ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ
INFORMATION TECHNOLOGY AND TELECOMMUNICATIONS

УДК 004.056.53

DOI: 10.21822/2073-6185-2024-51-3-154-162 Обзорная статья/ Review article

Обнаружение фишинговых порталов посредством алгоритмов машинного обучения

Е.А. Трушников
Московский технический университет связи и информатики,
123423, г. Москва, ул. Народного Ополчения, 32, Россия

Резюме. Цель. Целью исследования является анализ и практическая реализация функционала обнаружения фишинговых порталов посредством алгоритмов машинного обучения. **Метод**. Систематизация разрозненных сведений, анализ области, описание имеющихся наработок – основные методики, которые были применены в исследовании. Работа разбита на три больших подблока. В первом проводится анализ понятия машинного обучения, описываются основные способы верной интерпретации вводимых сведений, указываются наиболее ходовые методики, базы данных. Во второй части работы проводится анализ искусственных нейронных сетей. В частности, показываются их подвиды с описанием особенностей реализации, проводится сопоставление с живыми нейронами. В третьей части проводится практическая реализация двух методик и их сравнение, даются рекомендации относительно их использования при обнаружении фишинговых порталов. **Результат**. Проведено исследование методик анализа фишинговых порталов. Анализ показал, что наиболее рационально применять случайный лес, т.к. именно он обеспечивает качество по метрикам precession, recall, F1-score, 98% при значительном количестве введенных параметрических значений. **Вывод**. При реализации различных методологий поиска фишинговых порталов необходимо учитывать их снижение эффективности от вводимых параметров. Для этого важно проводить предварительные тесты. Однако результат тестов можно интерпретировать по-разному. В частности, эффективность методов может быть повышена, если ограничить количество вводимых параметров, но при этом жестко структурированных по одному критерию поиска.

Ключевые слова: фишинг, машинное обучение, нейронные сети, анализ порталов

Для цитирования: Е.А. Трушников. Обнаружение фишинговых порталов посредством алгоритмов машинного обучения. Вестник Дагестанского государственного технического университета. Технические науки. 2024; 51(3):154-162. DOI:10.21822/2073-6185-2024-51-3-154-162

Detection of phishing portals through machine learning algorithms

E.A. Trushnikov
Moscow Technical University of Communication and Informatics,
32 Narodnogo Opolcheniya St., Moscow 123423, Russia

Abstract. Objective Analysis and practical implementation of the phishing portal detection functionality through machine learning algorithms. **Method**. Systematization of disparate information, analysis of the field, description of available developments are the main methods that were used in the study. The work is divided into three large sub-blocks. The first one analyzes the concept of machine learning, describes the main ways to correctly interpret the information entered, indicates the most popular techniques and databases. In the second part of the work, an analysis of artificial neural networks is carried out. In particular, their subspecies are shown with a description of the implementation features, and a comparison with living neurons is carried out. In the third part, the practical implementation of the two techniques and their comparison are carried out, recommendations are given regarding their use in detecting phishing portals. **Result**. The paper investigates the methods of analyzing phishing portals. The analysis showed that it is most rational to use a random forest, because it provides quality according

to the precession, recall, F1-score, 98% metrics with a significant number of parametric values entered. **Conclusions**. When implementing various search methodologies for phishing portals, it is necessary to take into account their decrease in efficiency from the entered parameters. To do this, it is important to conduct preliminary tests. However, the test result can be interpreted in different ways. In particular, the effectiveness of the methods can be improved if you limit the number of input parameters, but at the same time rigidly structured according to one search criterion.

Keywords: phishing, machine learning, neural networks, portal analysis

For citation: E.A. Trushnikov. Detection of phishing portals through machine learning algorithms. Herald of Daghestan State Technical University. Technical Sciences. 2024; 51(3):154-162. DOI:10.21822/2073-6185-2024-51-3-154-162.

Введение. Искусственный интеллект (ИИ) представляет собой ни что иное, как прогрессивный метод имитации способностей человеческого мозга в программной среде. Одной из важных функций данной системы является способность обучаться, иначе именуемая машинным обучением (МО). К сожалению, в связи с все возрастающими угрозами кибербезопасности (КБ), человек все менее способен защититься от киберугроз (КУ), для того он и стал применять возможности ИИ. Более, того, многие специалисты уже не раз утверждали, что ИИ в некоторых аспектах даже лучше справляется с защитой от КУ, чем люди, т.к. при верном обучении подмечает многие закономерности, которые не заметны с первого взгляда.

Вообще, говоря об уже упоминавшемся МО, нельзя не отметить, что оно основывается на попытке интерпретировать сложные закономерности, которые свойственны конкретному виду КУ. Так как требования к КБ все возрастают, способы применения ИИ также активно развиваются, однако до создания идеальной модели искусственного мозга конечно далеко, в первую очередь из-за особенностей технологического развития компьютерных систем и применимости (для полной аналогизации необходимо использовать не виртуальные, а физические ядра, что крайне удорожает аппаратную часть).

