УДК 004.8

Выявление потенциально вредоносных постов в социальных сетях с помощью обучения на основе положительных и неразмеченных текстовых данных

Тушканова О. Н.

Постановка задачи: *быстрый рост аудитории социальных сетей, а также невозможность качественной ручной модерации их контента требуют разработки автоматических методов выявления потенциально вредоносной информации, основанных на машинном обучении, для защиты уязвимых групп населения от такой информации. Распространенным методом выявления потенциально вредоносных текстов в социальных сетях является классификация, однако известные алгоритмы, основанные на применении классической классификации, часто не демонстрируют достаточной стабильности и точности при наличии ложноотрицательных примеров в обучающей выборке.* ***Целью работы*** *является оценка применимости подхода к машинному обучению на основе положительных и неразмеченных данных в задаче выявления вредоносных постов в социальных сетях и разработка алгоритма, реализующего этого подход, для повышения точности идентификации вредоносных текстов.* ***Используемые методы****: в разработанном алгоритме предлагается использовать метод машинного обучения классификаторов на основе положительных и неразмеченных данных, а также подход многоклассовой классификации.* ***Новизна:*** *элементами новизны представленного решения является объединение подхода к обучению классификатора на основе положительных и неразмеченных данных и классического многоклассового классификатора в рамках разработанного алгоритма.* ***Результат****: обоснована целесообразность применения подходов к машинному обучению на основе положительных и неразмеченных данных в задаче выявления текстов, содержащих потенциально вредоносную информацию, и рассмотрены особенности некоторых таких подходов. Предложен алгоритм выявления потенциально вредоносных постов в социальных сетях с помощью машинного обучения на основе положительных и неразмеченных на текстовых данных и многоклассовой классификации. Представлен сценарий и результаты экспериментального исследования предложенного алгоритма на выборке текстовых постов социальной сети Вконтакте. Показано, что разработанный алгоритм работает стабильнее (с точки зрения оценок точности) традиционного подхода многоклассовой классификации при наличии ложноотрицательных примеров в обучающих данных.* ***Практическая значимость:*** *результаты исследования использованы при разработке системы выявления вредоносного контента в социальных сетях. Предложенный подход позволяет повысить точность классификации вредоносных текстов при наличии в обучающих данных ложноотрицательных примеров.*

Ключевые слова: *машинное обучение, PU-обучение, многоклассовая классификация, вредоносная информация, социальная сеть.*

Введение

Существует множество подходов к обнаружению вредоносной информации в Интернете, большая часть которых, так или иначе, использует алгоритмы машинного обучения и другие методы из области искусственного интеллек-

Библиографическая ссылка на статью:
Тушканова О. Н. Выявление потенциально вредоносных постов в социальных сетях с помощью обучения на основе положительных и неразмеченных текстовых данных // Системы управления, связи и безопасности. 2021. № 6. С. 30-52. DOI: 10.24412/2410-9916-2021-6-30-52.

Reference for citation:
Tushkanova O. N. Identification of potentially malicious posts on social networks using positive and unlabeled learning on text data. *Systems of Control, Communication and Security*, 2021, no. 6, pp. 30-52 (in Russian). DOI: 10.24412/2410-9916-2021-6-30-52.

та [1, 2]. Часто эти подходы применяются к контенту социальных сетей и связаны с защитой от информации.

Профиль пользователя социальной сети обычно представляет собой набор разнородных гетерогенных данных, таких как история активности пользователя, анкетные данные, фотографии, видео, данные о новостной ленте пользователя и его предпочтениях. Но именно текстовые посты, репосты и комментарии пользователя социальной сети представляют, по всей видимости, наибольшую ценность для анализа потенциально вредоносных постов.

Результаты такого анализа могут быть, в том числе, полезны правоохранительным органам для выявления мошенников и создателям сервисов социальных сетей для выявления ботов.

Далее в работе обоснована целесообразность применения подходов к машинному обучению на основе положительных и неразмеченных данных в задаче классификации выявления текстов применительно к защите от потенциально вредоносной информации в социальных сетях (раздел 1) и рассмотрены особенности некоторых таких подходов (раздел 2). Предложен алгоритм выявления потенциально вредоносных постов в социальных сетях с помощью машинного обучения на основе положительных и неразмеченных на текстовых данных (раздел 3), а также представлены дизайн и результаты экспериментального исследования предложенного алгоритма на выборке текстовых постов социальной сети Вконтакте (раздел 4).

В работе показано, что разработанный алгоритм работает стабильнее (с точки зрения оценок точности) традиционного подхода многоклассовой классификации при добавлении ложноотрицательных примеров в обучающие данные.

1. Основные проблемы классического подхода к классификации с точки зрения защиты от информации

Одним из наиболее распространенных методов, применяемых для выявления потенциально вредоносных текстов в социальных сетях, является классическая классификация.

Опишем задачу классификации в классической постановке, как это принято в области машинного обучения. Пусть некоторое множество объектов (называемых также экземплярами или записями) $\mathbf{X} = \{\boldsymbol{x}_i\}|^n_{i=1}$ характеризуется вектором значений их признаков $\boldsymbol{x}_i = \{\boldsymbol{x}_i^1, \ldots, \boldsymbol{x}_i^d\}$, где d – это общее количество признаков. Каждому объекту из этого множества, $\boldsymbol{x}_i \in \mathbf{X}$, можно поставить в соответствие значение y_i переменной y, которое называют меткой класса или просто классом объекта $\boldsymbol{x}_i$. Переменная y может принимать конечное число значений из некоторого множества $\mathbf{Y} = \{y_1, \ldots, y_m\}$. В зависимости от мощности множества $\mathbf{Y}$, задача классификации может быть унарной или одноклассовой, если множество содержит ровно один элемент, бинарной, если элементов в множестве ровно два, или многоклассовой, если множество $\mathbf{Y}$ содержит любое конечное количество элементов больше двух. При этом любая задача многоклассовой классификации легко сводится к бинарной.

Для решения любой задачи классификации необходимо построить алгоритм a, который далее будем также называть классификатором, наиболее «качественно» определяющий по вектору признаков $\boldsymbol{x}_i = \{\boldsymbol{x}_i^1, \ldots, \boldsymbol{x}_i^d\}$ для каждого объекта $\boldsymbol{x}_i \in \mathbf{X}$ метку класса y_i или вектор оценок принадлежности (апостериорных вероятностей) объекта $\boldsymbol{x}_i$ к каждому из классов множества $\mathbf{Y}$:$\{p(y_i \mid \boldsymbol{x}_i)\}|^n_{j=1}$. Таким образом, для решения задачи классификации необходимо построить такой классификатор, который минимизирует ошибку отнесения объектов к классам из множества $\mathbf{Y}$, чаще всего решение этой задачи выполняется путем оптимизации некоторого функционала, который называют функционалом эмпирического риска. Сам процесс построения такого решающего алгоритма называют обучением классификатора.

Для оценки качества обученного классификатора обычно используют метрики аккуратности (англ. accuracy), точности (англ. precision) и полноты (англ. recall) или их комбинацию – F-меру. Аккуратность классификатора - это отношение количества примеров в наборе данных, которое классификатор отнес к верному классу, к общему количеству примеров в наборе данных. Точность классификации – это отношение количества примеров в наборе данных, верно отнесенных классификатором к некоторому классу, к общему количеству примеров, которые классификатор отнес к этому классу. Полнота классификации – это отношение количества примеров, верно отнесенных классификатором к некоторому классу, к количеству примеров этого класса в наборе данных. F-мера – это среднее гармоническое значений точности и полноты. Перечисленные метрики, используемые с целью оценки качества классификации, могут быть рассчитаны на обучающей выборке, $\mathbf{Z} = \{\boldsymbol{x}_i, y_i\}|^n_{j=1}$, и на отложенной или тестовой выборке, примеры из которой не использовались в процессе обучения классификатора.

