УДК 004.056:004.89
DOI: 10.37661/1816-0301-2025-22-3-83-94

Оригинальная статья
Original Article

Программный модуль для детектирования мошеннических веб-сайтов с использованием классификации на основе методов машинного обучения

С. Н. Петров[1]✉, А. О. Мяделец[2], Е. В. Кундас[2]

[1]Белорусский государственный университет информатики и радиоэлектроники,
ул. П. Бровки, 6, Минск, 220013, Беларусь
✉E-mail: sergpetrov@inbox.ru

[2]Национальный детский технопарк,
ул. Франциска Скорины, 25/3, Минск, 220076, Беларусь

Аннотация

Цели. Целью исследования является разработка программного модуля для автоматического выявления фишинговых веб-сайтов с использованием алгоритмов машинного обучения для классификации сайтов.

Методы. Для достижения поставленной цели проведен анализ существующих датасетов, содержащих URL-адреса фишинговых сайтов, а также изучены датасеты для обработки естественного языка. Это позволило определить ключевые признаки, характерные для мошеннических ресурсов. Были созданы два набора данных (размерами 18,9 Мб и 1,08 Гб), включающих признаки URL и текстовое наполнение веб-страниц, с использованием разработанного парсера. Для классификации веб-ресурсов применялись алгоритмы машинного обучения, такие как SVM, Random Forest, Logistic Regression и Multilayer Perceptron (MLP). Также изучены возможности использования языковой модели TinyBERT для анализа текстового содержимого.

Результаты. По результатам проведенных исследований для работы с URL использована модель MLP (F1-score 99,3 %), а для анализа текстовой части веб-ресурса – модель TinyBERT (F1-score 95 %). Разработан программный модуль для выявления мошеннических веб-сайтов, состоящий из серверной части и браузерного расширения. Расширение собирает данные с веб-ресурса, передает их на сервер, где они анализируются обученными моделями машинного обучения. На сервере рассчитывается вероятность фишинговой активности, а результаты отображаются пользователю через интерфейс расширения. Реализация выполнена с использованием стека технологий Python 3.12, Flask, Pickle, Langdetect, Re и NLTK, а также JavaScript и Google Chrome API.

Заключение. Разработанный программный модуль был протестирован и продемонстрировал высокую эффективность в задачах классификации фишинговых сайтов. Теоретическая значимость работы заключается в применении современных алгоритмов машинного обучения для анализа текстового контента и URL. Практическая значимость заключается в создании готового решения для выявления фишинговых сайтов в реальном времени.

Ключевые слова: фишинговые сайты, мошенничество, машинное обучение, классификация, обработка естественного языка, датасеты

Для цитирования. Петров, С. Н. Программный модуль для детектирования мошеннических веб-сайтов с использованием классификации на основе методов машинного обучения / С. Н. Петров, А. О. Мяделец, Е. В. Кундас // Информатика. – 2025. − Т. 22, № 3. – С. 83–94. – DOI: 10.37661/1816-0301-2025-22-3-83-94.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 28.07.2025
Подписана в печать | Accepted 11.08.2025
Опубликована | Published 30.09.2025

Software module for detecting fraudulent websites using classification based on machine learning methods

Sergei N. Petrov[1]✉, Artyom O. Myadelets[2], Elizaveta V. Kundas[2]

[1]Belarusian State University of Informatics and Radioelectronics, st. P. Brovki, 6, Minsk, 220013, Belarus
✉E-mail: sergpetrov@inbox.ru

[2]National Children's Technopark, st. Francis Skorina, 25/3, Minsk, 220076, Belarus

Abstract

Objectives. Phishing web resources are among the most common tools of online fraud aimed at obtaining users' confidential information. The goal of this research was to develop a software module for the automatic detection of phishing websites using machine learning methods.

Methods. To achieve this goal, an analysis of existing datasets containing phishing website URLs was conducted, along with the study of datasets for natural language processing (NLP). This enabled the identification of key features characteristic of fraudulent resources. Two datasets were created (sizes: 18.9 MB and 1.08 GB), incorporating URL attributes and web page content, using a custom-developed parser. Machine learning algorithms such as SVM, Random Forest, Logistic Regression, and Multilayer Perceptron (MLP) were applied for website classification. The potential of the TinyBERT language model for analyzing textual content was also explored.

Results. The analysis revealed that the MLP model demonstrated the best performance for URL classification, while the TinyBERT model excelled in analyzing textual content. A software module was developed, consisting of a server-side application and a browser extension. The extension collects data from web resources, transmits them to the server, where trained machine learning models analyze the information. The server calculates the likelihood of phishing activity, and the results are displayed to the user via the extension's interface. The implementation utilized a technology stack including Python 3.12, Flask, Pickle, Langdetect, Re, NLTK, JavaScript, and the Google Chrome API.

Conclusion. The developed software module was tested and demonstrated high efficiency in phishing website classification tasks. The theoretical significance of the work lies in applying modern machine learning algorithms for analyzing textual content and URLs. The practical significance is reflected in the creation of a ready-to-use solution for real-time phishing site detection.

Keywords: phishing websites, fraud, machine learning, classification, natural language processing, datasets

For citation. Petrov S. N., Myadelets A. O., Kundas E. V. *Software module for detecting fraudulent websites using classification based on machine learning methods.* Informatika [*Informatics*], 2025, vol. 22, no. 3, pp. 83–94 (In Russ.). DOI: 10.37661/1816-0301-2025-22-3-83-94.