Постановка задачи. Одной из главных угроз современной КБ является фишинг. Как известно, это подставные порталы, максимально копирующие реальные, с одним лишь исключением – направленностью на получение конфиденциальных данных пользователей, их средств [1]. Более того, статистика показывает, что количество подобных интернет ресурсов постоянно растет. Это подтверждает статистика с портала Statista, рис. 1 [2]. Как можно заметить уже в 2022 году было обнаружено огромное количество новых фишинг порталов, дальнейший рост будет значителен.

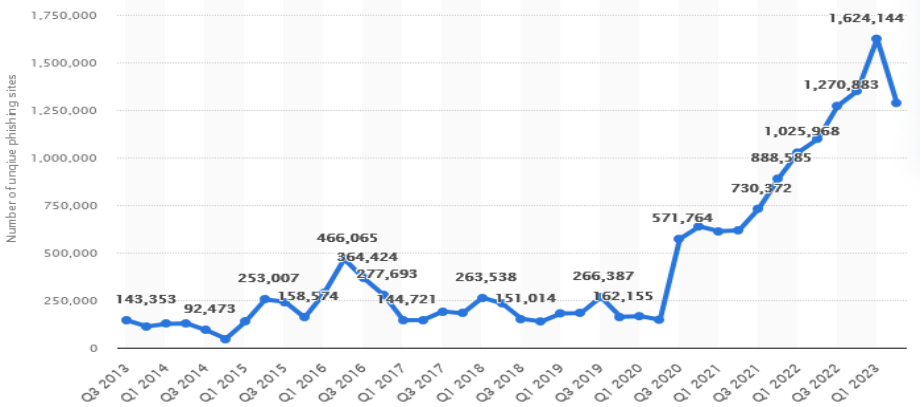


Рис.1. Статистика фишинг порталов по годам
Fig. 1. Phishing portal statistics by year

С целью максимизации информирования о средствах защиты от фишинговых порталов, доказательства важности применения ИИ для их обнаружения, а также выявления оправданных методик, данная работа и была проведена.

Методы исследования. Машинное обучение, как было отмечено ранее, получает информацию извне. Основные способы наполнения – методы: математической статистики, численные, оптимизации, графов и т.п. Важно также заметить, что по своему характеру, МО – это не только математическая часть, но и уже давно сформировавшаяся инженерная дисциплина, т.е. носящая сугубо практический характер.

Основными методами обучения выступают два классических способа: с учителем, без учителя.

Как несложно догадаться, в первом случае при процессе познания задается пара «прецендент-решение», т.е. вывод на основе полученных данных. Второй случай более сложный, т.к. закладывается логика отсутствия верных известных решений, т.е. алгоритм самостоятельно должен кластеризовать получаемые извне сведения, определить сходные черты и дать ответ [3].

Однако не стоит ограничиваться лишь классическим разбиением, а необходимо описать полную картину. Важно отметить такое МО, как полуконтролируемое, т.е. нечто среднее между озвученными ранее. Такой подход весьма эффективен, так как способен адаптироваться к различным ситуациям, что очень полезно для КБ.

Другой немаловажный подвид – МО с подкреплением. Механизм имеет жесткое предназначение – требования к точности выполнения алгоритма. По сути, в него закладывается логика реализации многоэтапных функциональных действий. Важно понимать, что при всей жесткости контроля, это лишь начальный этап, в основном такие МО сами определяют какие шаги должны выполняться [4].

Отдельную касту занимают методы интеллектуального анализа. Среди них выделяются: линейная и логическая регрессии, решающие деревья, случайный лес, а также опорные вектора. Дадим каждому из них краткую характеристику.

Линейная регрессия выводит гипотетический ответ на базе входящих признаков. Тем самым анализ направлен на попытку предсказания целевого значения вошедших характеристических зависимостей. Как и следует догадаться, идеологически описательный график будет из себя представлять линию, проходящую по усредненному значению случившихся событий, тем самым задавая направленность.

Логическая регрессия, в отличие от предыдущей, уже использует несколько видоизмененный подход – применяется теория вероятности. По сути, более рационально было бы охарактеризовать его как прогнозирующий бинарные зависимости входных признаков, т.к. именно для этого он чаще всего и применяется.

Метод «Решающие деревья» – это ни что иное, как группа МО, основной целью которой служит определение целевой переменной при помощи алгоритмизации действий. Причем в алгоритме все действия относятся к простейшим (предикатам). Данный метод предназначается для решения регрессионных и классификационных задач КБ.

Случайный лес – от части напоминает логическую регрессию, т.к. предназначается для определение классификационных признаков, но в тоже время реализуется кардинально иначе. В его основе лежит понятие случайности признаков. Эти признаки необходимы для построения древа. Так как таких древ будет достаточно много, появляется некорреляционное объединение – лес. Именно оно гарантирует наивысшую точность, чем у отдельно взятого древа. Имея большое разнообразие, выходит ансамбль методов. Среди них производится выбор и те, что имеют наибольшее количество становятся прогнозом.

И последний из названных – метод опорных векторов. По своей сути многофункционален, т.к. предназначается как для анализа регрессий, так и для определения классифицирующих характеристик. Априори он принадлежит к классу обучения с помощью учителя и линейным методам.

