']:
            print("-", reason)
```

Система на основе машинного обучения

В качестве датасета используется SMS Spam Collection Dataset, содержащий 5572 текстовых SMS-сообщений, размеченных как спам (747 сообщений) или не спам (4825 сообщений). Датасет представляет собой общедоступный набор данных, часто используемый для задач классификации текстов. Предварительная обработка включает:

- Очистку HTML-тегов и специальных символов;
- Приведение текста к нижнему регистру;
- Удаление стоп-слов с использованием библиотеки NLTK;
- Токенизацию и векторизацию текста с применением TF-IDF.

Для классификации рассматриваются модели:

- Наивный байесовский классификатор (MultinomialNB);
- Логистическая регрессия (Logistic Regression);
- Метод опорных векторов (SVM).

Обучение производится на 80% выборки, 20% оставляется для тестирования. Модель сохраняется через joblib и может дообучаться на новых пользовательских данных.

Для оценки используются метрики:

- Accuracy (доля верно классифицированных сообщений);
- Precision и Recall;
- F1-мера как гармоническое среднее.

Также выводится вероятность классификации для каждого сообщения, что повышает доверие пользователя к системе.

Скорость обработки одного сообщения не превышает 0.2 секунды при использовании TF-IDF и логистической регрессии. Система работает в автономном режиме и поддерживает обновление модели через загрузку локального файла с новыми данными.

Листинг 2. Классификатор сообщений на основе машинного обучения

```
import pandas as pd
import numpy as np
import re
import joblib
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split, cross_validate
from sklearn.metrics import classification_report
from sklearn.pipeline import Pipeline
```

```
    from sklearn.base import BaseEstimator, TransformerMixin
    from nltk.corpus import stopwords
    from nltk.stem.snowball import SnowballStemmer
    import nltk
    import os

    nltk.download('stopwords')

    # --- Кастомный препроцессор ---
    class TextCleaner(BaseEstimator, TransformerMixin):
        def __init__(self, language='english'):
            self.language = language
            self.stop_words = stopwords.words(language)
            self.stemmer = SnowballStemmer(language)

        def clean(self, text):
            text = re.sub(r'<[^>]+>', ' ', text)  # Remove HTML tags
            text = re.sub(r'http\S+|www\S+', '', text)  # Remove URLs
            text = re.sub(r'[^a-zA-Za-яА-Я\s]', ' ', text)  # Keep only
letters
            text = text.lower()
            tokens = text.split()
            tokens = [t for t in tokens if t not in self.stop_words]
            tokens = [self.stemmer.stem(t) for t in tokens]
            return ' '.join(tokens)

        def fit(self, X, y=None):
            return self

        def transform(self, X):
            return [self.clean(text) for text in X]

    # --- Загрузка и обучение модели ---
    def train_model(dataset_path='spam.csv', language='english'):
        df = pd.read_csv(dataset_path, encoding='latin-1')[['v1', 'v2']]
        df.columns = ['label', 'text']
        df['label'] = df['label'].map({'ham': 0, 'spam': 1})

        X, y = df['text'], df['label']

        pipeline = Pipeline([
            ('cleaner', TextCleaner(language=language)),
            ('tfidf', TfidfVectorizer(max_features=3000)),
            ('classifier', LogisticRegression())
        ])

        # 5-кратная кросс-валидация
        scoring = ['accuracy', 'precision', 'recall', 'f1']
        scores = cross_validate(pipeline, X, y, cv=5, scoring=scoring)
        print("Результаты 5-кратной кросс-валидации:")
        print(f"Точность (Accuracy): {scores['test_accuracy'].mean():.3f}")
        print(f"Прецизия (Precision): {scores['test_precision'].mean():.3f}")
        print(f"Полнота (Recall): {scores['test_recall'].mean():.3f}")
        print(f"F1-мера: {scores['test_f1'].mean():.3f}")

        # Обучение на всей тренировочной выборке
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_
size=0.2, random_state=42)
        pipeline.fit(X_train, y_train)
        y_pred = pipeline.predict(X_test)
        print("\nОтчёт по тестовой выборке:")
        print(classification_report(y_test, y_pred))
```

```
        joblib.dump(pipeline, 'spam_classifier.pkl')
        print("Модель сохранена в spam_classifier.pkl")

    # --- Классификация новых сообщений ---
    def classify_sms(text, model_path='spam_classifier.pkl'):
        if not os.path.exists(model_path):
            raise FileNotFoundError("Модель не найдена. Пожалуйста, обучите
её сначала.")

        model = joblib.load(model_path)
        pred = model.predict([text])[0]
        prob = model.predict_proba([text])[0][pred]
        return {
            'result': 'SPAM' if pred == 1 else 'NOT SPAM',
            'confidence': round(prob, 3)
        }