Conflict of interest. The authors declare of no conflict of interest.

Введение. Мошенническим сайтом (веб-ресурсом) является любой сайт, созданный для обмана пользователей с целью получения незаконной прибыли. Частным случаем мошеннического сайта является фишинговый сайт, который имитирует настоящий сайт (реально существующий, законный сайт) с целью получения конфиденциальной информации пользователей, такой как пароли учетных записей, номера кредитных карт, паспортные и персональные данные. В Республике Беларусь действует уголовная ответственность за мошенничество с использованием веб-ресурсов в соответствии со статьей 212 УК РБ «Хищение имущества путем модификации компьютерной информации».

Фишинг (от англ. phishing, производное от fishing – «рыбная ловля», «выуживание») представляет собой вид мошенничества, основанный на использовании методов социальной инженерии. Злоумышленники выдают себя за представителей известных организаций, банков или сервисов и создают поддельные веб-страницы, которые практически не отличаются от оригинальных [1]. Попав на такую страницу, пользователь может даже не подозревать, что взаимодействует с поддельным сервисом, созданным мошенниками.

Мошеннический сайт (веб-ресурс) можно определить по нескольким характерным признакам. Один из самых очевидных и часто встречающихся признаков – это искаженный URL-адрес. Мошенники часто используют тайпсквоттинг, заменяя буквы похожими символами или изменяя структуру адреса. Например, вместо «bank.com» пользователь может увидеть адрес «bank.com», где латинская буква «а» заменена на кириллическую «а», что визуально незаметно. Также мошенники могут использовать другую доменную зону (например, .com вместо .ru), допускать намеренные опечатки или добавлять лишние символы в адресе. Все это делает поддельные сайты (домены) похожими на оригинальные и затрудняет их распознавание.

Кроме того, фишинговые сайты часто можно узнать по содержимому страниц. Мошенники стремятся вызвать у пользователя чувство срочности и заставить его действовать немедленно. Текст на таких страницах может содержать ошибки или быть сформулирован так, чтобы побудить пользователя раскрыть личные данные. Например, популярным приемом является обещание крупных выигрышей, бонусов или специальных предложений, которые доступны только ограниченное время. Такие уловки призваны отвлечь внимание от подозрительных элементов сайта и побудить пользователя действовать импульсивно.

Мошеннические веб-ресурсы создаются массово, что также является важным признаком фальшивого сайта. Часто можно заметить, что на таких страницах используются одни и те же шаблоны, а контактные данные, такие как электронные адреса или номера телефонов, могут повторяться на разных ресурсах. Все это свидетельствует о том, что за созданием подобных сайтов стоят организованные группы мошенников, которые используют автоматизированные инструменты для быстрого клонирования веб-страниц.

Количество мошеннических сайтов растет с каждым годом, и фишинговые атаки становятся все более сложными и масштабными. Это создает серьезные риски как для обычных пользователей, так и для крупных компаний, которым приходится тратить значительные ресурсы на обеспечение безопасности своих клиентов. В условиях таких угроз традиционные методы защиты становятся недостаточными, и все большую роль начинают играть технологии машинного обучения.

Применение алгоритмов машинного обучения, позволяющих обрабатывать большие объемы данных и выявлять скрытые закономерности, обеспечивает возможность классификации веб-сайтов на подлинные и мошеннические. Например, обученная модель может учитывать различные признаки URL-адреса, структуру HTML-кода и даже содержимое страницы, чтобы предсказать вероятность того, что данный ресурс является фишинговым. Процесс детектирования мошеннических сайтов сводится к решению задачи классификации, в которой все множество сайтов делится на два подмножества (класса) – мошеннические сайты и легитимные (нормальные).

Целью исследовательского проекта является разработка программного модуля с использованием машинного обучения для выявления мошеннических веб-ресурсов. Для достижения этой цели необходимо решить несколько задач: проанализировать существующие алгоритмы классификации, создать набор данных для обучения моделей, выбрать оптимальные модели классификации, реализовать программный модуль анализа веб-ресурсов на основе выбранных моделей.

Наборы данных для обучения моделей. Датасет в контексте машинного обучения – это набор данных, который используется для обучения модели или ее тестирования. Датасеты для выявления мошеннических веб-ресурсов могут быть собраны из различных источников, включая специализированные базы данных, общедоступные репозитории, или создаваться вручную на основе анализа реальных примеров мошеннических сайтов. Датасет состоит из набора объектов, каждый из которых содержит одну или несколько функций и соответствующие признаки – индивидуальные измеряемые свойства или характеристики. Примером таких признаков являются параметры, извлеченные из URL-адресов [2]. Это статистические данные, исходя из которых можно определить, является сайт безопасным для использования или нет. Признаки могут включать в себя протокол, доменное имя, путь к ресурсу, количество определенных символов в URL-адресе, наличие iframe и т. д.

Программному модулю необходимо анализировать как URL-адреса веб-ресурсов, так и текстовое содержимое (контент) страниц [3]. Это требует подготовки двух различных датасетов. Первый будет предназначен для обучения модели выявлению признаков мошенничества непосредственно в URL, таких как подозрительные символы, нестандартные доменные зоны или подменные домены. Второй датасет необходим для обучения NLP-модели. Обработка естественного языка (Natural Language Processing, NLP) позволяет анализировать текст на страницах сайтов, чтобы выявлять подозрительные паттерны, такие как завлекающие сообщения или запросы конфиденциальной информации.