Для определения используются несколько объединенных множеств, в которые входят параметры. Эти множества разделены при помощи гиперплоскости, строящейся по некоторым заданным законам. Их логика сводится к попытке максимизации расстоя-

ния до так называемых опорных параметров множеств. Данное действие необходимо для максимизации результирующей классификационной характеристики.

После определения методологий классификаций по принадлежности к оригинальным / фишинг порталам, необходимо акцентировать внимание на базах данных (БД), из которых можно подчерпнуть основные параметры для МО. Среди них наиболее популярными стоит считать:

1 Google Dataset Search,
2 Visual Data Discovery (VDD),
3 Kaggle,
4 DataGov.

БД от компании Google не спроста занимает первое место среди перечисленных. Вследствие своей распространенности и известности, она имеет огромный набор сведений, по которым можно проводить МО. Более подробно показано в [5]. Второй представитель БД для МО обладает достаточно большой медиа базой, что крайне положительно сказывается при обучении ИИ на исследование фишинг страниц. Общие сведения по VDD представлены в [6]. Kaggle, в отличие от предыдущих, БД более специализирована именно на МО. Главной особенностью данного портала следует считать возможность живого общения и обмена сведениями со специалистами области КБ, тем самым увеличивая и так немалую БД. Больше сведений показано в [7]. И последний, но не менее значимый, - DataGov. На данном портале сохранены статистические сведения со значимых интернет-ресурсов госструктур. Кроме того, на портале представлены также сведения, явно связанные с работой госаппрата, причем как делового характера, так и опросного [8-9].

Так как дана достаточно подробная характеристика МО, перейдем к его значимому подвиду – нейронным сетям. Поскольку искусственные нейронные сети (ИНС) это алгоритмы МО, наиболее подражающие работе головного мозга, покажем их истинную сущность.

Как известно, нейронная клетка состоит из двух основных отростков – дендритов и аксонов, они являются основополагающими при получении сигналов со стороны синапсиса. Роль же аксонов сводится к выводу информации [3,10]. Аналогично действуют и ИНС, при составлении матмодели, синапсис заменяется весовым коэффициентом [10].

Обучение ИНС проводится определенным алгоритмом действия, включающем следующие четыре этапа:

1) Инициализация весовых коэффициентов в рандомном порядке.
2) Поиск функционального параметрического значения узловой активации.
3) Расчет функционального параметрического значения стоимости.
4) Определение частных производных и проведение коррекции весовых коэффициентов [11]

Перейдем далее к рассмотрению архитектурных особенностей ИНС.

Первым подвидом является ИНС прямого распространения. По своей сущности – это многоуровневое матричное распределение входных, исходящих, а также скрытых значений нейронов. Взаимосвязь нейронов происходит с другими столбцами и никогда не пересекаются со своими. Обучение такой ИНС, как правило, происходит при помощи метода обратного распространения ошибок. Наличие скрытых нейронов также имеет огромное значение. Именно они позволяют проводить процесс моделирования [12].

Другим вариантом ИНС можно назвать сеть Хопфилада. Она характеризуется симметричной матрицей связи нейронов. При поступлении сигнала со связанного нейрона, узлы ИНС переходят в статус скрытых, после чего переквалифицируется в исходящий, тем самым описывается процесс МО. Для составления сети Хопфилда необходимо определить весовые коэффициенты, желаемый поведенческий шаблон или шаблоны, если таких несколько. Нередко данную ИНС называют ассоциативной, т.к. она проводит логический анализ по восстановлению матричного поведения [13].

Говоря про виды ИНС, нельзя обойти вниманием сверточные сети. Несмотря на направленность в первую очередь на графические материалы и определение классификационных признаков, она может также применяться для анализа аудиопотока. Данные ИНС парсят все имеющиеся сведения за один прогон [14]. Рассматривая ее с точки зрения входящего матаппарата, сложно отметить присутствие операций свертки, что позволяет не задействовать все части нейронной системы, как это было в классическом варианте [15-16]. Более того, адаптивность данной системы позволяет ей объединяться с иными ИНС, как это отмечено в [17].

И последняя значимая группа ИНС – развертывающиеся сети. Как можно догадаться, принцип их работы противоположен предыдущему варианту. Важно отметить, что вне зависимости от выбора нейронной сети, для запуска работы ИНС, необходимо воздействие, иначе его еще называют активационной функцией. Пример наиболее распространенных показан в [18].

Тем самым, дав описание МО, показав особенности их реализации в виде ИНС. Можно перейти непосредственно к вопросу защиты от фишинговых атак.

Обсуждение результатов. Лучшее средство КБ от фишинга – программа-анализатор. Для ее МО необходимо в обязательном порядке знание верных доменов, а также примеры фишинговых порталов. В качестве основы обучения были приняты методики, показанные в работах [19-21].

Разработка программы поиска фишинговых порталов велась по логике, отображенной в блок-схемном представлении, показанном на рис. 2.

