# Evaluating Weightless Neural Networks for Bias Identification on News

Rafael Dutra Cavalcanti*, Priscila M.V. Lima*, Massimo De Gregorio† and Daniel Sadoc Menasche*
*Universidade Federal do Rio de Janeiro - PPGI
Rio de Janeiro - Brazil
†Istituto di Scienze Applicate e Sistemi Intelligenti - CNR
Pozzuoli (NA) - Italy

***Abstract*—Identifying biases in articles published in the news media is one of the most fundamental problems in the realm of journalism and communication, and automatic mechanisms for detecting that a piece of news is biased have been studied for decades. In this paper, we compare the WiSARD classifier, a lightweight efficient weightless neural network architecture, against Logistic Regression, Gradient Tree Boosting, SVM and Naive Bayes for identification of polarity in news. Motivated by the fast pace at which news feeds are published, we envision the increasing need for efficient and accurate mechanisms for bias detection. WiSARD presented itself as a good candidate for the task of bias identification, specially in dynamic contexts, due to its online learning ability and comparable accuracy when contrasted against the considered alternatives.**

## I. Introduction

Modern society is arguably becoming more polarized. A recent declaration by the pope decries the "virus of polarization" [1]. In the same week, the president of the United States tried to nail the root causes of such recent surge in polarization [2]. In his speech, he pointed the widespread publication of biased articles in news media and social networks as one of the key drivers of polarization. Indeed, identifying covert bias disguised in the news is an important challenge to be addressed in the pursuit of a more transparent and harmonic society.

In this paper, our goal is to automatically classify articles based on their political bias. In particular, we focus on recent news about Brazil, collected from two major sources. As the identification of biases is subjective, to have a solid ground-truth the selected publishers were the Brazilian Democratic Movement Party (PMDB) and the Workers Party (PT). These two parties are currently two of the major players in Brazilian politics, the first being responsible for the impeachment of the former president belonging to the latter.

We take a broad perspective on the meaning of "bias". In particular, one of our working assumptions consists of assuming that the two sources considered in this paper generate two "biased" news feeds. Our classification results account for the style of writing of the two sets of authors and differences in the vocabulary they typically use. Therefore, we frame the bias identification problem as a problem of source recognition.

Our problem consists of identifying, for each of the articles in the selected database, whether it appeared in the website of PMDB or PT. We address the following two questions:

- is it feasible to automatically classify the sources of articles on politics?
- what are the advantages and disadvantages of the classification tools, with respect to accuracy and efficiency/performance?

We provide an affirmative answer to the first question, and identify weightless neural networks (WNN) as simple and efficient tools to perform the classification. Note that not all the articles published in the websites of the major Brazilian parties are biased. In addition, even when the texts are biased they may do so in such a way that the bias is covert. Therefore, identifying the feasibility of classifying the sources solely based on the text content, with accuracy greater than 90%, is our major contribution.

We performed a comparative study of different classification tools to identify the most appropriate for news classification. SVM [3] and Naive Bayes [4] have been considered in [5], [6]. Entropy maximization, coupled with SVM, has been studied in [7], for articles on sports. By its own nature, the classification of articles in the news needs to be performed in a continuous and efficient manner. As the rate of content generated rapidly increases, even if the classification is not required to be done online still efficiency is key. For this reason, we identify the WiSARD WNN as one of the promising solutions for the purpose of source classification. In addition, we also indicate logistic regression [8] as an alternative competitive candidate, outperforming gradient tree boosting [9], SVM and naive Bayes. Model parameterization is conducted using 10-fold cross validation [10], and we report comparisons among models using measures of accuracy mean and dispersion.

The remainder of this article is organized as follows. First, we introduce basic background on weightless neural networks in Section II, followed by the methodology adopted in this paper in Section III. Then, we report our experimental results and conclusions in Sections IV and V, respectively.

## II. Basic Concepts

In this section we introduce the basic tools used in our experimental analysis. We start with the WiSARD classifier,

978-1- 5090-4429- 0/17/$31.00 

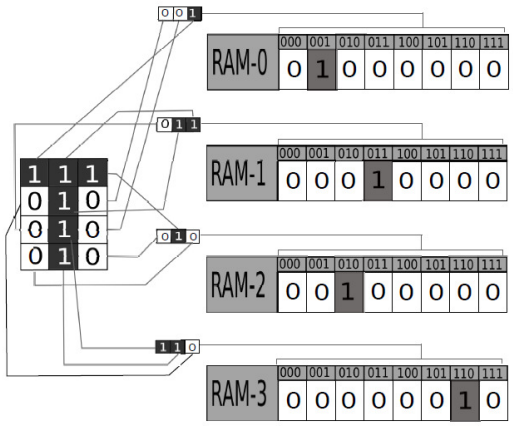


Fig. 1. Training a discriminator [11]