Были рассмотрены открытые ресурсы по тематике машинного обучения, и проведен анализ популярных датасетов [4], результаты которого представлены в табл. 1.

Таблица 1
Информация о датасетах для выявления фишинговых URL

Table 1
Information about Datasets for Phishing URL Detection

Название *Title*	PhiUSIIL Phishing URL	Phishing Websites Dataset	Phishing Websites Dataset Cleaned	Webpage Phishing Detection Dataset
Ссылка *Link*	https://archive.ics.uci.edu/dataset/967/phiusiil+phishing+url+dataset/	https://data.mendeley.com/datasets/72ptz43s9v/1/	https://www.kaggle.com/datasets/prishasawhney/phishing-url-website-dataset-cleaned/	https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset/
Год издания *Year of Publication*	2023	2020	2024	2020
Размер, Мб *Size (MB)*	54,2	23,9	21,77	3,66
Кол-во URL-адресов *Number of URLs*	235 795	88 647	176 263	11 430
Кол-во признаков *Number of Features*	54	111	20	87

На рис. 1 изображен график распределения классов датасетов, приведенных в табл. 1.

Датасет Webpage Phishing Detection Dataset обладает сбалансированным соотношением классов, что делает его привлекательным для обучения модели. Однако его существенным недостатком является небольшое количество URL-адресов по сравнению с другими наборами данных, что ограничивает его использование при работе с более сложными моделями.

Поиск датасетов для обучения NLP-модели не был результативным. В открытом доступе практически нет наборов данных на русском языке, которые бы подходили для анализа содержимого фишинговых сайтов. Большинство доступных датасетов основаны на английском языке или не содержат необходимой информации для успешного решения поставленной задачи.

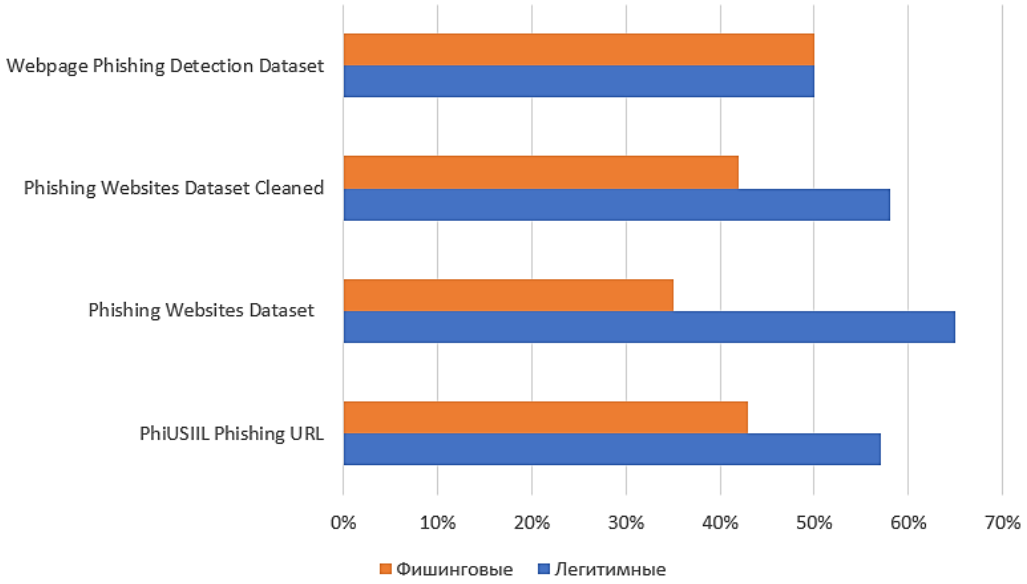


Рис. 1. Распределение классов в датасетах

Fig. 1. Classes Distribution in Datasets

Наиболее подходящим вариантом оказался Dataset Advert-Spam (Russia) (https://www.kaggle.com/datasets/sanarovmichael/dataset-advert-spam) на платформе Kaggle. Его объем составляет 2,6 Мб, и он включает только примеры фишинговых спам-сообщений. Основной контент датасета составляют тексты, побуждающие оформить кредит или банковский вклад, навязчивая реклама с заманчивыми предложениями, а также сообщения с орфографическими ошибками.

Анализ открытых источников показал отсутствие готовых решений для поставленной задачи. В связи с этим было принято решение создать собственные датасеты, которые позволят систематизировать данные, необходимые для обучения моделей на основе анализа как URL-адресов, так и текстов веб-страниц.

Создание собственных наборов данных. Для создания собственных датасетов был разработан программный компонент (парсер), который автоматически собирает и анализирует данные из URL-адресов и HTML-кода страниц. Код написан на языке программирования Python из-за его кроссплатформенности, простоты и наличия необходимых библиотек для обработки данных. В качестве среды разработки использовалась Visual Studio Code.

За основу создания датасета, содержащего признаки URL, был взят Webpage Phishing Detection Dataset. На его базе сформирован датасет, включающий данные из Webpage Phishing Detection Dataset и дополненный данными с платформ PhishTank (https://dev.phishtank.com) и OpenPhish (https://openphish.com). Эти платформы служат для обмена данными и информацией о фишинге и содержат большое количество ссылок на фишинговые ресурсы. Однако сами по себе данные на этих платформах не подходят для обучения моделей и должны быть обработаны для извлечения информативных признаков. Были извлечены признаки из URL. После этого парсером обрабатывались непосредственно веб-страницы, размещенные по указанным URL.