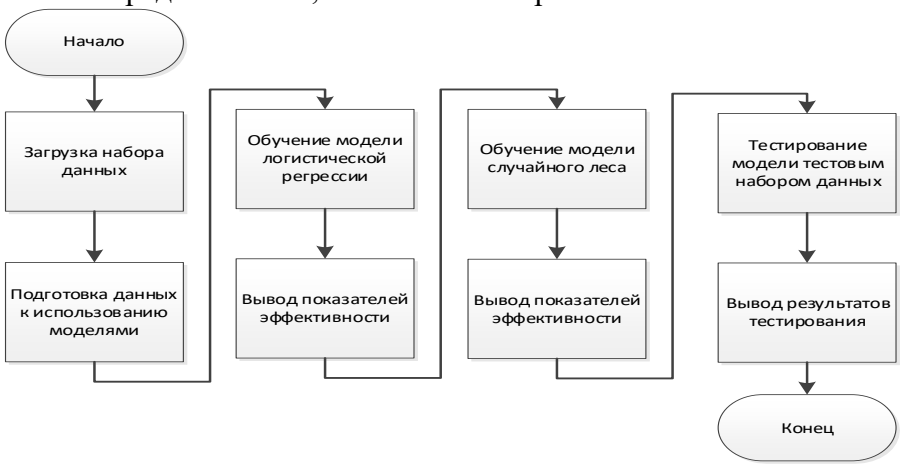


Рис. 2. Блок-схема работы программы
Fig. 2. Block diagram of the program operation

В качестве тестовых, были применены датасеты Чун Лин Тана, при этом специально выбирались 10000 вариационных примеров, которые отдельно классифицировались на фальшивые / оригинальные.

Характеристические признаки подробно показаны в [22].

Для обеспечения работоспособности ИНС были подключены следующие библиотеки из среды программирования Python: Pandas, Numpy, Matplotlib, Seaborn.

Первая библиотека предназначается для управления БД, вторая – для матвычислений, в т.ч. для работ с многомерными матрицами и массивами данных [23]. Matplotlib предназначается для графического отображения, т.е. занято со средой визуализации [24], и, последняя служит для интеграции со структурами данных [25]

Данные для МО в программу были загружены посредством функции read_csv – рис. 3(а), а после преобразования, информация стала иметь новый вид – рис. 3(б). Преобразованию было необходимо с целью минимизации затрачиваемых ресурсов памяти (float64, int64 в float32, int23).

Обучающие наборы были равномерно распределены. В дальнейшем обучающая информация была распределена по классам и решалась задача определения признаков, необходимых для непосредственного МО.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 50 columns):
id                                  10000 non-null int64
NumDots                             10000 non-null int64
SubdomainLevel                      10000 non-null int64
PathLevel                           10000 non-null int64
UrlLength                           10000 non-null int64
NumDash                             10000 non-null int64
NumDashInHostname                   10000 non-null int64
AtSymbol                            10000 non-null int64
TildeSymbol                         10000 non-null int64
NumUnderscore                       10000 non-null int64
NumPercent                          10000 non-null int64
NumQueryComponents                  10000 non-null int64
NumAmpersand                        10000 non-null int64
NumHash                             10000 non-null int64
NumNumericChars                     10000 non-null int64
NoHttps                             10000 non-null int64
RandomString                        10000 non-null int64
IpAddress                           10000 non-null int64
DomainInSubdomains                  10000 non-null int64
DomainInPaths                       10000 non-null int64
HttpsInHostname                     10000 non-null int64
HostnameLength                      10000 non-null int64
PathLength                          10000 non-null int64
QueryLength                         10000 non-null int64
DoubleSlashInPath                   10000 non-null int64
NumSensitiveWords                   10000 non-null int64
EmbeddedBrandName                   10000 non-null int64
PctExtHyperlinks                    10000 non-null float64
PctExtResourceUrls                  10000 non-null float64
ExtFavicon                          10000 non-null int64
InsecureForms                       10000 non-null int64
RelativeFormAction                  10000 non-null int64
ExtFormAction                       10000 non-null int64
AbnormalFormAction                  10000 non-null int64
PctNullSelfRedirectHyperlinks       10000 non-null float64
```

а

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 50 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              10000 non-null  int32
 1   NumDots                         10000 non-null  int32
 2   SubdomainLevel                  10000 non-null  int32
 3   PathLevel                       10000 non-null  int32
 4   UrlLength                       10000 non-null  int32
 5   NumDash                         10000 non-null  int32
 6   NumDashInHostname               10000 non-null  int32
 7   AtSymbol                        10000 non-null  int32
 8   TildeSymbol                     10000 non-null  int32
 9   NumUnderscore                   10000 non-null  int32
 10  NumPercent                      10000 non-null  int32
 11  NumQueryComponents              10000 non-null  int32
 12  NumAmpersand                    10000 non-null  int32
 13  NumHash                         10000 non-null  int32
 14  NumNumericChars                 10000 non-null  int32
 15  NoHttps                         10000 non-null  int32
 16  RandomString                    10000 non-null  int32
 17  IpAddress                       10000 non-null  int32
 18  DomainInSubdomains              10000 non-null  int32
 19  DomainInPaths                   10000 non-null  int32
 20  HttpsInHostname                 10000 non-null  int32
 21  HostnameLength                  10000 non-null  int32
 22  PathLength                      10000 non-null  int32
 23  QueryLength                     10000 non-null  int32
 24  DoubleSlashInPath               10000 non-null  int32
 25  NumSensitiveWords               10000 non-null  int32
 26  EmbeddedBrandName               10000 non-null  int32
 27  PctExtHyperlinks                10000 non-null  float32
 28  PctExtResourceUrls              10000 non-null  float32
 29  ExtFavicon                      10000 non-null  int32
 30  InsecureForms                   10000 non-null  int32
 31  RelativeFormAction              10000 non-null  int32
 32  ExtFormAction                   10000 non-null  int32
 33  AbnormalFormAction              10000 non-null  int32
 34  PctNullSelfRedirectHyperlinks   10000 non-null  float32
```