Предполагается, что данные используемые для обучения классификатора в рамках классической постановки задачи классификации, являются независимой и одинаково распределенной выборкой из некоторого распределения объектов рассматриваемых классов.

Итак, в соответствие с постановкой, приведенной выше, можно сказать, что целью классической задачи бинарной классификации является обучение классификатора, способного различать объекты двух классов, которые обычно называют положительным и отрицательным классами. Для обучения бинарного классификатора алгоритму необходима полностью размеченная обучающая выборка, которая традиционно содержит примеры положительного и отрицательного класса, при этом метка класса не может быть пропущена ни для одного примера обучающих данных. Такой тип обучения классификатора называют обучением с учителем.

К сожалению, в случае решения реальных практических задач, связанных, среди прочего, с защитой от информации, то есть, например, с выявлением потенциально вредоносных текстов, заключается в том, что в имеющемся наборе обучающих данных часто могут отсутствовать надежные размеченные примеры отрицательного класса. Эту проблему можно считать одной из основных

проблем применения традиционных методов классификации к задаче классификации текстов в области защиты от информации.

Как правило, отсутствие надежных отрицательных примеров обусловлено несколькими причинами. Во-первых, отрицательные обучающие примеры должны единообразно представлять универсальный набор всех возможных объектов, не принадлежащих положительному классу. Для многих задач это требование является практически невыполнимым в силу огромного разнообразия объектов отрицательного класса. А во-вторых, собранные вручную отрицательные примеры в данных могут быть необъективными из-за предубеждений человека, выполняющего разметку, что может нанести существенный ущерб точности будущей классификации [3].

Упомянутые проблемы актуальны для множества приложений, в которых пользователь заинтересован в идентификации определенного типа объектов, выступающих в качестве примеров положительного класса, например, текстовых документов.

Рассмотрим пример классификации статей, посещенных некоторой теме. Допустим, в некоторой прикладной задаче необходимо идентифицировать статьи, посвященные теме защиты от информации, среди некоторого конечного множества статей. Пусть для обучения классификатора были размечены статьи, относящиеся к теме защиты от информации (положительный класс), и статьи, не относящиеся к таковым (отрицательный класс), опубликованные в рамках некоторой конференции А. С помощью этих размеченных данных был обучен классификатор, показавший высокую точность на стратифицированном тестовом наборе, также сформированном из статей конференции А. С большой долей вероятности можно утверждать, что такой классификатор будет плохо работать для статей с конференций Б и В. И связано это с тем, что, хотя работы, посвященные защите от информации на этих конференциях аналогичны, статьи на другие темы (и сам набор тем) у всех трех конференций могут сильно отличаться [4].

Другой показательный пример – это обнаружение спама в системах электронной почты. Система классификации спама, построенная с использованием обучающих данных публичного почтового сервер, используемого преимущественно частными лицами, может плохо работать в коммерческих компаниях. Причина в том, что, хотя электронные письма со спамом (например, нежелательные коммерческие объявления) похожи в обеих средах, электронные письма без спама в них могут быть совершенно разными.

Выходом может стать разметка отрицательных примеров для каждой конкретной задачи и темы отдельно. Однако, часто это нецелесообразно или даже невозможно сделать. Например, пусть необходимо классифицировать блоги на некотором хостинге, как относящееся к теме фондовых рынков (положительный класс), и не относящиеся к ней (отрицательный класс). В этом случае отрицательные примеры охватывают произвольный диапазон тем. Разметка всех возможных отрицательных примеров является в этом случае совершенно нерациональной.

Все эти соображения справедливы и для задачи классификации текстов с целью выявления вредоносных постов в социальных сетях. В этом случае реалистичным представляется формирование выборки текстов, которые являются вредоносными с точки зрения пользователя системы (положительный класс), например, тексты, описывающие насилие. Однако совершенно невозможно сформировать выборку с отрицательными примерами, так как она должна включать разнообразные тексты, на все возможные темы, обсуждаемые в социальных сетях.

В отличии от традиционных подходов к бинарной классификации текстов методы обучения классификаторов на основе положительных и неразмеченных данных (англ. positive-unlabeled, далее PU-обучение) могут использовать небольшой набор размеченных примеров положительного класса и набор большего размера, состоящий из неразмеченных примеров, для построения точного классификатора.

Экспериментальные исследования, описанные например в [3, 4], показывают, что в случае значительных отличий распределения отрицательных примеров в обучающих и тестовых данных, PU-обучение показывает более высокую точность, чем традиционное обучение с использованием положительных и отрицательных примеров в обучающих данных. Это означает, что примеры отрицательного класса могут в некоторых случаях навредить обучению классификатора. Кроме того, даже когда распределения обучающей и тестовой выборок с точки зрения отрицательных примеров идентичны, PU-обучение не уступает в точности методам обучения с учителем.

Термин PU-обучение впервые появился в начале 2000-х годов, и в последние годы наблюдается всплеск интереса к этому направлению [5, 6, 7], что вполне соответствует тренду на разработку алгоритмов обучения, которые не требуют полностью размеченных данных, таких как обучение на основе только положительных данных или данных только одного класса (англ. one-class learning) [8] и полу-управляемое обучение (англ. semi-supervised learning) [9]. PU-обучение отличается от названных подходов тем, что оно явно включает неразмеченные данные в процесс обучения.

Одной из причин того, что PU-обучение привлекло внимание исследователей, является то, что PU-данные возникают во многих актуальных на сегодняшний день приложениях.

Далее рассмотрим принципы работы алгоритмов, основанных на подходе обучения классификаторов на основе положительных и неразмеченных данных, а также основные классы таких алгоритмов.

2. Обучение на основе положительных и неразмеченных данных

Как было сказано выше, целью PU-обучения, как и целью классической задачи бинарной классификации, является обучение классификатора, который может разделять объекты положительного и отрицательного класса на основе значений их признаков, но в случае PU на этапе обучения классификатору подаются на вход только некоторые примеры данных положительного класса, и ни одного примера отрицательного класса.

Рассмотрим более формальную постановку задачи PU-обучения. Набор PU-данных для обучения классификатора можно представить как набор триплетов ($\boldsymbol{x}_i$, y_i, s_i), где $\boldsymbol{x}_i$ – это вектор значений признаков для примера i, y_i – метка класса для примера i, а s_i – двоичная переменная, показывающая, был ли пример i размечен. Распределение переменной класса, y, для PU-данных неизвестно, но информация о нем может быть получена из значений переменной s. Если пример i размечен, то есть $s_i = 1$, то он всегда принадлежит положительному классу $P(y = 1 \mid s = 1) = 1$. Если пример i не размечен, то есть $s_i = 0$, то он может принадлежать как к положительному, так и к отрицательному классу.

Размеченные положительные примеры выбираются из всего набора положительных примеров в соответствии с вероятностным механизмом разметки, где каждый положительный пример $\boldsymbol{x}_i^{\mathbf{P}}$ имеет вероятность $e(\boldsymbol{x}) = P(s = 1 \mid y = 1, \boldsymbol{x})$ быть выбранным для разметки. Функцию $e(\boldsymbol{x})$ называют оценкой склонности (англ. propensity score) [10]. Следовательно, распределение вероятности разметки примера данных является смещенной версией распределения положительного класса в данных: $f_L(\boldsymbol{x}) = (e(\boldsymbol{x}) / c) \cdot f_P(\boldsymbol{x})$, где $f_L(\boldsymbol{x})$ и $f_P(\boldsymbol{x})$ – функции плотности вероятности распределений размеченных и положительный примеров.

Набор PU-данных, содержащий положительные и неразмеченные примеры, может быть сформирован двумя способами:

1) и положительные, и неразмеченные примеры взяты из одного и того же набора данных;
2) положительные и неразмеченные примеры взяты из двух независимо сформированных наборов: одного со всеми положительными примерами и другого со всеми неразмеченными примерами.

Эти два способа формирования выборки обычно называют сценарием с одним обучающим набором и сценарием «случай-контроль» соответственно [11].

