Анализ структуры машинного обучения для выявления спама

Эта работа направлена на решение задачи отделения спама при помощи машинного обучения. Сделан анализ структуры обнаружения спама, предложены способы типизации. Подробно отражен математический аппарат и его работа, представлена информация о четкой реализации изучаемых алгоритмов. Определен конструктивизм установленной оценки для эффективности распознания спама.

1. Введение

Спам – это массовая рассылка сообщений пользователям, которые не давали своего согласия на ее получение и предприняли меры по недопущению данного факта [Cormack, 2008]. Спам-война является животрепещущей темой долгие десятилетия и уже есть некоторые успехи в разработке подходов по борьбе с обозначенное проблемой, согласно исследованиям Лаборатории, Касперского [Спам и фишинг во втором квартале 2016]. Тем не менее процент злосчастной рассылки в отечественном и мировом почтовом трафике все еще велик (рис. 1).

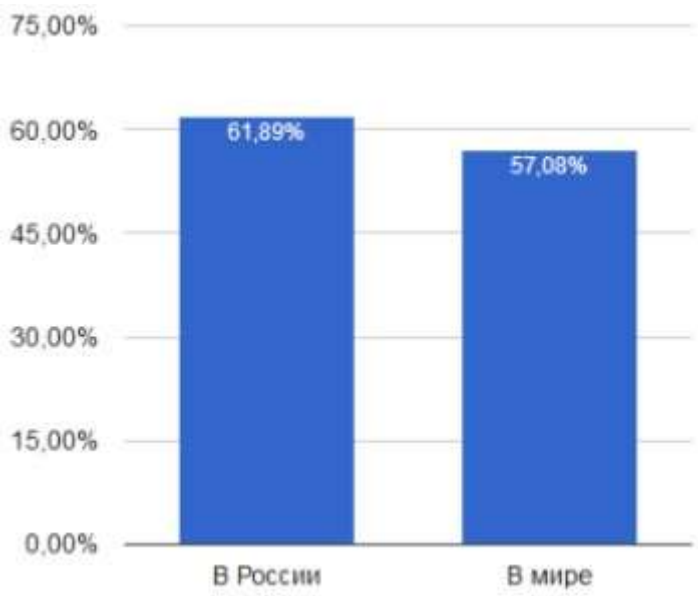

Рисунок 1 – Уровень спама мирового и российского почтового трафика за 2016 г.

Согласно рисунку 1, статистические данные показывают необходимость создания новых и совершенствования существующих методов определения спама. В этой работе рассматриваются алгоритмы решения задачи фильтрации спама через машинное обучение.

2. Классификация алгоритмов

Уже существует большое количество алгоритмов определения спама, поэтому целесообразно классифицировать их по некоторым критериям. Предлагается разделение по применяемому подходу или категориям математического аппарата, на котором строится алгоритм фильтрации (рис. 2). Такой вариант систематизирует, обобщает и расширяет классификации этой работы [Cormack, 2008].