Алгоритм работы парсера для формирования URL-датасета:

1. Чтение URL-адресов из исходного файла.
2. Анализ структуры URL (длина, количество символов, поддоменов и т. д.).

3. Проверка на наличие подозрительных признаков (наличие слов типа login, secure).

4. Скачивание веб-страницы (если доступно) для анализа HTML-кода.

5. Сохранение результатов (все извлеченные характеристики записываются в CSV-файл).

Получился датасет размером 18,9 Мб, содержащий около 88 687 URL-адресов, каждый из которых анализируется по 63 признакам. Примеры 25 признаков (выборочно) приведены в табл. 2.

Таблица 2
Информация о датасетах для выявления фишинговых URL

Table 2
Information about Datasets for Phishing URL Detection

Признак *Feature*	Описание признака *Description of the feature*
url	URL-адрес
length_url	Длина URL
length_hostname	Длина имени хоста
ip	Является ли имя хоста IP-адресом
nb_dots	Количество точек в URL
nb_hyphens.	Количество дефисов в URL
nb_at	Количество символов @ в URL
nb_qm	Количество символов ? в URL
nb_and	Количество символов & в URL
nb_or	Количество вхождений подстроки or в URL
http_in_path	Наличие подстроки http в пути
https_token	Наличие подстроки https в URL
ratio_digits_url	Отношение числа цифр к длине URL
ratio_digits_host	Отношение числа цифр к длине имени хоста
punycode	Наличие Punycode, т. е. подстроки xn--
port	Наличие порта в URL (например, :8080)
tld_in_path	Наличие доменного уровня в пути
tld_in_subdomain	Наличие доменного уровня в поддомене
nb_redirection	Количество перенаправлений
nb_external_redirection	Количество внешних перенаправлений
iframe	Наличие элемента iframe на странице
popup_window	Наличие всплывающих окон
right_clic	Блокировка правой кнопки мыши на странице
phish_hints	Наличие ключевых слов, часто встречающихся на фишинговых сайтах (например, login, secure, account)
domain_in_brand	Наличие известных брендов в домене

Алгоритм работы парсера для формирования текстового датасета:

1. Чтение URL-адресов из исходного файла.

2. Анализ HTML-кода (удаляет HTML-теги, выполняет стемминг и удаляет стоп-слова).

3. Векторизация текста с использованием TfidfVectorizer. TfidfVectorizer – это один из инструментов NLP, используемый для преобразования текстовых данных в числовой формат на основе статистической меры TF-IDF (Term Frequency-Inverse Document Frequency). Он оценивает важность слов в документе относительно всего корпуса, снижая влияние часто встречающихся, но малоинформативных слов.

4. Сохранение результатов (все извлеченные характеристики записываются в joblib файл).

Размер итогового текстового датасета составляет 1,08 Гб. В результате были созданы два датасета размерами 18,9 Мб и 1,08 Гб соответственно для выявления мошеннических веб-ресурсов путем анализа их URL и текстового наполнения.

Распределение классов URL-датасета показано на рис. 2, *a*, а распределение классов для текстового датасета – на рис. 2, *b*.

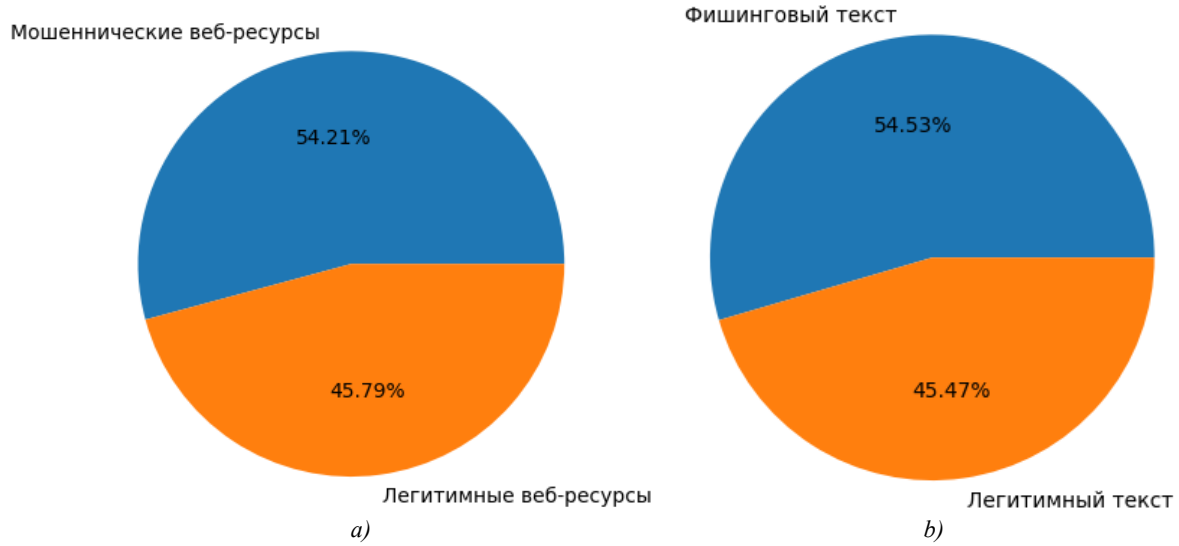


Рис. 2. Распределение классов в датасете: *a)* URL-датасет; *b)* текстовый датасет

Fig. 2. Class Distribution in the dataset: a) URL dataset; b) text dataset

Выбор оптимальной модели для классификации веб-ресурсов. Для выявления мошеннических веб-ресурсов были выбраны четыре алгоритма машинного обучения [5]: Support Vector Machine, Random Forest, Logistic Regression и Multilayer Perceptron.