Сценарий с одним обучающим набором предполагает, что примеры положительных и неразмеченных данных взяты из одного и того же набора данных, и что этот набор данных является независимой и одинаково распределенной выборкой из реального распределения, по аналогии с данными для классической задачи бинарной классификации. При этом доля c положительных примеров выбрана для разметки, и в PU-данных будет присутствовать доля $a \cdot c$ размеченных примеров:

$$\boldsymbol{x} \sim f(x) \sim \alpha \cdot f_P(\boldsymbol{x}) + (1 - \alpha) \cdot f_N(\boldsymbol{x}) \sim \alpha \cdot c \cdot f_L(\boldsymbol{x}) + (1 - \alpha \cdot c) \cdot f_U(\boldsymbol{x}).$$

Сценарий «случай-контроль» предполагает, что положительные и неразмеченные примеры взяты из двух независимых наборов данных, и неразмеченный набор данных является независимой и одинаково распределенной выборкой из реального распределения:

$$\boldsymbol{x} \mid s = 0 \sim f_U(\boldsymbol{x}) \sim f(\boldsymbol{x}) \sim \alpha \cdot f_P(\boldsymbol{x}) + (1 - \alpha) \cdot f_N(\boldsymbol{x}).$$

Наблюдаемые положительные примеры в обоих сценариях генерируются из одного и того же распределения. Следовательно, в обоих сценариях алгоритм обучения имеет доступ к набору PU-данных, сформированному независи-

мой и одинаково распределенной выборкой из реального распределения и набора примеров, которые взяты из распределения примеров положительного класса в соответствии с механизмом разметки, который определяется оценкой склонности $e(\boldsymbol{x})$. В результате большинство методов PU-обучения могут обрабатывать наборы, сформированными по обоим сценариям, однако оценки точности могут отличаться. Это необходимо учитывать при интерпретации результатов и использовании программного обеспечения.

Учитывая особенности формирования выборки для экспериментального исследования подхода к классификации текстов с целью выявления вредоносных постов в социальных сетях, описанного далее в работе, можно сказать, что она сформирована по сценарию «случай-контроль».

Большинство разработанных к сегодняшнему дню методов PU-обучения можно отнести к одной из следующих категорий:

1) двухэтапные методы;
2) смещённое обучение (англ. biased learning);
3) методы, использующие априорное распределение классов.

В подходе смещённого обучения PU-данные рассматриваются как полностью размеченные данные с шумом в примерах отрицательного класса. Методы, использующие априорное распределение классов, модифицирует стандартные методы обучения, в явном виде включая в них априорное распределение классов.

Рассмотрим немного подробнее двухэтапные методы PU-обучения, так как представитель именно этого класса методов апробирован автором в задаче выявления потенциально вредоносных постов в социальных сетях.

Двухэтапные методы основаны на допущениях о разделимости и гладкости данных: предполагается, что все положительные примеры аналогичны размеченным примерам, а отрицательные примеры сильно от них отличаются. Следуя этой идее, двухэтапные методы включают следующие шаги:

Шаг 1. Идентифицировать надежные отрицательные примеры. Дополнительные положительные примеры также могут быть идентифицированы в неразмеченных данных [12].

Шаг 2. Использовать обучение с учителем либо полуавтоматические методы обучения для размеченных положительных и надежных отрицательных примеров. Возможно также использование оставшихся неразмеченных примеров.

Шаг 3 (опциональный). Выбрать лучший классификатор, созданный на шаге 2.

Рассмотрим несколько ранних примеров двухэтапных методов PU-обучения.

Метод S-EM [12] основан на применении наивного байесовского классификатора и максимизации математического ожидания (англ. expectation maximization, далее EM-алгоритм). На первом шаге метода некоторые из размеченных примеров превращаются в т.н. «шпионов» (англ. spy) и добавляются в неразмеченную часть набора данных. Затем на модифицированных данных обучается наивный байесовский классификатор, рассматривающий все нераз-

меченные примеры как отрицательные, после чего набор итеративно модифицируется с применением EM-алгоритма согласно результатам работы наивного байесовского классификатора. Надежными отрицательными примерами считаются все неразмеченные отрицательные примеры, для которых апостериорная вероятность ниже, чем апостериорная вероятность любого из «шпионов». Для этого метода важно иметь достаточно количество размеченных примеров, в противном случае подвыборка «шпионов» будет слишком мала, а результаты ненадежны.

Метод Roc-SVM [13], как и S-EM, относится к двухэтапным. На первом этапе для нахождения надежных отрицательных примеров в неразмеченных данных используется классификатор Rocchio [14]. Для его обучения в качестве отрицательных примеров используются все неразмеченные данные. После этого обученный классификатор Rocchio применяется для классификации всех неразмеченных примеров исходного набора. Те примеры, которые классифицируются как отрицательные, далее рассматриваются как надежные отрицательные примеры. На втором этапе Roc-SVM итеративно выполняются запуски метода опорных векторов (англ. support vector machines, далее SVM) и окончательный выбор итогового классификатора.

Рассмотрим этапы работы метода Roc-SVM более подробно. Пусть обучающая выборка $\mathbf{Z}$ содержит примеры положительного класса $\mathbf{Z^P}$ и неразмеченные примеры $\mathbf{Z^U}$, $\mathbf{Z} = \mathbf{Z^P} \cup \mathbf{Z^U}$. На первом шаге метода Roc-SVM алгоритм Rocchio рассматривает все неразмеченные примеры $\mathbf{Z^U}$ как примеры отрицательного класса $\mathbf{Z^N} = \mathbf{Z^U}$ и использует примеры положительного класса $\mathbf{Z^P}$ и отрицательного класса $\mathbf{Z^N}$ для обучения классификатора Rocchio в рамках классического подхода к бинарной задаче классификации. Затем обученный классификатор Rocchio используется для классификации всей неразмеченной выборки $\mathbf{Z^U}$, и те примеры, которые классифицируются как отрицательные, обозначаются как надежные (англ. reliable) отрицательные примеры $\mathbf{Z^R}$.

Итак, для обучения классификатора Rocchio сперва с использованием некоторых весовых коэффициентов α и β рассчитываются векторы «усредненного» (англ. prototype vector) положительного примера $\bar{\boldsymbol{x}}^P$ и отрицательного примера $\bar{\boldsymbol{x}}^N$ (шаг 2 на рис. 1). Далее для классификации всех примеров $\boldsymbol{x}_i \in \mathbf{Z}$ применяется простое решающее правило, которое использует меру косинуса в качестве меры сходства примеров с усредненными векторами положительного и отрицательного классов: если косинусная метрика сходства, рассчитанная для пары «пример – усредненный положительный пример» меньше, чем косинусная метрика сходства, рассчитанная для пары «пример – усредненный отрицательный пример», то пример относится к положительному классу, в противном случае – к отрицательному (шаги 3-4 на рис. 1).

Далее примеры, классифицированные как отрицательные, образуют подвыборку надежных отрицательных примеров $\mathbf{Z^R}$ (шаг 5).

Алгоритм извлечения надежных отрицательных примеров показан на рис. 1.

1. Инициализировать подмножество отрицательных примеров для обучения классификатора:
   $\mathbf{Z^N} := \mathbf{Z^U}$.
2. Рассчитать векторы усредненного положительного примера и отрицательного примера:

$$\bar{x}^{P} := \alpha \cdot \frac{1}{\left|Z^{P}\right|} \cdot \sum_{x_i \in Z^{P}} \frac{x_i}{\left\|x_i\right\|} - \beta \cdot \frac{1}{\left|Z^{N}\right|} \cdot \sum_{x_i \in Z^{N}} \frac{x_i}{\left\|x_i\right\|}$$

$$\bar{x}^{N} := \alpha \cdot \frac{1}{\left|Z^{N}\right|} \cdot \sum_{x_i \in Z^{N}} \frac{x_i}{\left\|x_i\right\|} - \beta \cdot \frac{1}{\left|Z^{P}\right|} \cdot \sum_{x_i \in Z^{P}} \frac{x_i}{\left\|x_i\right\|}$$

3. **Для каждого** $x_i \in \mathbf{Z}$:
4. **Если** $sim\left(\bar{x}^{P}, x_i\right) \le sim\left(\bar{x}^{N}, x_i\right)$, **то**:
5. $\mathbf{Z^R} := \mathbf{Z^R} \cup \{x_i\}$.