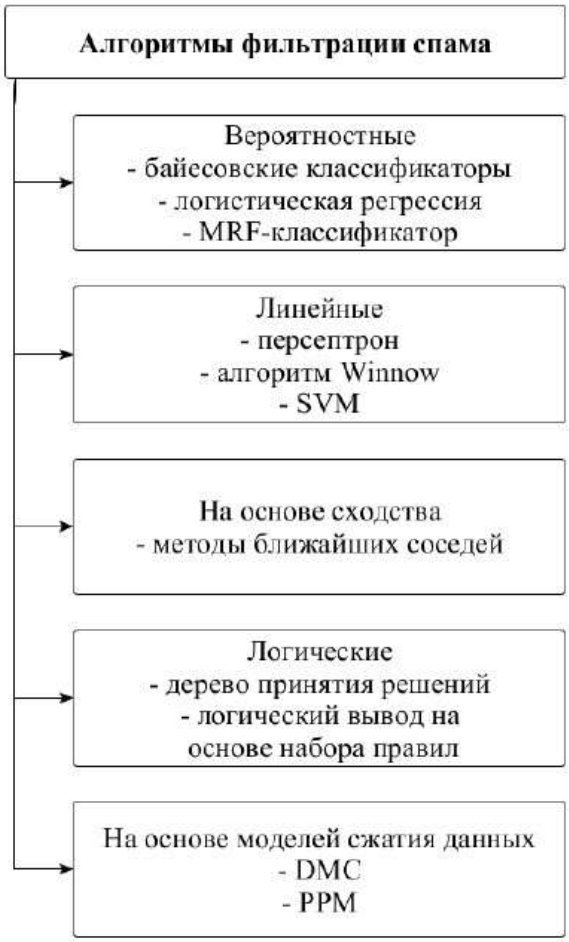

Рисунок 2 – Разделение алгоритмов определения спама по применяемому подходу

2.1 Условные обозначения

Допустим существует большое количество документации $\mathrm{D} = \{d_i\}$, классов, $\mathrm{C} = \{\mathrm{spam}, \mathrm{non\text{-}spam}\}$, терминов $T = \{t_i\}$. Определим $|\mathrm{T}| = \mathrm{n}$. Представим, что для обучения алгоритмов существует необходимое множество обучающих документов $\mathrm{D}' \subseteq \mathrm{D}$, то есть набор документов с заранее неизвестным классом. Добавим функцию $\mathrm{class}\colon \mathrm{D} \to \mathrm{C}$, корректно определяющую класс каждого документа.

Документ может представляться в различных видах (например, как множество понятий или вектор в пространстве понятий), чтобы это учесть добавим понятие формы документа $\mathrm{d} \in \mathrm{D}\,\mathrm{R}(\mathrm{d})$. Форма документа и идентификация понятия зависят от определенного алгоритма извлечения понятия из документа [Cormack, 2008].

Основываясь на этой терминологии изучим разработанные алгоритмы определения спама при помощи машинного обучения. Чтобы сравнивать классификаторы используем точность (отношение количества корректно распознанных документов к их общему количеству) и статистику (1-AUC)%, определяющую область под ROC-кривой (выражение AUC = 0,999 как 0,1%, это означает, что чем ближе полученное значение к 0, тем точнее сработал алгоритм) [Hanley, McNeil, 1983].

2.2 Вероятностные классификаторы

Добавим понятие «мягкий» (soft) и «жесткий» (hard) вероятностный классификатор. «Мягкий» вероятностный классификатор принимает функцию вида:

$$c_{soft}(d, c) = p(class(d) = c|R(d) = x), \quad (1)$$

Выше представлена по сути функция, равная вероятности того, что классифицируемый документ *x* относится к спаму. В данном случае термин «soft» равен «fuzzy» и определяет нечетную классификацию. «Жесткий» вероятностный классификатор принимает функцию вида:

$$c_{hard}(d, c) = p(class(d) = c|R(d) = x) > t.$$

В итоге, «жесткий» и «мягкий» классификаторы различается неким пороговым значением $t \in [0; 1)$, а его превышение относит документ в категорию спама (получается бинарное разделение). Самые известные представители этого класса алгаритмов:

- логическая регрессия [Cormack, 2008];
- алгоритм работающие на базе марковских полей [Chahabra, Yerazunis, Siefkes, 2004; CRM114 Notes for the TREC 2005 Spam Track];
- алгоритм «наивной» байесовской классификации [Barder D. 2012; Better Bayeasian Filtering].