Support Vector Machine (SVM) – надежный алгоритм для задач классификации, особенно с высокоразмерными данными. Линейное ядро подходит для линейно разделимых данных, RBF – для нелинейных, полиномиальное ядро применяется для сложных зависимостей, сигмоидное используется реже из-за нестабильности.

Random Forest (RF) объединяет несколько деревьев решений для повышения точности и устойчивости модели, особенно эффективен при большом количестве признаков.

Logistic Regression (LR) хорошо подходит для бинарной классификации и позволяет оценивать вероятность принадлежности к классам.

Multilayer Perceptron (MLP) – это нейросеть, решающая как линейные, так и нелинейные задачи за счет скрытых слоев и функций активации.

Для оценки моделей машинного обучения в задаче классификации используются различные метрики, каждая из которых отражает определенный аспект работы модели [6]:

– Accuracy отражает долю верных предсказаний, но на несбалансированных данных может вводить в заблуждение, так как модель может игнорировать редкий класс и все равно показывать высокий результат;

– Precision показывает, сколько из предсказанных мошенничеств действительно оказались мошенничеством;

– Recall измеряет, сколько из всех реальных случаев мошенничества модель смогла обнаружить;

– F1-score объединяет Precision и Recall в одно значение и позволяет сбалансированно оценить модель. Это необходимо, чтобы минимизировать ложные срабатывания и не пропускать реальные случаи мошенничества.

Для подбора гиперпараметров использовался метод GridSearchCV, который перебирает все возможные комбинации параметров и оценивает их с помощью кросс-валидации, что позволяет автоматически находить оптимальные настройки модели и избегать ручного подбора. Обучение моделей проводилось в Google Colab, который предоставляет бесплатный доступ к облачным вычислительным ресурсам, включая GPU, что ускорило процесс обучения.

В табл. 3 приведены лучшие результаты каждой модели (по точности и времени обучения и предсказания) по обработке URL.

Таблица 3
Результаты моделей при работе с URL

Table 3
Model Results for working with URL

Алгоритм *Algorithm*	Точность, % *Accuracy, %*	Полнота, % *Recall, %*	Достоверность, % *Precision, %*	F1-мера, % *F1-score, %*	Время обучения, с *Trainig time, s*	Время предсказания одной строки, с *Time for predicting a single line, s*
RF	99,3	99,4	99,6	99,5	5,7	0,21
LR	98,7	98,6	99,6	99,1	2	0,01
SVM	91	99,5	89,4	94,1	2113	0,08
MLP	99	98,9	99,7	99,3	40	0,02

Рассмотренные модели не показали высоких результатов при анализе текстового содержимого веб-страниц. Лучший результат показала модель Random Forest (точность 54,5 %, полнота 99,8 %, достоверность 54,5 %, F1-мера 70,6 %). По результатам поиска решения для анализа текстовой части веб-ресурса использована большая языковая модель (LLM) BERT [7, 8], а именно TinyBERT [9] – компактная версия модели, оптимизированная для ускорения работы и уменьшения вычислительных затрат. TinyBERT эффективна для задач обработки естественного языка, таких как классификация текстов и анализ тональности, а за счет меньшего числа слоев и параметров подходит для приложений, работающих в реальном времени. Модель была дообучена с использованием созданного текстового датасета, описанного ранее.

Дообучение модели проводилось с использованием технологии переноса знаний, что позволило сократить время обучения и повысить точность. TinyBERT научилась успешно распознавать признаки фишинговых текстов и сохранять высокую точность даже на коротких сообщениях или текстах с орфографическими ошибками. Полученные результаты представлены в табл. 4.

Таблица 4
Результаты работы модели TinyBERT

Table 4
Results of TinyBERT working

Модель *Model*	Точность, % *Accuracy, %*	Полнота, % *Recall, %*	Достоверность, % *Precision, %*	F1-мера, % *F1-score, %*	Время обучения, с *Trainig time, s*	Время предсказания одной строки, с *Time for predicting a single line, s*
TinyBERT	95	91	99	95	1113	0,044

Из табл. 4 видно, что LLM-модель показала очень высокую точность распознавания фишинговых текстов, а время предсказания оказалось меньше, чем у моделей SVM, Logistic Regression, Random Forest и Multilayer Perceptron. Единственным недостатком оказалось долгое время обучения модели – порядка 18 мин. Однако при работе модуля в реальных условиях это не будет являться существенным недостатком. Проводить дообучение модели можно будет с определенной периодичностью по мере необходимости.

Таким образом, при разработке программного модуля для анализа URL выбрана модель MLP, а для анализа текстовой части – модель TinyBERT.

Разработка программного модуля для классификации веб-сайтов. Программный модуль для выявления мошеннических веб-сайтов состоит из серверной части и браузерного расширения. Расширение собирает данные с веб-ресурса и отправляет их на сервер, где они анализируются обученными ML-моделями. Сервер принимает данные методом POST, обрабатывает их (стемминг, удаление стоп-слов), предсказывает вероятность того, что сайт является фишинговым, после чего по методу POST возвращает результат в расширение, которое отображает результаты для пользователя (рис. 3). В проекте использовался стек технологий, включающий Python 3.12

для серверной части и анализа данных с помощью библиотек Flask, Pickle, Langdetect, Re и NLTK. Браузерное расширение разработано на JavaScript с использованием Google Chrome API и оформлено с помощью HTML/CSS. Для написания и отладки кода использовалась среда Visual Studio Code. Результат работы программного модуля показан на рис. 4.