Рис. 1. Алгоритм извлечения надежных отрицательных примеров $x_i \in \mathbf{Z^R}$ с помощью классификатора Roccio [13]

На втором этапе Roc-SVM итеративно выполняется обучение классификатора с помощью метода опорных векторов на подвыборках $\mathbf{Z^P}$ и $\mathbf{Z^R}$ и окончательный выбор итогового классификатора.

Метод опорных векторов получил широкую известность в 1990-е годы, и до сих пор считается одним из лучших методов классификации текстов. Основная идея метода опорных векторов заключается в построении оптимальной разделяющей гиперплоскости в пространстве признаков таким образом, чтобы примеры разных классов оказались по разную сторону от этой гиперплоскости и были максимально от нее отдалены. В ходе обучения SVM классификатора выполняется решение задачи квадратичного программирования с единственным решением для определения положения этой оптимальной разделяющей гиперплоскости, которое задается конечным небольшим количеством примеров из набора данных, которые называют опорными векторами. Метод опорных векторов может быть обобщен на случай нелинейных разделяющих поверхностей с помощью так называемых функций ядра.

Обозначим через $\mathbf{Z^Q}$ подвыборку примеров, оставшихся неразмеченными после извлечения надежных отрицательных примеров, $\mathbf{Z^Q} = \mathbf{Z^U} - \mathbf{Z^R}$. На рис. 2 приведен полный алгоритм итеративного обучения SVM классификаторов на этом шаге Roc-SVM.

Обучение SVM классификаторов на шагах 5-16 выполняется итеративно, так как множество надежных отрицательных примеров $\mathbf{Z^R}$, сформированное на шаге 2, может быть не достаточно большим для обучения качественного классификатора. SVM классификаторы S_i могут использоваться для извлечения дополнительных надежных отрицательных примеров из $\mathbf{Z^Q}$ в итеративном режиме. Цикл извлечения надежных отрицательных примеров останавливается, когда больше невозможно извлечь дополнительные отрицательные примеры (шаги 11-12). В качестве критерия выбора итогового алгоритма *k* авторы предлагают использовать долю неверно классированных положительных примеров из

$\mathbf{Z^P}$ и пороговое значение k^* равное 5%. Если 5% положительных примеров из множества $\mathbf{Z^P}$ классифицируются как отрицательные, это может указывать на то, что SVM классификатор переобучился. В этом случае в качестве итогового классификатора будет использовать классификатор, обученный на первой итерации цикла, S_1. В противном случае в качестве итогового будет использован классификатор S_{last}.

1. Присвоить всем примерам $\boldsymbol{x}_i \in \mathbf{Z^P}$ метку класса +1.
2. Присвоить всем примерам $\boldsymbol{x}_i \in \mathbf{Z^R}$ метку класса –1.
3. Инициализировать итератор: j: = 0.
4. Инициализировать множество примеров, классифицированных как отрицательные: $\tilde{\mathbf{Z}}_\mathbf{N} := \varnothing$
5. **Выполнять**:
6. Обучить SVM классификатор S_j на примерах $\boldsymbol{x}_i \in \mathbf{Z^P}$ и $\boldsymbol{x}_i \in \mathbf{Z^R}$.
7. **Для каждого** $\boldsymbol{x}_i \in \mathbf{Z^Q}$:
8. классифировать $\boldsymbol{x}_i$ с помощью классификатора S_j.
9. **если** $\boldsymbol{x}_i$ классифицирован как пример отрицательного класса, **то**:
10. $\tilde{\mathbf{Z}}_N := \tilde{\mathbf{Z}}_N \cup \{\boldsymbol{x}_i\}$.
11. **Если** $\tilde{\mathbf{Z}}_N := \varnothing$, **то**:
12. выйти из цикла,
13. **иначе**:
14. $\mathbf{Z^Q} := \mathbf{Z^Q} - \tilde{\mathbf{Z}}_N$,
15. $\mathbf{Z^R} := \mathbf{Z^R} \cup \tilde{\mathbf{Z}}_N$,
16. $j := j + 1$.
17. Использовать SVM классификатор, полученный на последнем шаге цикла, S_{last}, для классификации примеров $\boldsymbol{x}_i \in \mathbf{Z^P}$.
18. Рассчитать долю неверно классифицированных положительных примеров $\boldsymbol{x}_i \in \mathbf{Z^P}$, k.
19. **Если** $k > k^*$, то:
20. использовать классификатор S_1 в качестве итогового SVM классификатора $S^* = S_1$,
21. **иначе**:
22. использовать S_{last} в качестве итогового SVM классификатора $S^* = S_{last}$.

Рис. 2. Алгоритм обучения итогового SVM классификатора [13]

Далее в разделе 5 выполнено экспериментальное исследование алгоритма Roc-SVM применительно к задаче выявления потенциально вредоносных постов в социальных сетях.

Еще один двухэтапный алгоритм CR-SVM [4] включает следующие два этапа:

1) извлечение набора потенциальных отрицательных примеров с использованием метрики косинуса для подсчета сходства (идея состоит в том, что неразмеченные примеры, которые сильно отличаются от положительных примеров, вероятно, являются отрицательными);
2) извлечение итогового набора надежных отрицательных примеров из неразмеченных данных.

Второй шаг алгоритма CR-SVM аналогичен шагу в методе Roc-SVM: создание окончательного классификатора путем многократного запуска SVM и окончательный выбор итогового классификатора. Таким образом, в некоторых задачах метод CR-SVM позволяет сформировать более качественный набор отрицательных примеров, что впоследствии ведет к улучшению точности итогового классификатора CR-SVM.

3. Подходы к оценке качества модели PU-обучения

Оценка качества алгоритмов PU-обучения может представлять проблему, так как метки классов известны только для положительных примеров данных. Поэтому часто для оценки качества алгоритмов PU-обучения используют размеченные наборы данных с частично обезличенными примерами [15]. Этот подход предполагает разделение множества примеров одного из классов на два подмножества, при этом примеры из первого подмножества будут использованы в качестве положительных примеров, а примеры из второго подмножества будут обезличены и смешаны с остальными неразмеченными примерами [15].

Многие современные алгоритмы PU-обучения, например, метод оценки плотностей распределений классов на основе PU-обучения (англ. difference of estimated densities based positive-unlabeled learning, далее DEDPUL) [15], метод априорной оценки распределения классов для обучения на основе положительных и неразмеченных данных (англ. class-prior estimation for dearning from positive and unlabeled data) [16] или метод оценка смеси распределений с помощью ядерных вложений для распределений (англ. mixture proportion estimation via kernel embeddings of distributions) [17], не используют традиционные метрики для оценки качества обучение, такие как аккуратность, точность, полнота и др. Авторы упомянутых подходов предлагают использовать модуль разницы между реальной и вычисленной оценкой склонности, не учитывая оценки качества обучения по конкретным данным.