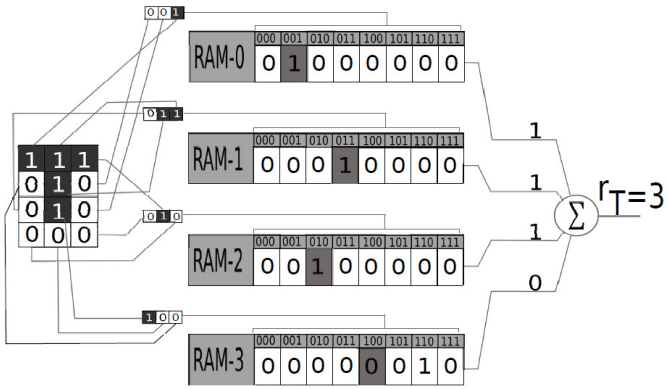


Fig. 2. Discriminator associated to class "T" [11]

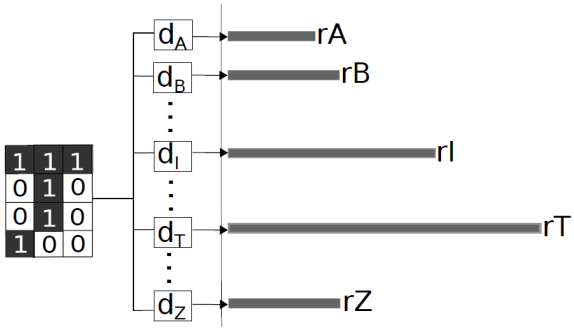


Fig. 3. Discriminator response on an instance of a "T" [11]

which is our reference model of weightless neural networks. Then, we present a brief overview of logistic regression, SVM, naive Bayes, gradient boosting and terms weighting.

### A. The WiSARD classifier

The weightless neural system adopted in this work is called WiSARD (Wilkie, Stonham & Aleksanders Recognition Device) [12]. The input to these systems is usually formed by binary patterns and, in those cases where the data is not in this format, it needs to be converted ("binarized") [13].

The WiSARD is a multi-discriminator system because it is formed by as many discriminators as the number of classes it has to work with. Each discriminator is formed by a set of RAM-nodes and it is responsible for the recognition of object belonging to the class it is trained with [14]. In the model here adopted, the content of a RAM-node (RAM neuron) is incremented by 1 for each sample of the training set, as can be seen in the figure 1. At the end of the training phase, the RAM-neuron memory contents may vary between 0 and the cardinality of the training set. Zero is the initial value set before training the system. Each RAM neuron maps a set of $n$ bits ($n$-tuple) pseudo-randomly extracted from a binary input pattern, also called retina. See Figures 2 and 3 for a discriminator trained on an instance of the class "T".

In the classification phase, upon a read stimulus (input), a RAM neuron outputs the value of the accessed RAM cell. If $n^\star$ neurons output 1, then $r = n^\star/n$ is the discriminator response (where $n$ is the number of RAM neurons in a discriminator).

Whenever the amount of data or the cardinality of the training set is quite big, it is possible that the RAM neurons can quickly reach the saturation in the training phase (all the memory cell contents are different from 0). To avoid this problem, we added a firing condition to RAMs that is: the RAM neuron outputs 1 if the value of the accessed RAM cell is greater than a threshold, namely $\beta$ (bleaching), otherwise the RAM neuron outputs 0. There exists two different kind of bleaching: homogeneous and heterogeneous. The former refers to the use of the same $\beta$ value for all discriminators. The latter refers to the use of a $\beta$ value proportional to the knowledge degree of each discriminator. There is no way of automatically choose in advance the kind of bleaching to use in the system [14].

### B. Logistic Regression

In both simple and multiple linear regression models, the dependent variable is a random continuous variable. However, some models require that variables be of a qualitative (discrete) nature. A good approximation to encompass this characteristic can be obtained by logistic regression that uses regression to calculate or predict the probability of a specific event [15].

Though similar to linear regression, the logistic regression model, in its simplest form, determines that the dependent variable be binary. The binary variable assumes two values, usually, 0 and 1, respectively named, in this work, as "Class 1" and "Class 2". In our case, the event of interest is to know the classification of a determined piece of text.

Following the notation of [16], given a set of $p$ independent explanatory variables, we organize the $n$ provided data points

into matrix $\mathcal{X}$,

$$\mathcal{X} = \begin{pmatrix} x_{10} & x_{11} & x_{12} & \dots & x_{1p} \\ \vdots & & & & \\ x_{i0} & x_{i1} & x_{i2} & \dots & x_{ip} \\ \vdots & & & & \\ x_{n0} & x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix} \quad (1)$$

where $\mathbf{x}_i^T = (x_{i0}, x_{i1}, x_{i2}, ..., x_{ip})$ is the $i$-th datapoint vector, equal to the $i$-th line of matrix $X$, for $i = 1, 2, \dots, n$ and $j = 0, 1, \dots, p$. We assume that $x_{i0} = 1$ for all $i$.