Алгоритм «наивной» байесовской классификации. Допустим документ $d \in D$ представлен в виде вектора в пространстве понятий $x^{|d|} = \{x_1, x_2, \dots, x_I\}$. «Наивный» байесовский классификатор идентифицирует класс документа на основе оценки апостериорного максимума (MAP), то есть как наиболее возможный среди всех классов:

$$c_{MAP} \equiv argmax_{c \in C} P(c|x^{[d]}).$$

При обнаружении спама эта формула вводит следующий идентификатор для классификации:

$$c_{MAP} == \{spam, \text{если } c_{soft}(d, spam) > c_{soft}(d, \text{non-spam}), \text{non-spam, иначе.} \quad (2)$$

Используем теорему Байеса к формуле (1):

$$c_{soft}(d, c) = \frac{p(c) * p(x^{[d]}|c)}{p(R(d) = x^{[d]})}$$

Обе вероятности рассчитываются для одного документа и вероятность не влияет на итог сравнения с применением критерия (2), поэтому ее можно исключить:

$$c_{soft}(d, c) = p(c) * p(x^{[d]}|c). \quad (3)$$

Вероятность $p(x^{[d]}|c)$ получают из множества обучающих данных, однако для этого потребуется громадное количество информации [Hovold, 2005]. Чтобы решить эту проблему предлагалось использовать «наивное» предположение, что все термины документа между собой статистически независимы (хотя это в действительности ошибочно). При этом:

$$p(x^{[d]}|c) = \prod_i p(x_i|c). \quad (4)$$

Принимая во внимание формулу (4), формула (3) преобразовывается в такой вид:

$$c_{soft}(d,c) = p(c) * \prod_i p(x_i|c)$$

Главное преимущество представленного классификатора – быстрое обучение [Christina, Karpagavalli, Suganya, 2010], относительно высокая точность (≥98,6% [Christina, Karpagavalli, Suganya, 2010]). Слабость заключается в уязвимости к атакам, где подбираются «хорошие» слова [Lowd, Meek, 2005].

Алгоритм «не столь наивной» байесовской классификации. Со временем были предложены модификации [Su, Xu, 2009] к раннее рассмотренному алгоритму, улучшающие идентификацию спама. Ключевая идея модернизации заключается в аппроксимации статистической связи между понятиями. Приведем следующую вероятность:

$$p\left(x^{[d]}\middle|c\right) = p(x_1, x_2, \dots, x_I|c) = p(x_1|c) * p(x_2, x_3, \dots, x_I|x_1, c)$$
$$= p(x_1|c) * \theta\left(x^{[d]}, x_1, c\right) * p(x_2, x_3, \dots, x_I|c)$$

В этой формуле θ обозначает существующее неизвестное распределение. Теперь, на замену формуле (4) предлагается такая аппроксимация:

$$p\left(x^{[d]}\middle|c\right) = \prod_i p(x_i|c) * \theta(x_i|c). \quad (5)$$

В части этой формулы $\theta(x_i|c)$ i изначально устанавливается в 1 (в результате вся формулы (5) и (4) становятся аналогичными – классическим «наивным» байесовским классификатором). Когда выявляется ошибка в идентификации параметр θ с помощью коэффициентов уверенности α и β корректируется.

Опубликованный анализ в статье [Su, Xu, 2009] показывает, что представленный классификатор отличается повышенной точностью определения спама, быстротой анализа и сравнительно малой требовательностью к использованию памяти. Единственной проблемой считается то, что представленный алгоритм почти не изучен (алгоритм упоминается лишь в статье [Su, Xu, 2009]).

Вероятностный алгоритм Роккио. Классический вариант алгоритма Роккио уступает в точности вероятностным (точность идентификации спама 86% [Joachims, 1997]). В статье [Joachims, 1997] Thorsten Joachims сравнивает классический алгоритм Роккио с предложенной автором вероятностной версией и «наивным» байесовским классификатором. В результате предложенный вероятностный алгоритм Роккио по точности находится на одном уровне с «наивным» байесовским классификатором.

MRF-классификатор. Допустим, что $F = \{F_1, F_2, \dots, F_m)$ – множество случайных значений, распределенные на конкретном дискретном множестве ячеек S, с точностью до показателей параметров, при этом в каждой ячейке случайное значение F_i заменяется на f_i из дискретного множества меток L. Под настройками случайного поля F = f понимаем проявление событий «случайная величина F_i заменена на f_i».