Большинство исследователей, тем не менее, настаивают на тестировании и оценке качества обученных PU-классификаторов на наборе данных с заранее известными метками классов. Для этого обычно используются следующие метрики:

– среднеквадратичная ошибка для оценки соответствия наблюдаемого и предсказанного классов, которая может быть рассчитана по классической формуле:

$$RMSE = \sqrt{\frac{1}{N} \cdot \sum_i \left(y_i - \tilde{y}_i \right)^2} ,$$

где N – общее число классифицированных примеров; y_i – подлинный класс примера i, $\tilde{y}_i$ – класс примера i, предсказанный классификаторов, который необходимо оценить. Метрика среднеквадратичной ошибки достаточно проста, быстро вычисляется и достаточно показательна в случае, когда множества примеров различных классов соизмеримы;

- стандартная F-мера, которая представляет собой среднее гармоническое оценок точности и полноты классификатора. Считается, что F-мера лучше подходит для оценки качества PU-классификатора, чем метрика среднеквадратичной ошибки, в случае, если множества примеров различных классов несоизмеримы, а также в случае, когда цена ошибок первого и второго рода различна;
- некоторые другие метрики, например, метрика, описанная авторами [3], которая позволяет оценить долю достоверно выявленных отрицательных примеров и может быть рассчитана следующим образом:

$$ERR = \frac{\tilde{\mathbf{Z}}_{\boldsymbol{P}}^{\boldsymbol{R}}}{\mathbf{Z}_{\boldsymbol{P}}^{\boldsymbol{U}}},$$

где $\tilde{\mathbf{Z}}_{\boldsymbol{P}}^{\boldsymbol{R}}$ – это подмножество примеров из множества надежных отрицательных примеров $\mathbf{Z}^{\mathrm{R}}$, классифицированных как положительные, а $\mathbf{Z}_{\mathrm{P}}^{\mathrm{U}}$ – подмножество положительных примеров, которые были добавлены к неразмеченным примерам.

4. Алгоритм выявления потенциально вредоносных постов с использованием PU-классификации текстов

Подход PU-обучения предполагается использовать для обучения классификатора в рамках программного компонента анализа данных системы выявления потенциально вредоносного контента в социальных сетях, которая разрабатывается при поддержке Российского научного фонда[1]. Основной функцией программного компонента является семантический анализ и идентификация потенциально вредоносных текстов, а также отнесение этих текстов в некоторой категории из заданного списка.

Для решения этой задачи предлагается использовать машинное обучение, а именно, методы классификации. Однако из-за проблем, подробно описанных в разделе 1, в частности, из-за особенностей обучающих данных для классификации – сложностей с формированием подмножества отрицательных примеров, которое должно включать тексты, на все возможные темы, обсуждаемые в социальных сетях – классический подход к классификации может работать не достаточно хорошо.

Далее описаны этапы разработанного алгоритма выявления потенциально вредоносных постов в социальных сетях, содержащих текст на естественном языке, с использованием PU-классификации и последующей многоклассовой классификации текстов.

[1] Проект № 18-71-10094.

Описанные ниже этапы относятся к стадии функционирования алгоритма в рамках компонента анализа данных системы выявления потенциально вредоносного контента в социальных сетях. Стадии функционирования предшествует стадия обучения классификаторов. Классификаторы, функционирующие на этапах 3 и 4, для решения некоторый задачи должны быть обучены отдельно друг от друга, но на одном и том же наборе данных. Процесс обучения классификаторов тривиален, и по этой причине его подробное описание в работе опущено.

Этап 1. *Предварительная обработка текстов.*

На первом этапе выполняется традиционная предобработка текстов с помощью регулярных выражений. В частности, из текстов должны быть удалены лишние пробельные символы, знаки препинания, URL адреса, хэштеги и ссылки на других пользователей социальной сети[2]. Предварительная обработка текста также обычно включает в себя токенизацию и лемматизацию, удаление стоп-слов (семантически нейтральных слов, таких как предлоги, союзы, вспомогательные глаголы, междометия, артикли и т.п.), стемминг и морфологический анализ.

Этап 2. *Выделение признаков текстов (векторизация).*

На этапе извлечения признаков выполняется построение векторной модели текста, которая позволит представить текст в подходящем для дальнейшей обработки виде.

В текущей версии экспериментального прототипа компонента для векторизации текстов применяется модель «мешка слов» (англ. bag-of-words) со взвешиванием слов или словосочетаний по метрике TF-IDF.

В рамках модели «мешка слов» каждый исходный текст t_i обучающей выборки размера n представляется в виде множества содержащихся в нем слов $t_i = \{w_1^i, \ldots, w_v^i\}$. Далее из всех слов w_j^i, которые хотя бы единожды встречаются во всех текстах обучающей выборки, формируется словарь уникальных слов $\mathbf{D} = \{w_j^*\}|_{j=1}^h$. После этого каждый текст t_i может быть представлен списком пар $l_i = \{(w_1^*, H(w_1^*, t_i)), \ldots, (w_h^*, H(w_h^*, t_n))\}$, где $H(w, t)$ – это некоторая функция, позволяющая оценить частоту появления некоторого слова в тексте. В простейшем случае функция H может подсчитывать количество упоминаний слова в тексте, однако чаще всего, в качестве оценки частоты используют различные более сложные метрики. Наиболее популярной такой метрикой сегодня является метрика TF-IDF, которая может быть рассчитана следующим образом: $\text{TF-IDF}(w, t) = \text{TF}(w, t) \cdot \text{IDF}(w, t)$, где $\text{TF}(w,t)$ – это отношение количества вхождений слова к общему числу слов в тексте, а $\text{IDF}(w,t)$ – это так называемая инверсированная частота документа (англ. inverse document frequency), т.е. инверсия частоты, с которой некоторое слово встречается во всех текстах выборки. Именно метрику TF-IDF предлагают использовать авторы алгоритма Roc-SVM [13].

[2] URL адреса, хэштеги и ссылки на пользователей в дальнейшем будут использованы отдельно в рамках других алгоритмов анализа данных.

Вместо отдельных слов в модели «мешка слова» также могут использоваться пары, тройки и т.д., часто встречающихся слов, так называемые *n*-граммы.

Вместо модели «мешка слов» на этапе извлечения признаков из текстов может применяться любая другая модель векторизации текстов, например, word2vec [18], GloVe [19], Elmo [20], GPT-2 [21], BERT [22], RoBerta [23] и др. Эти модели и их аналоги положительно зарекомендовали себя в вопросно-ответных системах, при создании чат-ботов, в автоматических переводчиках и некоторых видах анализа текстов, например, в тематическом моделировании. Следует, однако, отметить, что такие модели не предназначены для работы с узкоспециализированными текстами – для этого требуется их интенсивное и вычислительно сложное до-обучение, поэтому их применимость к задаче выявления вредоносных информационных объектов является предметом дальнейших исследований.

Этап 3. *Выявления потенциально вредоносных текстов.*

На данном этапе для выявления потенциально вредоносных текстов предлагается использовать автоматический классификатор, предварительно обученный с помощью подхода PU-обучения, а именно, двухэтапного метода Roc-SVM [13], описанного выше. Оптимальные параметры классификатора подбираются в ходе обучения для каждого отдельного набора данных на этапе обучения.

Этап 4. *Определение категории потенциально вредоносного текста.*

На последнем этапе выполняется оценка степени принадлежности текстов к той или иной категории вредоносной информации с помощью классического автоматического многоклассового классификатора. В качестве классификатора на этом этапе может быть использован любой классический классификатор, например, метод опорных векторов, случайный лес или нейронная сеть. В экспериментальном исследовании, результаты которого представлены ниже, был использован метод опорных векторов.

На рис. 3 представлена общая схема разработанного алгоритма выявления потенциально вредоносных постов в социальных сетях, содержащих текст на естественном языке, с привлечением PU-классификации текстов.

Как было сказано выше, предполагается, что программный компонент семантического анализа, в рамках которого реализован описанный выше алгоритм, на стадии функционирования использует заранее обученные классификаторы. На стадии обучения классификаторов используются предварительно размеченные экспертами потенциально вредоносные тексты из постов в социальных сетях на заранее заданные вредоносные темы. Множество таких текстов выступает в качестве положительного класса PU-классификатора. Кроме того, обучающая выборка должна содержать подмножество контрпримеров – примеров отрицательного класса. В качестве контрпримеров могут выступать тексты из социальных сетей, собранные случайно и выборочно проверенные экспертами.