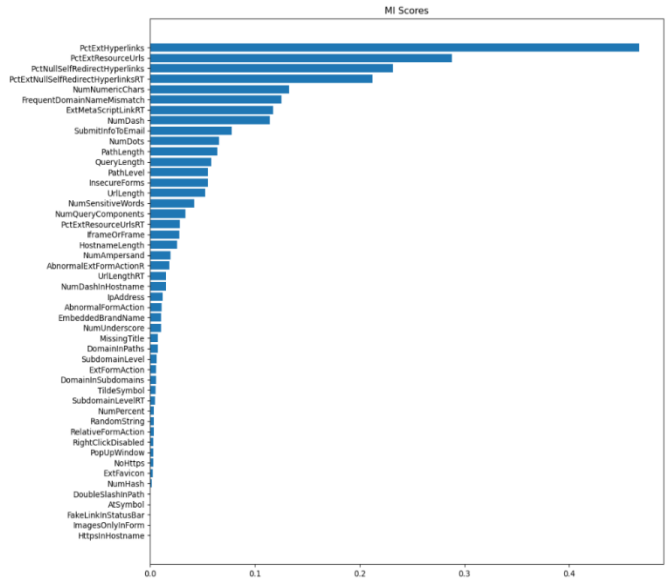
б

Рис. 3. Фрагмент переданной (а), преобразованной (б) информации
Fig. 3. Fragment of transmitted (a), transformed (b) information

Кроме того, проводился поиск линейных / нелинейных зависимостей между выведенными классами и полученными признаками. При помощи классификатора «mutual_info_classif», описанного в работе [26] были получены результаты, рис. 4(а), и структурированы – рис. 4(б).

```
PctExtHyperlinks                  4.686999e-01
PctExtResourceUrls                2.968365e-01
PctNullSelfRedirectHyperlinks     2.342147e-01
PctExtNullSelfRedirectHyperlinksRT 2.123850e-01
NumNumericChars                   1.326711e-01
FrequentDomainNameMismatch        1.254908e-01
ExtMetaScriptLinkRT               1.175434e-01
NumDash                           1.140136e-01
SubmitInfoToEmail                 7.802617e-02
NumDots                           6.553700e-02
PathLength                        6.413269e-02
QueryLength                       5.818557e-02
PathLevel                         5.543887e-02
InsecureForms                     5.508096e-02
UrlLength                         5.263855e-02
NumSensitiveWords                 4.180522e-02
NumQueryComponents                3.361768e-02
PctExtResourceUrlsRT              2.831059e-02
IframeOrFrame                     2.805496e-02
HostnameLength                    2.564249e-02
NumAmpersand                      1.962894e-02
AbnormalExtFormActionR            1.816610e-02
UrlLengthRT                       1.513853e-02
NumDashInHostname                 1.502933e-02
IpAddress                         1.207265e-02
AbnormalFormAction                1.118934e-02
EmbeddedBrandName                 1.066208e-02
NumUnderscore                     1.037366e-02
MissingTitle                      7.353661e-03
DomainInPaths                     7.003483e-03
SubdomainLevel                    6.392610e-03
ExtFormAction                     5.613946e-03
DomainInSubdomains                5.497419e-03
TildeSymbol                       5.362924e-03
SubdomainLevelRT                  4.503644e-03
NumPercent                        3.790407e-03
RandomString                      3.643235e-03
```

а



б

Рис. 4. Коэффициенты корреляции признаков и их влияние
Fig. 4. Correlation coefficients of features and their influence

МО проводилось по следующему принципу – 70% набор данных обрабатывался с целью формирования модели, наиболее точно совпадающей с целевым набором. Тестовый набор, 30%, направлен на получение объективной оценки сформированной модели, в него не входили данные из обучающего набора. Далее, зная, что процесс поиска фишинговых порталов в чистом виде относится к классификационному анализу, была проведена оценка по принципам логистической регрессии и случайного леса.

Для этого применялись метрики, определялись классы. Логика полностью совпадает с показанной в [26] (применена библиотека scikit-learn). Метрики, соответственно – precision (точность) и recall (полнота) и F1-score (среднее гармоническое первых двух). Покажем для каждой модели полученные показатели эффективности – рис. 5-6.