We denote by $\tilde{\boldsymbol{\beta}} = (\tilde{\beta}_0, \tilde{\beta}_1, \dots, \tilde{\beta}_p)^T$ the vector of unknown parameters, where $\tilde{\beta}_j$ is the $j$-th parameter, associate to the $j$-th explanatory variable.

Let $X_i$ and $Y_i$ be two random variables denoting the given vector of features (independent explanatory variables) and the outcome (dependent variable). In the multiple logistic regression model, the probability of success is given by

$$\pi_i = \pi(\mathbf{x}_i) = P(Y_i = 1 | X_i = \mathbf{x}_i) = \frac{\exp(\mathbf{x}_i^T \tilde{\boldsymbol{\beta}})}{1 + \exp(\mathbf{x}_i^T \tilde{\boldsymbol{\beta}})}.$$

The probability of failure is given by $1 - \pi_i$, and the logarithm of the likelihood function is given by

$$\mathcal{L}(\tilde{\boldsymbol{\beta}}) = \sum_{i=1}^{n} y_i \mathbf{x}_i^T \tilde{\boldsymbol{\beta}} - \ln(1 - \mathbf{x}_i^T \tilde{\boldsymbol{\beta}}).$$

The value that maximizes $\mathcal{L}(\tilde{\boldsymbol{\beta}})$ is obtained by deriving $\mathcal{L}(\tilde{\boldsymbol{\beta}})$ with respect to each of its parameter. Unlike linear regression, the solution of the resulting optimization problem is not amenable to a closed-form expression. An iterative process must be used to find the coefficient values $\tilde{\boldsymbol{\beta}}$ that maximize the likelihood. In the experimental results reported in this paper, we consider the default iterative process used by `scikit-learn` [17], as implemented at `liblinear` [18].

### C. Support Vector Machine Classifiers

Support Vector Machines (SVM) were first proposed to solve pattern recognition problems. Data is mapped into a higher-dimensional space and a hyperplane is built so as to optimally separate points of the two classes [19]. The elements that define the hyperplane are called the *support vectors*. Finding the optimal set of support vectors, i.e., the hyperplane that optimizes separation of the two classes in question, constitutes the goal of the algorithms related to SVM. Such algorithms must be computationally efficient, typically being able to deal with samples of 100.000 instances [20].

In the linearly separable case, the key idea behind a SVM is quite simple. Given a training set $S$ containing points of the two classes, the SVM separates them by a hyperplane, determined by a certain subset of points of $S$ (the support vectors). In the separable case, that hyperplane maximizes the *margin*, defined as twice the minimum distance of each class to the hyperplane. All the support vectors must comply with minimum margin, being called margin vectors. In real-life applications, the two classes may not be linearly separable, so the hyperplane and support vectors are obtained as solution to a problem of optimization with constraints. Such a compromise solution is controlled by a regularization parameter involving higher margin and least number of errors [21].

To separate both linearly and non-linearly separable datasets, a key ingredient to the SVM methods is a kernel function. Using a kernel function the SVM builds a decision surface that is nonlinear with respect to the input space, but is linear with respect to the attribute space. Some of the following kernel functions have been used by SVM methods [22]:

- linear:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j \quad (2)$$

- radial basis function (RBF):

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2\gamma^2} \parallel \mathbf{x}_i - \mathbf{x}_j \parallel^2\right), \gamma > 0 \quad (3)$$

- polynomial:

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + r)^d, \gamma > 0 \quad (4)$$

- sigmoidal:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i^T \mathbf{x}_j + r). \quad (5)$$

Small values of $\gamma$ make the surface quasi-linear, while increasing its value increases flexibility of the decision surface. Parameter $r$ controls the displacement threshold in the cases of polynomial and sigmoidal kernels. Parameter $d$ gives the degree of the polynomial kernel. All parameters must be specified a priori by the user.

### D. Naive Bayes

Based on the Bayes Theorem, the Naive Bayes classifier is one of the most commonly used for text categorization [23]. The technique enables the computation of the probability that a certain document belongs to a class, by using the conditional probabilities that some words appear in the document. The class with higher probability is chosen for the document [24].

A Bayesian classifier considers that the position probability of words in the text is independent from their actual position. It also considers that the probability of a set of words in an example is equivalent to the product of the probabilities of each isolated attribute (word). This way, the basic formulation in the Naive Bayes technique is given by:

$$V_{NB} = \text{argmax}_{v_j \epsilon V} \prod_{a_i \in \mathcal{D}} P\left(a_i | v_j\right) P(v_j) \quad (6)$$

where $V_{NB}$ is the estimated category, $V$ is the set of categories, $v_j$ is one of the categories that belongs to $V$, $a_i$ is a word in document $\mathcal{D}$, $P(v_j)$ is the prior probability of $v_j$ and $P\left(a_i | v_j\right)$ is the probability of occurrence of $a_i$ in category $v_j$.