После обучения классификаторов анализ потенциальных вредоносных текстов может быть выполнен по запросу от других компонентов разрабатываемой системы.

Программный прототип алгоритма выявления потенциально вредоносных постов в социальных сетях разработан на языке Python с применение библиотеки Scikit-learn.

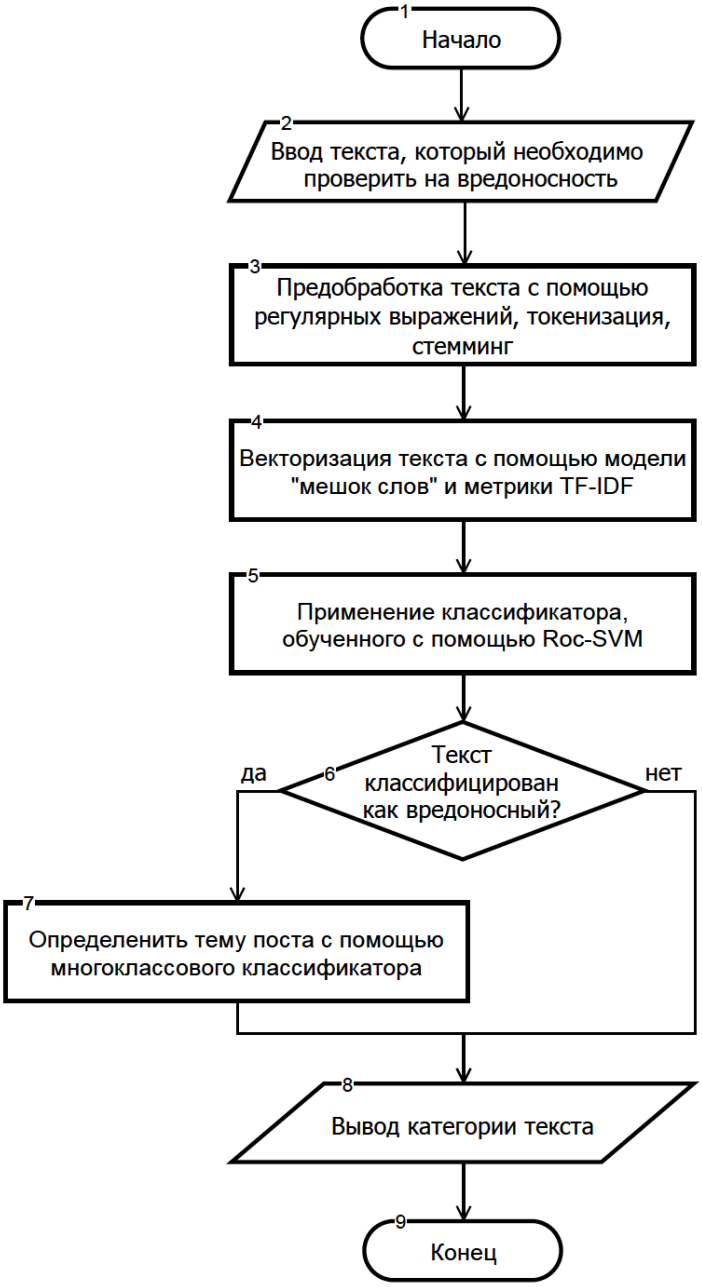

Рис. 3. Общая схема алгоритма выявления потенциально вредоносных постов в социальных сетях с использование PU-классификации

5. Экспериментальное исследование применения PU-обучения в области классификации вредоносных текстов

Для проведения экспериментального исследования разработанного подхода к идентификации вредоносных текстов с помощью PU-классификации в

рамках задачи выявления потенциально вредоносных постов в социальных сетях в ходе упомянутого выше проекта был сформирован обучающий набор данных, содержащий текстовую информацию потенциально вредоносных постов из социальной сети Вконтакте на следующие темы: агрессия, азартные игры, опасные теории заговоров, проституция, радикализм, секты.

Кроме того, для экспериментального исследования в качестве отрицательных примеров обучающей выборки дополнительно были собраны случайные посты из социальной сети Вконтакте.

В ходе исследования было проведено несколько серий экспериментов, в которых для выявления вредоносного контента использовались различные подходы к классификации и данные, описанные выше. В ходе экспериментов использовался только полный текст поста на русском языке. Тексты длиной более 2000 символов в эксперименте не использовались. Количество случайных постов было сокращено до 25000 с помощью случайного выбора.

В таблице 1 представлено распределение использованных в экспериментах текстов по темам.

Таблица 1 – Распределение текстов экспериментального набора данных по темам

Тема	**Количество постов**
Агрессия	2946
Азартные игры	2510
Опасные теории заговоров	2766
Проституция	3441
Радикализм	3251
Секты	9532
Случайная тема	25000

Основной целью эксперимента являлось сравнение точности классического подхода к многоклассовой классификации и подхода к классификации, основанного на PU-обучении, при выявлении потенциального вредоносных постов социальной сети.

Для проведения экспериментального исследования программное приложение, реализующее программный прототип алгоритма, было развернуто на виртуальной машине под управлением оперативной системы CentOS на базе гипервизора VMware ESXi 6.0.0. Виртуальная машина располагает 4 вычислительными ядрами с частотой работы 2,1 ГГц и 64 Гб оперативной памяти.

Как было сказано выше, для предобработки текстов и выделения признаков в ходе экспериментального исследования использовалась модель векторизации «мешок слов» с метрикой TF-IDF, также из текстов были удалены стандартные и специфичные для набора данных стоп-слова.

В первой серии экспериментов для построения базового (англ. baseline) алгоритма использовался подход классической многоклассовой классификации и классификатор, обученный с помощью метода опорных векторов. Набор дан-

ных был разбит на две части в соотношении 80% на 20%. На первой части выборки с помощью скользящего контроля с пятью блоками выполнялся подбор оптимальных параметров для классификатора. На второй части выборки выполнялась оценка классификатора с оптимальными параметрами из каждой категории с помощью аккуратности, точности, полноты и F-меры.

Одним из подходов к тестированию методов PU-обучения является добавление небольшой доли примеров положительного класса к неразмеченным данным [4] с целью оценить, насколько сильно это повлияет на итоговую точность классификации. В ходе первой серии экспериментов в подвыборку постов на случайную тему были добавлены 5, 10, 15 и 20% ложноотрицательных примеров, то есть текстов, которые в действительности относятся к вредоносным темам, для того, чтобы оценить, как это повлияет на точность. Такой сценарий эксперимента соответствует ситуации реального сбора данных из социальной сети, так как у исследователей, разработчиков и экспертов обычно очень ограничены временные ресурсы на разметку, поэтому самым простым и реализуемым способом собрать хотя бы в какой-то мере репрезентативную выборку текстов на нейтральную тему является сбор случайных постов из социальной сети. Однако среди собранных случайных постов с некоторой вероятностью могут оказаться вредоносные тексты. Для обучения классификатора использовался метода опорных векторов (SVM). Эксперимент направлен на то, чтобы продемонстрировать, как с этой проблемой может справиться классический многоклассовый классификатор и комбинация классификатора, основанный на PU-обучении с последующей многоклассовой классификацией.

Оценки точности классического многоклассового классификатора текстов представлены в таблице 2.

Таблица 2 – Оценки точности
многоклассовой классификации текстов с помощью SVM

Доля ложноотрицательных примеров в неразмеченных данных, %	Аккуратность, %	Точность, %	Полнота, %	F-мера, %
0	84	85	84	84
5	81	83	81	81
10	80	79	80	80
15	77	76	75	75
20	75	74	73	73

Во второй серии экспериментов был исследован разработанный алгоритм выявления потенциально вредоносных постов в социальных сетях, описанный в разделе 3 и включающий метод PU-обучения Roc-SVM и последующую многоклассовую классификацию текстов с помощью метода опорных векторов. В ходе второй серии экспериментов в подвыборку постов на случайную тему также были добавлены 5, 10, 15 и 20% ложноотрицательных примеров для того, чтобы оценить, как это повлияет на итоговую оценку качества. Оценка точности

разработанного алгоритма выявления потенциально вредоносных постов в социальных сетях в таблице 3.