Первоначально, на что следует обратить внимание при анализе полученных графиков, – большая эффективность второго метода. Однако, нельзя категорически относиться и к первому подходу, просто необходимо учесть некоторые особенности, которые были выявлены в результате моделирования.

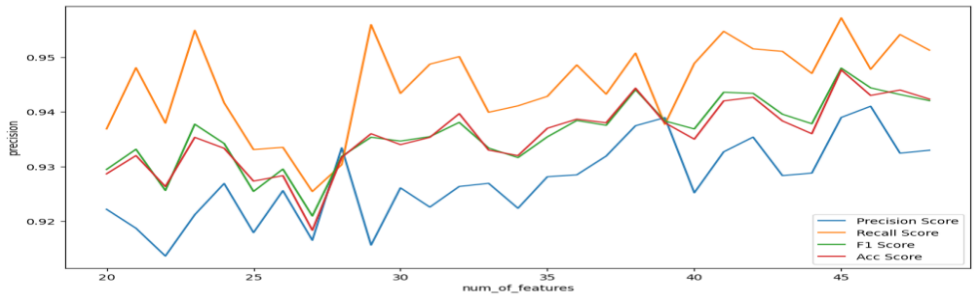


Рис. 5. Показатели логистической регрессии
Fig. 5. Logistic regression indicators

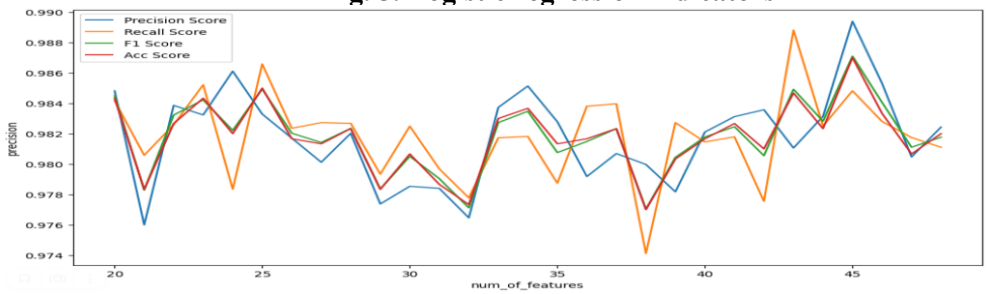


Рис. 6. Показатели случайного леса
Fig. 6. Random Forest Performance

В частности, опыт показал, что при увеличении количества признаков, постепенно теряется параметр качества поиска. Наиболее оптимальное значение находится в районе 39 характеристических параметров, тогда точность падала до 93%. Аналогичное ухудшение, в реальности, видно и в случайном лесе, просто снижение эффективности намного меньше и не так явно выражено. В целом, по проведенному исследованию можно сразу отметить тот факт, что даже при 32 характеристических признаках, расчетное качество модели последнего вида анализатора составляла 98%. Тестовый набор эксперимента показал 100% выявление фишинговых и подставных порталов. Это показано на рис. 7.

	precision	recall	f1-score	support
0	0.98	0.98	0.98	1500
1	0.98	0.98	0.98	1500
accuracy			0.98	3000
macro avg	0.98	0.98	0.98	3000
weighted avg	0.98	0.98	0.98	3000

Рис. 7. Тестирование случайного леса
Fig. 7. Testing Random Forest

Вывод. Проведенное исследование – обзорно-теоретическое. В работе проводился анализ понятия МО, ИИ, нейронных сетей, их видов, а также тестировались некоторые подходы по определению фишинговых порталов. Анализ показал, что имеется достаточно большое количество обобщенных методик обучения ИИ.

Главной же особенностью, на которую следует обратить внимание – это практические способы, к которым относятся линейная и логическая регрессии, решающие деревья, случайный лес, опорные вектора. Именно с помощью отмеченных подходов можно обучить ИИ на поиск фишинговых порталов.

Для реализации ИИ были определены четыре необходимых. Так как имеется несколько видов архитектур, они были также в работе рассмотрены. Это ИНС прямого распространения, сеть Хопфилада, сверточные и развертывающиеся сети.

В части практической реализации была использована программная среда Python. С помощью подключенных баз были загружены 10000 сведений порталам, проводилось распределение по классам и поиск линейных / нелинейных зависимостей. Далее была проведена оценка по принципам логистической регрессии и случайного леса. Опыт показал, что случайный лес более качественен в своем анализе.