Вероятность того, что случайная величина F_i заменена на f_i для множества меток L обозначим $P(F_i = f_i)$, вероятность проявления событий $(F_1 = f_1, F_2 = f_2, \dots, F_m = f_m)$ обозначим $P(F = f)$. Чтобы решить задачу идентификации спама положение слова в последовательности будем считать одной ячейкой, а метки соотнесем к терминам.

Предположим, что F становится марковским случайным полем над S относительно соседства N (neighborhood) только при выполнении условия:

$$P(f) > 0, \forall f \in F,$$
$$P\left(f_i \middle| f_{s-\{i\}}\right) = P(f_i | f_{N_i}),$$

где $f_{N_i} = \{f_{i'} | i' \in N_i\}$, или множество появляющихся событий по соседству с ячейкой i.

В действительности MRF-классификатор получается с помощью изменения стандартного «наивного» байесовского классификатора [Yerazunis, 2003]. Добиться этого можно следующим образом:

- извлекая термины учитывается не отдельный токен, а вся цепочка;
- принять во внимание, что с увеличением цепочек в большую сторону изменяется вес (подходы к изучению назначения веса цепочек представлены в работе [5].

В работе [Yerazunis, 2003] отмечается исключительная эффективность MRF-классификатора, с точностью от 99,95%. Однако практическая реализация в фильтре спама CRM114, разработчики считают малоэффективной и сложной [CRM114 Notes for the TREC 2205 Spam Track]. В настоящее время требуется дальнейшее совершенствование и тестирование алгоритма.

2.3 Линейные классификаторы

Предположим, что линейный классификатор выражен в векторе из n коэффициентов $\beta = (\beta_1, \beta_2, \dots, \beta_n)$, n = |T|, с некой границей t. В пространстве терминов рассмотрим уравнение плоскости:

$$\beta * x^{|m|} = t$$

Эта гиперплоскость разделит пространство терминов на две части, одна из которых содержит точки, определяемые как спам, а другая – как не спам. Гиперплоскость назовем разделяющей лишь при выполнении следующего условия:

$$(\forall d \in D)\left(\left(\beta * x^{|m|} > t\right) \Leftrightarrow class(m) == spam\right) \& ((\beta * x^{|m|} \leq t) \Leftrightarrow class(m) = \\ = \text{non-spam})$$

Предположим, что множество документов D линейно разделимо, когда есть разделяющая гиперплоскость. Главной трудностью в линейных классификаторах является поиск наилучшей разделяющей гиперплоскости в множестве документов.

Персептрон. Этот алгоритм самостоятельно определяет любую разделяющую плоскость в заданном множестве документов D, если такая существует или множество документов на практике возможно разделить. В ситуациях, когда разделяющей плоскости нет алгоритм без определения максимального количества итераций бесконечно зацикливается. С каждой новой итерацией, при некорректной классификации точек, коэффициенты β изменяется, прибавляя или удаляя некоторые постоянные. Главным преимуществом этого алгоритма является скорость, адаптивность и простота [Cormack, 2008]. К недостаткам относится то, что не всегда полученная гиперплоскость является наилучшим решением для выбранного множества документов.

Алгоритм Winnow. Этот алгоритм [Littlestone, 1988; Siefkes et al., 2004] является аналогом персептрона, но у него есть несколько отличий:

- элементы β могут быть только положительными;
- коэффициенты корректируются не аддитивно, а мультипликативно.

При появлении первой ошибки из списка выше коэффициенты β умножаются на некий коэффициент $\alpha > 1$, соответствующий терминам ошибочно выявленного документа. То же самое происходит и со второй ошибкой, когда коэффициент β умножается на некий коэффициент $0 < \gamma < 1$. Используя ортогонально разряженные биграммы (собирающие основу в пространстве терминов и допускающие наличие других слов между словами биграммы в тексте) [Siefkes et al., 2004] для модификации этого алгоритма можно добиться точности обнаружения спама до 99%.