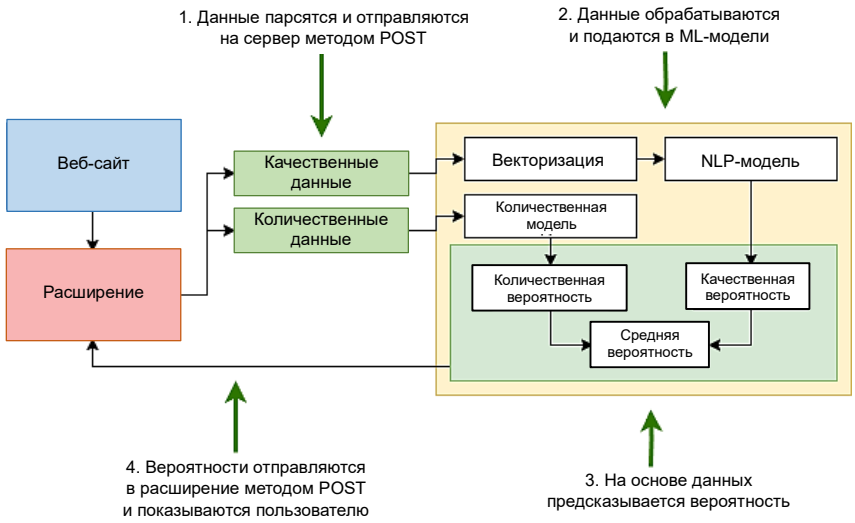


Рис. 3. Схема работы программного модуля

Fig. 3. Operating diagram of the software module

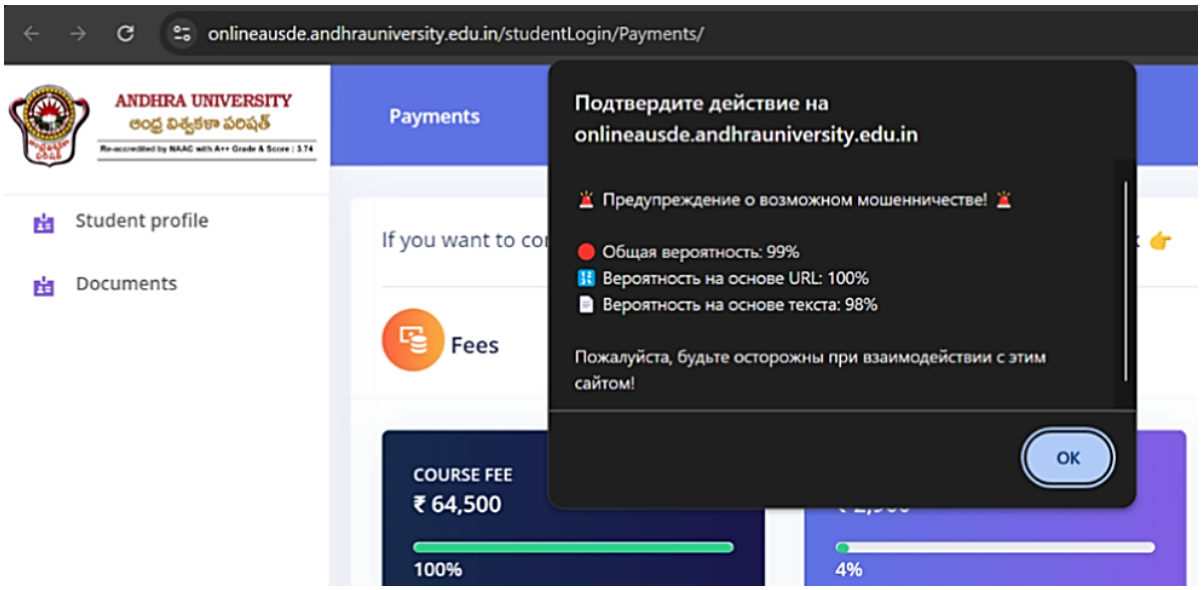
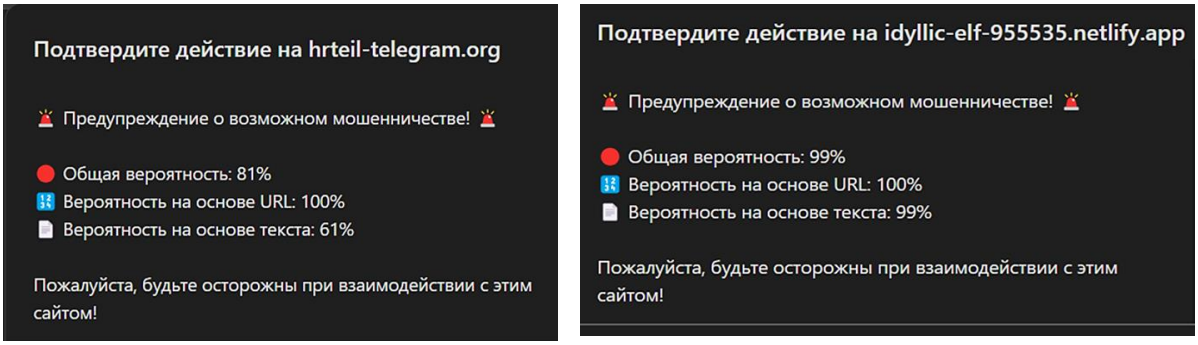