Библиографический список:

1. Завьялов А.Н. Интернет-мошенничество (фишинг): проблемы противодействия и предупреждения // Baikal Research Journal. - 2022. - Т. 13. - № 2. - С20-26
2. Number of unique phishing sites detected worldwide from 3rd quarter 2013 to 2nd quarter 2023. - URL: https://www.statista.com/statistics/266155/number-of-phishing-domain-names-worldwide/ (дата обращения 21.02.2024).
3. Платонов А.В. Машинное обучение : учебное пособие для вузов. М.: Издательство Юрайт, 2023. – 85 с.
4. Techtarget. Machine learning. - URL: https://searchenterpriseai.techtarget.com/ definition/machine-learning-ML (дата обращения 21.12.2022).
5. Google Dataset Search. - URL: https://datasetsearch.research.google.com/ (дата обращения: 25.12.2022)
6. Visual Data Discovery. - URL: https://visualdata.io/discovery (дата обращения: 25.12.2022)
7. Kaggle. - URL: https://www.kaggle.com/ (дата обращения: 25.12.2022)
8. Портал открытых данных Российской Федерации. – URL: https://data.gov.ru/opendata (дата обращения: 25.12.2022)
9. Воронова Л.И., Брус В.Р., Воронов В.И., Баширов А.Н. Предобработка данных для нейросетевого управления: учебное пособие. – М.: МТУСИ, 2021. - 44 с.
10. Потапкин К.О. Искусственные нейронные сети. Нейронная сеть // XLVI Огарёвские чтения : Материалы научной конференции: В 3-х частях, Саранск, 06–13 декабря 2017 года / Ответственный за выпуск П.В. Сенин. Том Часть 1. – Саранск: Национальный исследовательский Мордовский государственный университет им. Н.П. Огарёва, 2018. – С. 315-320.
11. Воронова Л.И., Воронов В.И. Machine Learning: регрессионные методы интеллектуального анализа данных : учеб. пособ.. М.: Московский технический университет связи и информатики, 2018. - 82 с.
12. The perceptron: a probabilistic model for information storage and organization in the brain. - URL: https://www.ling.upenn.edu/courses/cogs501/Rosenblatt1958.pdf (дата обращения: 21.12.2022)
13. Neural networks and physical systems with emergent collective computational abilities. - URL: https://bi.snu.ac.kr/Courses/g-ai09-2/hopfield82.pdf (дата обращения: 21.12.2022)
14. Разновидности нейронных сетей. - URL: https://tproger.ru/translations/neural-network-zoo-1/ (дата обращения: 21.12.2022)
15. Что такое нейронные сети, что они могут. - URL: https://neural-university.ru/neural-networks-basics (дата обращения: 20.12.2022)
16. Нейронные сети: распознавание образов и изображений с помощью ИИ. - URL: https://center2m.ru/ai-recognition (дата обращения: 20.12.2022)
17. Маркова С.В., Жигалов К.Ю. Применение нейронной сети для создания системы распознавания изображений // Фундаментальные исследования. – 2017. – № 8-1. – С. 60-64
18. Как работает нейронная сеть: алгоритмы, обучение, функции активации и потери. - URL: https://neurohive.io/ru/osnovy-data-science/osnovy-nejronnyh-setej-algoritmy-obuchenie-funkcii-aktivacii-i-poteri/
19. Xuan C.D. Nguyen H.D., Nikolaevich T.V. Malicious URL detection based on machine learning // International Journal of Advanced Computer Science and Applications. – 2020. – Vol. 11. – No 1. – P.148-153
20. Manjeri A.S., Kaushir R., Ajay M.N.V., Nair P.C. A Machine Learning Approach for Detecting Malicious Websites using URL Features // 3rd International Conference on Electronics, Communication and Aerospace Technology(ICECA), 12-14 June 2019. - Coimbatore, India. 555–561 pp.
21. Sarasjati W. et al., Comparative Study of Classification Algorithms for Website Phishing Detection on Multiple Datasets // International Seminar on Application for Technology of Information and Communication (iSemantic). - Semarang, Indonesia, 2022. - 448-452 pp.
22. A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. - URL: https://www.sciencedirect.com/science/article/abs/pii/ S0020025519300763?via%3Dihub (дата обращения: 21.12.2022)
23. Учебник по библиотеке NumPy: учитесь на примерах. - URL: https://pythonist.ru/ uchebnik-po-biblioteke-numpy-uchites-na-primerah/ (дата обращения: 27.12.2022)
24. Руководство по Matplotlib. - URL: https://indico-hlit.jinr.ru/event/151/attachments/ 340/492/Project_school_Matplotlib_original.pdf (дата обращения: 27.12.2022)
25. Seaborn для визуализации данных в Python. - URL: https://pythonru.com/biblioteki/ seaborn-plot (дата обращения: 27.12.2022)
26. Scikit-learn Machine Learning in Python. - URL: https://scikit-learn.org/ (дата обращения: 21.12.2022)

References:

1. Zav'yalov A.N. Internet-moshennichestvo (fishing): problemy protivodejstviya i preduprezhdeniya. *Baikal Research Journal.* 2022;13(2):20-26 (In Russ)
2. Number of unique phishing sites detected worldwide from 3rd quarter 2013 to 2nd quarter 2023. - URL: https://www.statista.com/statistics/266155/number-of-phishing-domain-names-worldwide/ (data obrashcheniya 21.02.2024).