    # --- Пример использования ---
    if __name__ == '__main__':
        train_model('spam.csv')
        test_text = "Выиграй миллион прямо сейчас! Быстро, бесплатно, без
регистрации!"
        result = classify_sms(test_text)
        print("Результат:", result['result'])
        print("Уверенность:", result['confidence'])
```

Сравнение эффективности эвристического и машинного методов

Эксперименты проводились на подмножестве SMS Spam Collection Dataset, включающем 1000 текстовых сообщений (500 — спам, 500 — не спам), предварительно очищенных и токенизированных. Подмножество было сформировано путём случайной стратифицированной выборки для сохранения баланса классов. Для эвристического метода порог в 3 балла был выбран эмпирически на основе анализа ложных срабатываний на валидационной выборке. Для машинного обучения гиперпараметры логистической регрессии (параметр регуляризации C в диапазоне [0.1, 1, 10]) подбирались с использованием поиска по сетке (GridSearchCV). Параметры TF-IDF включали максимальный размер словаря 3000 слов и исключение слов с частотой менее 5. Время классификации измерялось как среднее время предсказания для одного сообщения, исключая предобработку данных (табл. 1).

Для сравнения применялись стандартные метрики качества классификации:

- Accuracy (точность) — доля правильно классифицированных сообщений;
- Precision (прецизия) — доля правильно распознанных спамов среди всех помеченных как спам;
- Recall (полнота) — доля правильно распознанных спамов среди всех реальных спамов;
- F1-мера — гармоническое среднее между Precision и Recall, отражает баланс между ними;
- Время классификации одного сообщения.

Таблица 1

Результаты сравнения.

Метрика	Эвристический метод	Машинное обучение
Accuracy	84.1%	96.7%
Precision	87.4%	97.9%
Recall	81.6%	95.5%
F1-мера	84.3%	96.7%
Время на сообщение	0.014 с	0.097 с
Обновление правил	Вручную	Автоматически
Гибкость	Низкая	Высокая

Машинное обучение демонстрирует значительно лучшие результаты по всем основным метрикам качества классификации, особенно в условиях увеличенного объёма данных. Эвристический подход выигрывает по скорости обработки и простоте реализации, но уступает в гибкости и способности адаптироваться к новым типам спама.

Разработанные системы классификации текстовых сообщений демонстрируют два подхода к решению задач информационной безопасности: эвристический и основанный на машинном обучении. Эвристический классификатор показывает приемлемую точность и высокую скорость обработки, что делает его подходящим для использования в условиях ограниченных ресурсов и необходимости полной автономности. В то же время система, построенная на методах машинного обучения, продемонстрировала значительно более высокие показатели точности, полноты и F1-меры, что указывает на её эффективность в условиях сложной и динамичной структуры входящих данных. Сравнительный анализ подтвердил, что каждый из подходов имеет свои преимущества в зависимости от целей, требований и условий эксплуатации. Однако, система на основе машинного обучения оказывается более перспективной в контексте дальнейшего развития, особенно с учётом возможности адаптации к новым видам угроз. Полученные результаты подчёркивают актуальность применения современных методов анализа текста в задачах обеспечения информационной безопасности и позволяют сформировать основу для разработки интеллектуальных фильтрующих систем нового поколения.

Литература

1. Волкова, С. С. Введение в машинное обучение. Линейные модели : учебное пособие / С. С. Волкова. – Вологда: Вологод, 2023. – 76 с. – ISBN 978-5-907606-46-3. – EDN RXAMCS (дата обращения 28.10.2025).

2. Спам-фильтрация электронных почтовых сообщений на основе нейросетевой и нейронечеткой моделей / А. С. Катасев, Д. В. Катасева, А. П. Кирпичников, Я. Е. Семенов // Вестник Технологического университета. – 2015. – Т. 18, № 15. – С. 217-220. – EDN ULRPSX (дата обращения 03.11.2025).

3. Гуров, В. В. Спам-фильтры для предприятий / В. В. Гуров // Сети и системы связи. – 2007. – № 6. – С. 80-89. – EDN HZTOIL (дата обращения 03.11.2025).

4. Анализ данных и процессов / А. Барсегян, М. Куприянов, И. Холод [и др.]. – 3-е изд. – Санкт-Петербур: БХВ-Петербург, 2010. – 512 с. – ISBN 978-5-9775-0368-6. – EDN SDPQDT (дата обращения 05.11.2025).

5. Информационная безопасность и защита персональных данных. Проблемы и пути их решения: сборник материалов и докладов XVI межрегиональная научно-практическая конференция, Брянск, 29 апреля 2024 года. – Брянск: Брянский государственный технический университет, 2024. – 320 с. – ISBN 978-5-907570-87-0. – EDN LVASON (дата обращения 05.11.2025).