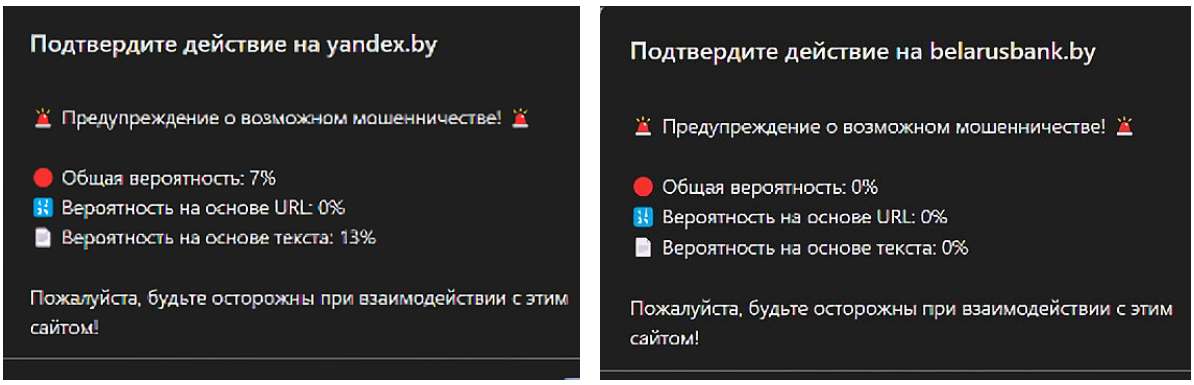
Рис. 4. Пример уведомления о мошенничестве

Fig. 4. Fraud notification example

Модуль был протестирован на группе веб-сайтов, заранее разделенных на заведомо мошеннические (рис. 5, *a*) и легитимные (рис. 5, *b*). В результате тестирования модуль показал высокую степень эффективности выявления такого типа ресурсов. Среднее время предсказания составило 0,11 с.



a)



b)

Рис. 5. Результаты тестирования: *a)* на заведомо мошеннических веб-сайтах; *b)* на легитимных веб-сайтах

Fig. 5. Test results: a) on obviously fraudulent websites; b) on obviously legitimate websites

Результаты проверки ряда веб-ресурсов представлены в табл. 5.

Таблица 5
Результаты проверки веб-ресурсов разработанным модулем

Table 5
Results of checking web resources by the developed module

URL	Вероятность по URL, % *Probability by URL, %*	Вероятность по тексту, % *Probability by text, %*	Средняя вероятность, % *Average probability, %*
https://www.bsuir.by	0	0	0
https://mail.ru	0	0	0
https://habr.com/	0	0	0
https://yandex.by	0	13	7
https://bsu.by	0	32	16
https://domain.by	0	34	17
http://myfin.by	0	32	16
https://aliexpress.by	2	59	30
https://belta.by	0	29	15
https://president.gov.by	0	0	0
https://connect-chain-wallet.web.app/app	100	53	77
https://accoun-at-risk.github.io/anesh-here	100	99	100
https://steamcummunity.ru	100	99	100
https://busca-lphone.com	98	100	100
https://metmasklоginie.webflow.io/	100	99	99
https://www.metamask-io-help.walletallinone.com/	100	45	73
http://facebook.focal.us.kg/	99	99	95
https://v5-uniswap.info/	100	91	95

Заключение. Проведен анализ открытых датасетов для обучения моделей анализу URL и текстового наполнения веб-ресурсов, результаты которого показали отсутствие подходящих решений для данного проекта. С применением разработанного для этих целей парсера были созданы два датасета размерами 18,9 Мб и 1,08 Гб для выявления мошеннических веб-ресурсов путем анализа их URL и текстового наполнения. По результатам проведенных исследований для работы с URL использована модель MLP (F1-score 99,3 %), а для анализа текстовой части веб-ресурса – модель TinyBERT (F1-score 95 %).

Разработан программный модуль для выявления мошеннических веб-сайтов, состоящий из серверной части и браузерного расширения. Результаты проверки ряда веб-ресурсов показали высокую эффективность разработанного модуля. В рамках работы сервер запускался локально. В перспективе – реализация серверной части онлайн, что позволит большому числу пользователей осуществлять проверку веб-ресурсов, используя одно только легковесное браузерное расширение, а поддержку серверной части будут осуществлять разработчики.

С исходным кодом программного модуля, а также файлами исследования ML-моделей можно ознакомится в репозитории Github по ссылке https://github.com/Param0rph/PhishDetect. Для того чтобы установить браузерное расширение, необходимо перейти в chrome://extensions/ в браузере Google Chrome, включить режим разработчика (Developer mode), в панели управления расширениями выбрать Load unpacked и загрузить папку с расширением.

Дополнительные материалы к проекту расположены по ссылке https://drive.google.com/drive/folders/10oIYFJDF_rrpgru4l1cgiZSIeVVJbUZq?usp=sharing.

Вклад авторов. *С. Н. Петров* определил цели исследования и задачи, которые необходимо было решить для их достижения, принял участие в интерпретации и обобщении полученных результатов, написал текст рукописи. *А. О. Мяделец* провел экспериментальные исследования эффективности моделей классификации, спроектировал и разработал программный модуль. *Е. В. Кундас* выполнила сравнительный анализ существующих подходов к обнаружению мошеннических веб-ресурсов, провела анализ открытых датасетов для выявления мошеннических веб-ресурсов, сформировала датасеты для обучения моделей с использованием разработанного парсера.