3. Platonov A.V. Mashinnoe obuchenie : uchebnoe posobie dlya vuzov M.: Izdatel'stvo Yurajt, 2023; 85 (In Russ)
4. Techtarget. Machine learning. - URL: https://searchenterpriseai.techtarget.com/ definition/machine-learning-ML (data obrashcheniya 21.12.2022).
5. Google Dataset Search. - URL: https://datasetsearch.research.google.com/ (data obrashcheniya: 25.12.2022)
6. Visual Data Discovery. - URL: https://visualdata.io/discovery (data obrashcheniya: 25.12.2022)
7. Kaggle. - URL: https://www.kaggle.com/ (data obrashcheniya: 25.12.2022)
8. Portal otkrytyh dannyh Rossijskoj Federacii. – URL: https://data.gov.ru/opendata (data obrashcheniya: 25.12.2022)
9. Voronova L.I., Brus V.R., Voronov V.I., Bashirov A.N. Predobrabotka dannyh dlya nejrosetevogo upravleniya: uchebnoe posobie. M.: MTUSI, 2021; 44. (In Russ)
10. Potapkin K.O. Iskusstvennye nejronnye seti. Nejronnaya set' // XLVI Ogaryovskie chteniya : Materialy nauchnoj konferencii: V 3-h chastyah, Saransk, 06–13 dekabrya 2017 goda / Otvetstvennyj za vypusk P.V. Senin. Tom Chast' 1. – Saransk: Nacional'nyj issledovatel'skij Mordovskij gosudarstvennyj universitet im. N.P. Ogaryova, 2018;315-320. (In Russ)
11. Voronova L.I., Voronov V.I. Machine Learning: regressionnye metody intellektual'nogo analiza dannyh : uchebnoe posobie. – M.: Moskovskij tekhnicheskij universitet svyazi i informatiki, 2018; 82. (In Russ)
12. The perceptron: a probabilistic model for information storage and organization in the brain. - URL: https://www.ling.upenn.edu/courses/cogs501/Rosenblatt1958.pdf (data obrashcheniya: 21.12.2022)
13. Neural networks and physical systems with emergent collective computational abilities. - URL: https://bi.snu.ac.kr/Courses/g-ai09-2/hopfield82.pdf (data obrashcheniya: 21.12.2022)
14. Raznovidnosti nejronnyh setej. - URL: https://tproger.ru/translations/neural-network-zoo-1/ (data obrashcheniya: 21.12.2022)
15. Chto takoe nejronnye seti, chto oni mogut. - URL: https://neural-university.ru/neural-networks-basics (data obrashcheniya: 20.12.2022)
16. Nejronnye seti: raspoznavanie obrazov i izobrazhenij c pomoshch'yu II. - URL: https://center2m.ru/ai-recognition (data obrashcheniya: 20.12.2022)
17. Markova S.V., Zhigalov K.Yu. Primenenie nejronnoj seti dlya sozdaniya sistemy raspoznavaniya izobrazhenij. *Fundamental'nye issledovaniya*. 2017;8-16 60-64(In Russ)
18. How a neural network works: algorithms, training, activation functions and losses. - URL: https://neurohive.io/ru/osnovy-data-science/osnovy-nejronnyh-setej-algoritmy-obuchenie-funkcii-aktivacii-i-poteri/
19. Xuan C.D. Nguyen H.D., Nikolaevich T.V. Malicious URL detection based on machine learning. I*nternational Journal of Advanced Computer Science and Applications*. 2020; 11(1):148-153
20. Manjeri A.S., Kaushir R., Ajay M.N.V., Nair P.C. A Machine Learning Approach for Detecting Malicious Websites using URL Features // 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA), 12-14 June 2019; 555–561. Coimbatore, India.
21. Sarasjati W. et al., Comparative Study of Classification Algorithms for Website Phishing Detection on Multiple Datasets. *International Seminar on Application for Technology of Information and Communication (iSemantic).* Semarang, Indonesia, 2022; 448-452.
22. A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. - URL: https://www.sciencedirect.com/science/article/abs/pii/ S0020025519300763?via%3Dihub (data obrashcheniya: 21.12.2022)
23. Uchebnik po biblioteke NumPy: uchites' na primerah. - URL: https://pythonist.ru/ uchebnik-po-biblioteke-numpy-uchites-na-primerah/ (data obrashcheniya: 27.12.2022)
24. Rukovodstvo po Matplotlib. - URL: https://indico-hlit.jinr.ru/event/151/attachments/ 340/492/Project_school_Matplotlib_original.pdf (data obrashcheniya: 27.12.2022)
25. Seaborn dlya vizualizacii dannyh v Python. - URL: https://pythonru.com/biblioteki/ seaborn-plot (data obrashcheniya: 27.12.2022)
26. Scikit-learn Machine Learning in Python. - URL: https://scikit-learn.org/ (data obrashcheniya: 21.12.2022)

Сведения об авторе:
Трушников Евгений Александрович**,** магистрант; udonem@mail.ru
Information about the author:
Evgenij A. Trushnikov, Graduate student; udonem@mail.ru
Конфликт интересов/Conflict of interest.
Автор заявляет об отсутствии конфликта интересов/The author declare no conflict of interest.
Поступила в редакцию/ Received 26.02.2024.
Одобрена после рецензирования/ Reviced 02.04.2024.
Принята в печать/ Accepted for publication 02.04.2024.