Таблица 3 – Оценка точности разработанного алгоритма выявления потенциально вредоносных текстов

Доля ложноотрицательных примеров в неразмеченных данных, %	Аккуратность, %	Точность, %	Полнота, %	F-мера, %
0	87	85	85	85
5	85	86	85	84
10	84	85	84	83
15	82	83	82	82
20	81	83	81	80

Сравнение результатов экспериментального исследования показывает, что разработанный алгоритм выявления потенциально вредоносных постов работает стабильнее (с точки зрения оценок точности) традиционного подхода многоклассовой классификации при добавлении ложноотрицательных примеров в данные. Следовательно, подход к классификации, основанный на PU-обучении является перспективной альтернативой классическому подходу к классификации и должен быть исследован применительно к другим задачам в области защиты от информации.

Заключение

В данной работе рассмотрены аспекты применения подхода к обучению классификаторов на основе положительных и неразмеченных примеров (PU-обучение) в задаче выявления потенциально вредоносных постов в социальной сети. Эта задача обычно решается в рамках такого направления, как защита от информации.

В работе обоснована целесообразность применения алгоритмов PU-обучения к задаче выявления потенциально вредоносных постов в социальных сетях и рассмотрены особенности некоторых подходов к PU-обучению. Новизна предложенного решения заключается в том, что разработанный алгоритм предусматривает применение подхода PU-обучения в комбинации с многоклассовой классификации к постам в социальных сетях, содержащих текст на естественном языке, для решения задачи из области защиты от информации.

Для проведения экспериментального исследования предложенного алгоритма выявления потенциально вредоносных постов в социальных сетях был использован набор данных, содержащий текстовую информацию 24 446 потенциально вредоносных постов из социальной сети Вконтакте на следующие темы: агрессия, азартные игры, опасные теории заговоров, проституция, радикализм, секты; а также текстовую информацию 25 000 постов из социальной сети Вконтакте на случайную тему.

Сравнение результатов проведенного экспериментального исследования показывает, что разработанный алгоритм выявления потенциально вредонос-

ных постов в социальных сетях работает стабильнее (с точки зрения оценок точности) традиционного подхода многоклассовой классификации при добавлении ложноотрицательных примеров в данные на описанном наборе данных.

В дальнейшем планируется рассмотреть и сравнить другие известные методы PU-обучения при решении описанной задачи, а также выполнить экспериментальное исследование разработанного алгоритма с использованием данных большего объема, включающих разнообразные потенциально вредоносные темы.

Работа выполнена при частичной финансовой поддержке бюджетной темы 0073-2019-0002 в СПб ФИЦ РАН и Российского научного фонда (проект №18-71-10094).

Литература

1. Kotenko I., Vitkova L., Saenko I., Tushkanova O., Branitskiy A. The intelligent system for detection and counteraction of malicious and inappropriate information on the Internet // AI Communication. 2020. Vol. 33. № 1. P. 13-25. doi: 10.3233/aic-200647.

2. Vitkova L., Kotenko I., Kolomeets M., Tushkanova O., Chechulin A. Hybrid Approach for Bots Detection in Social Networks Based on Topological, Textual and Statistical Features // Proceedings of the Fourth International Scientific Conference Intelligent Information Technologies for Industry. Springer. 2020. Vol. 1156. P. 412-421. doi: 10.1007/978-3-030-50097-9_42.

3. Liu L., Peng, T. Clustering-based method for positive and unlabeled text categorization enhanced by improved TFIDF // Journal of Information Science and Engineering. 2014. Vol. 30. P. 1463-1481.

4. Li X.L., Liu B., Ng S. K. Negative training data can be harmful to text classification // Proceedings of the 2010 conference on empirical methods in natural language processing. Association for Computational Linguistics. 2010. P. 218-228.

5. Liu B., Dai Y., Li X., Lee W.S., Yu P.S. Building text classifiers using positive and unlabeled examples // Proceedings of the Third IEEE International Conference on Data Mining. IEEE. 2003. P. 179-186.

6. Denis F., Laurent A., Gilleron R., Tommasi M. Text classification and co-training from positive and unlabeled examples // Proceedings of the ICML 2003 workshop: The Continuum from Labeled to Unlabeled Data. 2003. P. 80-87.

7. Li X.L., Liu B. Learning from positive and unlabeled examples with different data distributions // European Conference on Machine Learning. Springer. 2005. P. 218-229.

8. Khan S., Madden M. One-class classification: taxonomy of study and review of techniques // The Knowledge Engineering Review. 2014. Vol. 29. № 3. P. 345-374. doi: 10.1017/S026988891300043X.

9. Chapelle O., Scholkopf B., Zien A. Semi-supervised learning // IEEE Transactions on Neural Networks. 2009. Vol. 20. №3. P. 542-542.

10. Bekker J., Davis J. Beyond the selected completely at random assumption for learning from positive and unlabeled data // CoRR. 2018. arXiv preprint arXiv:

1809.03207. — URL: http://arxiv.org/abs/1809.03207 (дата обращения: 25.11.2021).

11. Bekker J., Davis J. Learning from positive and unlabeled data: A survey // Machine Learning. 2020. Vol. 109. № 4. P. 719-760.

12. Liu B., Lee W. S., Yu P.S., Li X. 2002. Partially supervised text classification // Proceedings of 19th International Conference on Machine Learning. 2002. Vol. 2. № 485. P. 387-394.

13. Li X., Liu B. Learning to classify texts using positive and unlabeled data // Proceedings of the eighteenth International Joint Conference on Artifical Intelligence. 2003. Vol. 3. P. 587-592.

14. Rocchio J. Relevance feedback in information retrieval // The Smart retrieval system-experiments in automatic document processing. 1971. P. 313-323.

15. Ivanov D. DEDPUL: Difference of Estimated Densities based Positive-Unlabeled Learning // Proceedings of 19th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2020. P. 782-790.

16. Christoffel M., Niu G., Sugiyama M. Class-prior estimation for learning from positive and unlabeled data // Proceedings of Asian Conference on Machine Learning. PMLR, 2016. P. 221-236.

17. Ramaswamy H., Scott C., Tewari A. Mixture proportion estimation via kernel embeddings of distributions // Proceedings of International conference on machine learning. PMLR, 2016. P. 2052-2060. – URL: https://arxiv.org/abs/1603.02501 (дата обращения: 25.11. 2021).

18. Mikolov T., Chen K., Corrado G., Dean J. Efficient Estimation of Word Representations in Vector Space // ArXiv.org – e-print archive [Электронный ресурс]. 07.09.2013. – URL: https://arxiv.org/pdf/1301.3781.pdf (дата обращения: 25.11.2021).

19. Pennington J., Socher R., Manning C. D. Glove: Global vectors for word representation // Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014. P. 1532-1543. – URL: https://nlp.stanford.edu/projects/glove (дата обращения: 25.11.2021).

20. Peters M.E., Neumann M., Iyyer M., Gardner M., Clark C., Lee K., Zettlemoyer L. Deep contextualized word representations // ArXiv.org – e-print archive [Электронный ресурс]. 22.03.2018. URL: https://arxiv.org/pdf/1802.05365.pdf (дата обращения: 25.11.2021).

21. Radford A., Wu J., Child R., Luan D., Amodei D., Sutskever I. Language models are unsupervised multitask learners // OpenAI Blog. 2019. Vol. 1. № 8. P. 9.

22. Devlin J., Chang M.W., Lee K., Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding // ArXiv.org – e-print archive [Электронный ресурс]. 24.05.2019. URL: https://arxiv.org/pdf/1810.04805.pdf (дата обращения: 25.11.2021).

23. Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V. Roberta: A robustly optimized BERT pretraining approach // ArXiv.org – e-print archive [Электронный ресурс]. 26.07.2019. URL: https://arxiv.org/pdf/1907.11692.pdf (дата обращения: 25.11.2021).

References

1. Kotenko I., Vitkova L., Saenko I., Tushkanova O., Branitskiy A. The intelligent system for detection and counteraction of malicious and inappropriate information on the Internet. *AI Communication*, 2020, vol. 33, no. 1, pp. 13-25. doi: 10.3233/aic-200647.

2. Vitkova L., Kotenko I., Kolomeets M., Tushkanova O., Chechulin A. Hybrid Approach for Bots Detection in Social Networks Based on Topological, Textual and Statistical Features. *Proceedings of the Fourth International Scientific Conference Intelligent Information Technologies for Industry*. *Advances in Intelligent Systems and Computing*, 2020, vol. 1156, pp. 412-421. doi: 10.1007/978-3-030-50097-9_42.

3. Liu L., Peng, T. Clustering-based method for positive and unlabeled text categorization enhanced by improved TFIDF. *Journal of Information Science and Engineering*, 2014, vol. 30, pp. 1463-1481.

4. Li X.L., Liu B., Ng S.K. Negative training data can be harmful to text classification. *Proceedings of the 2010 conference on empirical methods in natural language processing. Association for Computational Linguistic*, 2010, pp. 218-228.

5. Liu B., Dai Y., Li X., Lee W. S., Yu P. S. Building text classifiers using positive and unlabeled examples. *Proceedings of the Third IEEE International Conference on Data Mining*, IEEE, 2003, pp. 179-186.

6. Denis F., Laurent A., Gilleron R., Tommasi M. Text classification and co-training from positive and unlabeled examples. *Proceedings of the ICML 2003 workshop: The Continuum from Labeled to Unlabeled Data*, 2003, pp. 80-87.

7. Li X.L., Liu B. Learning from positive and unlabeled examples with different data distributions. *European Conference on Machine Learning*, Springer, 2005, pp. 218-229.

8. Khan S., Madden M. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 2014, vol. 29, no. 3, pp. 345-374. doi: 10.1017/S026988891300043X.

9. Chapelle O., Scholkopf B., Zien A. Semi-supervised learning (Chapelle, o. et al., eds., 2006). *IEEE Transactions on Neural Networks*, 2009, vol. 20, no. 3, pp. 542-542.

10. Bekker J., Davis J. Beyond the selected completely at random assumption for learning from positive and unlabeled data. *CoRR*, 2018, arXiv preprint arXiv: 1809.03207. Avialable at: http://arxiv.org/abs/1809.03207 (accessed 25 November 2021).

11. Bekker J., Davis J. Learning from positive and unlabeled data: A survey. *Machine Learning*, 2020, vol. 109, no. 4, pp. 719-760.

12. Liu B., Lee W.S., Yu P.S., Li X. 2002. Partially supervised text classification. *Proceedings of 19th International Conference on Machine Learning*, 2002, vol. 2, no. 485, pp. 387-394.

13. Li X., Liu B. Learning to classify texts using positive and unlabeled data. *Proceedings of the eighteenth International Joint Conference on Artifical Intelligence*, 2003, vol. 3, pp. 587-592.

14. Rocchio J. Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*, 1971, p. 313-323.

15. Ivanov D. DEDPUL: Difference of Estimated Densities based Positive-Unlabeled Learning. *Proceedings of 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2020, pp. 782-790.

16. Christoffel M., Niu G., Sugiyama M. Class-prior estimation for learning from positive and unlabeled data. *Proceedings of Asian Conference on Machine Learning*, PMLR, 2016, pp. 221-236.

17. Ramaswamy H., Scott C., Tewari A. Mixture proportion estimation via kernel embeddings of distributions. *Proceedings of International conference on machine learning*, PMLR, 2016, pp. 2052-2060. Available at: https://arxiv.org/abs/1603.02501 (accessed 25 November 2021).

18. Mikolov T., Chen K., Corrado G., and Dean J. Efficient Estimation of Word Representations in Vector Space. Available at: https://arxiv.org/pdf/1301.3781.pdf (accessed 25 November 2021).

19. Pennington J., Socher R., Manning C. D. Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532-1543. Available at: https://nlp.stanford.edu/projects/glove (accessed 25 November 2021).

20. Peters M.E., Neumann M., Iyyer M., Gardner M., Clark C., Lee K., Zettlemoyer L. Deep contextualized word representations. *ArXiv.org – e-print archive*. Available at: https://arxiv.org/pdf/1802.05365.pdf (accessed 25 November 2021).

21. Radford A., Wu J., Child R., Luan D., Amodei D., Sutskever I. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019, vol. 1, no. 8, pp. 9.

22. Devlin J., Chang M.W., Lee K., Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv.org – e-print*. Available at: https://arxiv.org/pdf/1810.04805.pdf (accessed 25 November 2021).

23. Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V. Roberta: A robustly optimized BERT pretraining approach. *ArXiv.org – e-print.* Available at: https://arxiv.org/pdf/1907.11692.pdf (accessed 25 November 2021).

Статья поступила 29 ноября 2021 г.

Информация об авторе

Тушканова Ольга Николаевна – кандидат технических наук. Старший научный сотрудник лаборатории проблем компьютерной безопасности. Санкт-Петербургский Федеральный исследовательский центр Российской академии наук. Область научных интересов: машинное обучение; представление знаний; семантические технологии; причинный анализ; большие данные; интеллектуальные системы принятия решений; информационная безопасность. E-mail: tushkanova.on@gmail.com

Адрес: 199178, Россия, г. Санкт-Петербург, 14 линия В.О., д. 39.

Identification of potentially malicious posts on social networks using positive and unlabeled learning on text data

O. N. Tushkanova

Purpose. *The rapid growth of social networks audience and the lack of the possibility of high-quality manual content moderation require the development of automatic methods for detecting potentially malicious information based on machine learning to protect vulnerable groups from this information. A standard method of identifying potentially malicious texts in social networks is classical classification. However, well-known algorithms based on classical classification often do not demonstrate sufficient stability and accuracy in the presence of false negative examples in the training sample. The paper aims to assess the applicability of the machine learning approach based on positive and unlabeled training within the task of identifying malicious posts in social networks and develop an algorithm implementing this approach to improve the accuracy of malicious texts detection.* ***Methods****: in the developed algorithm, it is proposed to use machine learning based on positive and unlabeled classifiers training and the multiclass classification approach.* ***Novelty****: the novelty elements of the presented solution are the combination of an approach to training a classifier based on positive and unlabeled data and a classical multiclass classifier within the framework of an algorithm for solving the problem.* ***Result****: the expediency of applying approaches to machine learning based on positive and unlabeled data in the task of identifying texts containing potentially malicious information is substantiated, and the features of some such approaches are considered. An algorithm for identifying potentially malicious posts in social networks using machine learning based on positive and unlabeled text data and multiclass classification is proposed. The design and results of an experimental study of the proposed algorithm on a sample of the Vkontakte social network text posts are presented. It is shown that the developed algorithm works more stable (in terms of accuracy) than the traditional multiclass classification approach when false negative examples are in the training data.* ***Practical relevance****: the results of the study were used in the development of a system for detecting malicious content in social networks. The proposed approach makes it possible to increase the classification accuracy of malicious texts when false negative examples are in the training data.*

Key words*: machine learning, PU-learning, multiclass classification, malicious information, social network.*

Information about the author

Olga Nikolaevna Tushkanova – Ph.D. of Engineering Sciences. Senior Researcher at the Laboratory of Computer Security Problems. St. Petersburg Federal Research Center of the Russian Academy of Sciences. Research interests: machine learning; knowledge representation; semantic technologies; causal analysis; big data; intelligent decision-making systems; information security. E-mail: tushkanova.on@gmail.com

Address: 199178, Russia, St. Petersburg, 14 line V.O., 39.