Алгоритм опорных векторов. Этот алгоритм определяет максимально удаленную гиперплоскость от ближайших точек обучающего массива документов. Разделяющая гиперплоскость высчитывается на базе малого количества точек линейной комбинации опорных векторов, являющихся классификатором. Уже разработаны эффективные реализации алгоритма для выявления спама [Drucker, Wu, Vapnik, 1999; Sculley, Wachman, 2007].

2.4 Классификаторы, основанные на сходстве

Рассмотрим отдельный документ $d \in D$ и множество правильно идентифицированных документов $D' \subseteq D$. Предположим, что задана некая функция расстояния $distsnce: D\ x\ D \rightarrow R$, как метрика на пространстве терминов T, и выполняющая следующие условия:

1) $\forall_{x,y} \in D, distance(x,y) = 0 \Leftrightarrow x = y$;
2) $\forall_{x,y} \in D, distance(x,y) = distance(y,x)$;
3) $\forall_{x,y,z} \in D, distance(x,z) \leq distance(x,y) + distance(y,z)$.

Этот алгоритм подразумевает идею, что документы одного класса в векторном пространстве расположены рядом, а документы разных классов далеко друг от друга.

Алгоритм k-ближайших соседей. Самым простым способом выявления спама является присваивание такого же класса документу, что и у его ближайшего соседа. Основываясь на этой идее наиболее эффективным считается алгоритм k-ближайших соседей [Barber D. 2012; Shakhnarovich, G.Darrell, T., Indyk, 2006]. Он учитывает класс k самых близких к оцениваемому документу соседей и делающий выбор в пользу класса большинства. Этот метод не способен с высокой точностью выявить спам [Cormack, 2008].

2.5 Логические классификаторы

Эта группа базируется на применении аппарата логики для определения связи среди терминов.

Деревья принятия решений. Этот алгоритм [Breiman et al. 1984] во время обучения последовательно разбивает тренировочный набор документов по неким критериям добавляя или исключая один атрибут за раз. Например, таким критерием может быть информационная выгода по Большакову. На практике деревья принятия решений плохо справляются с фильтрацией спама, однако при внедрив некоторые модификации можно получить значительный прирост точности [Cormack, 2008].

Выводы, основанные на наборе правил. По своей продуктивности эти алгоритмы схожи с деревом принятия решений [Cormack, 2008], но уже есть разработанные комбинации [Wu, 2009] подходов с высокой точность выявления спама.

2.6 DCM классификаторы

Примем X как случайную величину, при реализации которой x становится текстом документа из набора D. Пусть документ $d \in D$ будет последовательностью символов случайной длины $x^{|d|} = x_1, x_2, \dots, x_n \in \Sigma^*$, относящихся к алфавиту Σ ($x^{|d|}$ – воплощение случайного значения X).

Модель сжатия данных (Data Compression Model) [Bratko et al., 2006] D определяет информационный состав документа $x^{|d|}$:

$$I_D\left(x^{|d|}\right) = -\log(P_D(x = x^{|d|})),$$

где $P_D(x = x^{|d|})$ является вероятностью сообщения $x^{|d|}$ внутри модели сжатия данных D.

Предположим, что при выполнении условия $I_{D_1}\left(x^{|d|}\right) < I_{D_2}\left(x^{|d|}\right)$ модель сжатия данных D_1 лучше справляется с моделированием сообщения $x^{|d|}$.

Чтобы решить проблему идентификации спама выстроим две модели: D_{spam} и $D_{non-spam}$, в результате получим следующие условия классификации:

$$c(d) = \{spam, \text{если } I_{D_{spam}}\left(x^{|d|}\right) < I_{D_{non-spam}}\left(x^{|d|}\right), \text{иначе non-spam.}$$

Часто применяются итерационные модели: динамическое марковское сжатие (DMC) [Bratko et al., 2006] или предсказание по частичному совпадению (PPM) [Bratko, Filipic, Zupan, 2006; Bratko et al., 2006].