6. Повышение эффективности фильтрации спама на основе методов машинного обучения в сообщениях различной природы / И. А. Чижова, Е. И. Кублик, М. С. Чипчагов, А. И. Лабинцев // Нейрокомпьютеры: разработка, применение. – 2022. – Т. 24, № 5. – С. 5-18. – DOI 10.18127/j19998554-202205-01. – EDN RYRSXG (дата обращения 05.11.2025).

References

1. Volkova, S. S. Vvedeniye v mashinnoye obucheniye. Lineynyye modeli : uchebnoye posobiye / S. S. Volkova. – Vologda: Vologod, 2023. – 76 s. – ISBN 978-5-907606-46-3. – EDN RXAMCS (data obrashcheniya 28.10.2025).

2. Spam-fil'tratsiya elektronnykh pochtovykh soobshcheniy na osnove neyrosetevoy i neyronechetkoy modeley / A. S. Katasev, D. V. Kataseva, A. P. Kirpichnikov, YA. Ye. Semenov // Vestnik Tekhnologicheskogo universiteta. – 2015. – T. 18, № 15. – S. 217-220. – EDN ULRPSX (data obrashcheniya 03.11.2025).

3. Gurov, V. V. Spam-fil'try dlya predpriyatiy / V. V. Gurov // Seti i sistemy svyazi. – 2007. – № 6. – S. 80-89. – EDN HZTOIL (data obrashcheniya 03.11.2025).

4. Analiz dannykh i protsessov / A. Barsegyan, M. Kupriyanov, I. Kholod [i dr.]. – 3-ye izd. – Sankt-Peterbur: BKHV-Peterburg, 2010. – 512 s. – ISBN 978-5-9775-0368-6. – EDN SDPQDT (data obrashcheniya 05.11.2025).

5. Informatsionnaya bezopasnost' i zashchita personal'nykh dannykh. Problemy i puti ikh resheniya: sbornik materialov i dokladov KHVI mezhregional'naya nauchno-prakticheskaya konferentsiya, Bryansk, 29 aprelya 2024 goda. – Bryansk: Bryanskiy gosudarstvennyy tekhnicheskiy universitet, 2024. – 320 s. – ISBN 978-5-907570-87-0. – EDN LVASON (data obrashcheniya 05.11.2025).

6. Povysheniye effektivnosti fil'tratsii spama na osnove metodov mashinnogo obucheniya v soobshcheniyakh razlichnoy prirody / I. A. Chizhova, Ye. I. Kublik, M. S. Chipchagov, A. I. Labintsev // Neyrokomp'yutery: razrabotka, primeneniye. – 2022. – T. 24, № 5. – S. 5-18. – DOI 10.18127/j19998554-202205-01. – EDN RYRSXG (data obrashcheniya 05.11.2025).

КУШНИР Надежда Владимировна, старший преподаватель кафедры информационных систем и программирования ФГБОУ ВО «Кубанский государственный технологический университет». 350072, г. Краснодар, ул. Московская, д. 2. E-mail: kushnir.06@mail.ru

ВЛАСЕНКО Александра Владимировна, кандидат технических наук, доцент, Врио заведующего кафедрой информационной безопасности, профессор кафедры ФГКОУ ВО «Краснодарский университет МВД России», 350005, г. Краснодар, ул. Ярославская 128. E- mail: alex_vlasenko@list.ru

БУЛГАКОВ Владислав Витальевич, студент 3 курса направления подготовки 09.03.04 (программная инженерия) факультета информационных технологий и кибербезопасности (ФИТК) ФГБОУ ВО «Кубанский государственный технологический университет». 350072, г. Краснодар, ул. Московская, д. 2. E-mail: vlad1230943@mail.ru

АКУЛИНИН Глеб Александрович, студент 3 курса направления подготовки 09.03.04 (программная инженерия) факультета информационных технологий и кибербезопасности (ФИТК) ФГБОУ ВО «Кубанский государственный технологический университет». 350072, г. Краснодар, ул. Московская, д. 2. E-mail: akuliningleb1@mail.ru

KUSHNIR Nadezhda Vladimirovna, Senior Lecturer at the Department of Information Systems and Programming of the Federal State Budgetary Educational Institution of Higher Education «Kuban State Technological University». 350072, Krasnodar, Moskovskaya St., 2. E-mail: kushnir.06@mail.ru

VLASENKO Alexandra Vladimirovna, Candidate of Technical Sciences, Associate Professor, Acting Head of the Department of Information Security Professor of the Department, of the Krasnodar University of the Ministry of Internal Affairs of Russia. 350005, Krasnodar, Yaroslavskaya str., 128. E-mail: alex_vlasenko@list.ru

BULGAKOV Vladislav Vitalievich, Third-year year student of the training course 09.03.04 (software Engineering) of the Federal State Budgetary Educational Institution of Higher Education «Kuban State Technological University». 350072, Krasnodar, Moskovskaya St., 2. E-mail: vlad1230943@mail.ru

AKULININ Gleb Alexandrovich, Third-year year student of the training course 09.03.04 (software Engineering) of the Faculty of Information Technology and Cybersecurity (FITC) of the Federal State Budgetary Educational Institution of Higher Education «Kuban State Technological University». 350072, Krasnodar, Moskovskaya St., 2. E-mail: akuliningleb1@mail.ru