Список использованных источников

1. Завьялов, А. Н. Интернет-мошенничество (фишинг): проблемы противодействия и предупреждения / А. Н. Завьялов // Baikal Research Journal. – 2022. – Т. 13, № 2. – С. 36.

2. Machine learning techniques for detecting phishing URL attacks / D. T. Mosa, M. Y. Shams, A. A. Abohany [et al.] // Computers, Materials & Continua. – 2023. – Vol. 75, no. 1. – P. 1271–1290. – DOI: 10.32604/cmc.2023.036422.

3. A phishing-attack-detection model using natural language processing and deep learning / E. Benavides-Astudillo, W. Fuertes, S. Sanchez-Gordon [et al.] // Applied Sciences. – 2023. – Vol. 13, iss. 9. – P. 5275.

4. Петров, С. Н. Датасеты для обучения моделей обнаружению мошеннических веб-ресурсов / С. Н. Петров, А. О. Мяделец, Е. В. Кундас // Лучшие студенческие исследования 2025 : сб. ст. IV Междунар. науч.-исслед. конкурса. – Пенза : МЦНС «Наука и Просвещение». – 2025. – С. 27–32.

5. Жерон, О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем / О. Жерон ; пер. с англ. – СПб. : ООО «Альфа-книга», 2018. – 688 с.

6. Шакла, Н. Машинное обучение и TensorFlow / Н. Шакла ; пер. с англ. – СПб. : Питер, 2019. – 336 с.

7. Koroteev, M. V. BERT: A Review of Applications in Natural Language Processing and Understanding / M. V. Koroteev. – URL: https://arxiv.org/pdf/2103.11943 (date of access: 17.03.2025).

8. Worth, P. J. Word embeddings and semantic spaces in natural language processing / P. J. Worth // International Journal of Intelligence Science. – 2023. – Vol. 13, no. 1. – P. 1–21. – DOI: 10.4236/ijis.2023.131001.

9. TinyBERT: Distilling BERT for Natural Language Understanding / X. Jiao, Y. Yin, L. Shang [et al.]. – URL: https://arxiv.org/pdf/1909.10351v5 (date of access: 17.03.2025). – DOI: 10.48550/arXiv.1909.10351.

References

1. Zavyalov A. N. Internet fraud (phishing): problems of counteraction and prevention. *Baikal Research Journal*, 2022, vol. 13, no. 2, p. 36 (In Russ.).

2. Mosa D. T., Shams M. Y., Abohany A. A., El-kenawy E.-S. M., Thabet M. Machine learning techniques for detecting phishing URL attacks. *Computers, Materials & Continua*, 2023, vol. 75, no. 1, pp. 1271–1290. DOI: 10.32604/cmc.2023.036422.

3. Benavides-Astudillo E., Fuertes W., Sanchez-Gordon S., Nuñez-Agurto D., Rodríguez-Galán G. A phishing-attack-detection model using natural language processing and deep learning. *Applied Sciences*, 2023, vol. 13, iss. 9, p. 5275.

4. Petrov S. N., Myadelets A. O., Kundas E. V. *Datasets for training models to detect fraudulent web resources*. Luchshie studencheskie issledovanija 2025 : sbornik statej IV Mezhdunarodnogo nauchno-issledovatel'skogo konkursa [*Best Student Research 2025: Collection of Articles of the IV International Research Competition*]. Penza, Nauka i Prosveshchenie, 2025, pp. 27–32 (In Russ.).

5. Géron A. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2017, 572 p.

6. Shukla N. *Machine Learning with TensorFlow.* Manning, 2018, 272 p.

7. Koroteev M. V. *BERT: A Review of Applications in Natural Language Processing and Understanding*. Available at: https://arxiv.org/pdf/2103.11943 (accessed 17.03.2025).

8. Worth P. J. Word embeddings and semantic spaces in natural language processing. *International Journal of Intelligence Science*, 2023, vol. 13, no. 1, pp. 1–21. DOI: 10.4236/ijis.2023.131001.

9. Jiao X., Yin Y., Shang L., Jiang X., Chen X., …, Liu Q. *TinyBERT: Distilling BERT for Natural Language Understanding*. Available at: https://arxiv.org/pdf/1909.10351v5 (accessed 17.03.2025). DOI: 10.48550/arXiv.1909.10351.

Информация об авторах

Петров Сергей Николаевич, кандидат технических наук, доцент, доцент кафедры защиты информации, факультет инфокоммуникаций, Белорусский государственный университет информатики и радиоэлектроники.
E-mail: sergpetrov@inbox.ru
https://www.elibrary.ru/author_profile.asp?authorid=1088896

Мяделец Артем Олегович, учащийся, Национальный детский технопарк.
E-mail: artemmuadzelets@gmail.com

Кундас Елизавета Владимировна, учащийся, Национальный детский технопарк.
E-mail: kundaselizaveta@gmail.com

Information about the authors

Sergei N. Petrov, Ph. D. (Eng.), Assoc. Prof., Assoc. Prof. of the Information Security Department, Faculty of Infocommunications, Belarusian State University of Informatics and Radioelectronics.
E-mail: sergpetrov@inbox.ru
https://www.elibrary.ru/author_profile.asp?authorid=1088896

Artyom O. Myadelets, Student, National Children's Technopark.
E-mail: artemmuadzelets@gmail.com

Elizaveta V. Kundas, Student, National Children's Technopark.
E-mail: kundaselizaveta@gmail.com