3. Выводы

В существующую ранее классификацию были добавлены «не столь наивный» байесовский и MRF классификатор. Произведено описание математического аппарата всех упомянутых классов и предоставлено подробное описание работы алгоритмов (расширены сведения данных из [Cormack, 2008]). Учтен результат более поздних исследований: например, в статье [Cormack, 2008] алгоритмы логического вывода, основанные на наборе конкретных правил, сравниваются по точности с деревьями принятия решений, в то время как приведенные в работе примеры комбинированного метода позволяют добиться большей точности.

На основе проведенного обзора способов фильтрации спама составлена таблица, показывающая степень точности описанных методик (табл. 1) по показателям точности выявления и мере (1-AUC) %.

Данные таблицы показывают, что наибольшей эффективностью обладают линейный и вероятностный классификатор.

Основываясь на проведенном исследовании можно определить следующие перспективные направления изучения:

- более подробное исследование малоизученных алгоритмов;
- создание комбинированных решений для повышения точности обнаружения спама;
- анализ эффективности уже разработанных алгоритмов на одинаковых корпусах с едиными требованиями определения точности.

В особенности перспективной можно считать комбинацию MRF-классификатора и метод «не столь наивного» байесовского классификатора, так как первый алгоритм эффективно выявляет спам и основан на «наивном» байесовском классификаторе, а второй, являясь модификацией базы MRF, повышает точность фильтрации.

Таблица 1 – Степень точности описанных методик

Класс	Метод	Точность	
		(1-AUC) %	Доля правильно классифицированных документов, %
Вероятностные классификаторы	«наивный» байесовский классификатор	-	90-97 [Christina, Karpagavalli, Suganya, 2010; Better Bayesian Filtering; Joachims, 1997]
	«не столь наивный» байесовский классификатор	0.0344 [Su, Xu, 2009]	-
	вероятностный алгоритм Роккио	-	91 [Joachims, 1997]
	MRF- классификатор	-	99.95 [Yerazunis, 2003]
Линейные классификаторы	персептрон	-	-
	алгоритм Winnow	-	98-99 [Siefkes et al., 2004]
	SVM	0.024 [Sculley, Wachman, 2007]	99 [Drucker, Wu, Vapnik, 1999]
Классификаторы на базе сходства	kNN	0.3056 [Yerazunis, 2006]	-
Логические классификаторы	деревья принятия решений	-	88 [Carreras, Marquez, 2001]
	логические выводы на основе правил	-	92 [Androutsopoulos, Paliouras, Michelakis, 2006]
DCM классификаторы	DMC	0.013 [Bratko et al., 2006]	-
	PPM	0.019 [Bratko et al., 2006]	-

4. Заключение

В статье рассмотрены алгоритмы для фильтрации спама с помощью методов машинного обучения. Для расширения существующей статьи [Cormack, 2008], была предложена сортировка алгоритмов фильтровки спама, по 5 классам соответственно применяемых в рамках их подходов. Сделан анализ описанных алгоритмов с указанием точности фильтрация ими спама. Определены самые перспективные направления исследования и применения на практике методов выявления спама.

Список литературы

1) Cormack G. V. Email spam filtering: A systematic review //Foundations and Trends in Information Retrieval. – 2008. – Vol. 1. – №. 4. – P. 335-455.

2) Спам и фишинг во втором квартале 2016. [Электронный ресурс]. Режим доступа: https://securelist.ru/analysis/spam-quarterly/29116/spam-and-phishing-in-q2-2016/

3) Большакова Е.И., Клышинский Э.С., Ландэ Д.В., Носков А.А., Пескова О.В., Ягунова Е.В. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие – М.: МИЭМ, 2011. – 272 с.

4) Barber D. Bayesian reasoning and machine learning. – Cambridge University Press, 2012.

5) Chhabra S., Yerazunis W. S., Siefkes C. Spam filtering using a markov random field model with variable weighting schemas //Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on. – IEEE, 2004. – Pp. 347-350.

6) CRM114 Notes for the TREC 2005 Spam Track [Электронный ресурс]. Режим доступа http://crm114.sourceforge.net/docs/NIST_TREC_2005_paper.html

7) Hovold J. Naive Bayes Spam Filtering Using Word-Position-Based Attributes //CEAS. – 2005. – Pp. 41-48.

8) Christina V., Karpagavalli S., Suganya G. Email spam filtering using supervised machine learning techniques //International Journal on Computer Science and Engineering (IJCSE). – 2010. – Vol. 2. – Pp. 3126-3129.

9) Lowd D., Meek C. Good Word Attacks on Statistical Spam Filters //In Proceedings of the Second Conference on Email and Anti-Spam (CEAS). – 2005.

10) Su B., Xu C. Not So Naıve Online Bayesian Spam Filter //Proceedings of the Twenty-First Innovative Applications of Artificial Intelligence Conference. – 2009.

11) Better Bayesian Filtering. [Электронный ресурс]. Режим доступа http://www.paulgraham.com/better.html

12) Joachims T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. // Proceedings of ICML-97, 14th International Conference on Machine Learning. – 1997. – Pp. 143-151.

13) Yerazunis W. S. The spam-filtering accuracy plateau at 99.9% accuracy and how to get past it //Proceedings of the 2004 MIT Spam Conference. – 2004.

14) Littlestone N. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm //Machine learning. – 1988. – Vol. 2. – №. 4. – Pp. 285-318.

15) Siefkes C., Assis, F., Chhabra, S., Yerazunis, W. S. Combining winnow and orthogonal sparse bigrams for incremental spam filtering //European Conference on Principles of Data Mining and Knowledge Discovery. – 2004. – Pp. 410-421.

16) Wu C. H. Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks //Expert Systems with Applications. – 2009. – Vol. 36. – №. 3. – Pp. 4321-4330.

17) Bratko A., Filipic B., Zupan B. Towards Practical PPM Spam Filtering: Experiments for the TREC 2006 Spam Track // Proceedings of the 15th Text REtrieval Conference (TREC 2006). – 2006.

18) Bratko A., Cormack, G. V., Filipič, B., Lynam, T. R., Zupan, B. Spam filtering using statistical data compression models //Journal of machine learning research. – 2006. – Vol. 7. – Pp. 2673-2698.

19) Breiman L., Friedman, J., Stone, C. J., Olshen, R. A. Classification and regression trees. – CRC press, 1984.

20) Shakhnarovich, G. Darrell, T., Indyk, P. Nearest-neighbor methods in learning and vision. Theory and Practice. – MIT Press. – 2006.

21) Hanley J. A., McNeil B. J. A method of comparing the areas under receiver operating characteristic curves derived from the same cases //Radiology. – 1983. – Vol. 148. – №. 3. – Pp. 839-843.

22) Drucker H., Wu D., Vapnik V. N. Support vector machines for spam categorization //IEEE Transactions on Neural networks. – 1999. – Vol. 10. – №. 5. – Pp. 1048-1054.

23) Sculley D., Wachman G. M. Relaxed online SVMs for spam filtering //Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. – 2007. – Pp. 415-422.

24) Carreras X., Marquez L. Boosting trees for anti-spam email filtering //In 4th International Conference on Recent Advances in Natural Language Processing. – 2001. – Pp. 58-64.

25) Androutsopoulos I., Paliouras G., Michelakis E. Learning to filter unsolicited commercial e-mail. Technical report 2004/2, National Center for Scientific Research "Demokritos". – 2004.

26) Yerazunis W. S. Seven Hypothesis about Spam Filtering // Proceedings of the 15th Text REtrieval Conference (TREC 2006). – 2006.