### E. Gradient Tree Boosting

Decision trees are structures that recursively partition the input space, creating disjunct regions associated to terminal nodes (representing the classes). It is possible to enhance the predictive power of a learning model through a committee. That is the central idea of boosting techniques which also follow a sequential strategy, seeking to gradually reduce prediction residues by using errors made in the previous step.

The Gradient Boosting(GB) algorithm consists of an additive iterative process that initiates with constant predictions whose values correspond to the "mean answer" associated to the training sample: $f_0(x) = \bar{y}$. At each iteration, a new term is added to the current model aiming to reduce prediction error. These updates follow the inverse direction of the goal function gradient $\Psi(y_i, f(x_i))$ with respect to the current approximations, $f(x_i)$. The process is repeated until a stop condition, such as maximum number of iterations, is met [25].

### F. Terms weighting

Usually, the contents of a document are described by a set of terms, referred to as the document *indices*. However, we notice that not all of the terms are equally useful, or representative, for that description. Some terms are more vague than others, or pertain to the core subject but occur in most of the documents in the same collection. The determination of terms importance in the indexing of a document is not an easy task. So, terms weighting techniques are used to determine which terms are more important in a given application [26]. This work makes use of two objective properties of a term that are relatively easy to measure: the term frequency (TF) in the document and the inverse document frequency (IDF) [27]. The composition of these two measures constitutes the TF-IDF joint measure, which aims towards selecting terms that are (i) frequent in a certain subset of documents while (ii) occurring less frequently in the remainder subset.

## III. Methodology

### A. Dataset construction

Web information collectors, such as crawlers or spiders, automatically traverse the Web, storing the pages visited as well as their resources. Their goal is to find the greatest number of pages of interest with the least computational cost. When driven by the page contents, they are classified as specific or focused [28].[1]

Since no repositories of Brazilian political news of January to July of 2016, labeled as either pro or against the government, were available, crawlers were used to build such repository. In order to train the system with examples of the first class, pro-government, 207 news from the site of the PMDB party (Partido do Movimento Democrático Brasileiro) were collected at http://pmdb.org.br/. Training examples of the second class, government opposition, was accomplished by collecting 173 news from the PT party (Partido dos Trabalhadores), http:// www.pt.org.br/. The main idea behind such strategy was that each party would present its own point of view, being consistently biased.

[1]All the scripts and datasets used to obtain the results presented in this paper are available at http://sites.google.com/view/wisardbiasidentification.

### B. Text pre-processing and accuracy

At the pre-processing stage, documents, considered as raw text without formatting, are manipulated so that their new representation is more adequate to the execution of the intended task [29]. Besides the conversion of text to a sequence of words, the normalization and the removal of punctuation and accentuation, this procedure, also known as indexing [30], involves the following phases:

1) **Stop words elimination:** according [27], highly frequent words in the documents of a collection are not good discriminants. Their removal reduces considerably the structures in the dataset.

2) **Stemming:** syntactic variations of words such as gerunds, plurals and suffixes may hinder the recognition of words with similar semantic. Therefore words were substituted by their semantic core, called stem [27].

3) ***N*-grams representation:** proposed in [31], it constitutes an alternative form of text representation that aggregates terms in groups of size *n*. It enhances the confidence on the recognition of sentiment, independently on the language under analysis. In the case of this work, $n = 2$.

4) **TF-IDF:** application of TF-IDF to select the groups of terms [27].

5) **Bag-of-words:** we conclude with the representation using the bag-of-words approach [32].

The accuracy of this work adopted the *k*-fold cross validation [10], with $k$=10. The set was divided into $k$ subsets, where $k$-1 subsets were used for training while the remaining subset was used for validation. This process was repeated 30 times and the standard deviation calculated.

## IV. Experimental results

The experimental platform used in this work was based on Python, version 2.7. We made use of the WiSARD libraries provided at [33]. All other classifiers were executed using the corresponding `scikit-learn` implementation [17]. After the acquisition and categorization of the documents from the parties sites, their pre-processing was accomplished by the Portuguese module of the NLTK library [34]. Text was represented as bag-of-words, where order is not considered.

In natural language processing (NLP) applications, it is common to encounter, at first, a large number of features. This work did not exception, having an initial figure of 12500 features. In order to reduce this number of features and to enhance the semantic distinction of terms, two techniques were

applied. The first was to group the words in bigrams (therefore $n$=2) to strengthen semantic information thus improving chances of a good classification. The second strategy adopted was to apply the technique of TF-IDF attribution of weights to terms, so that only the most significant were selected, thus reducing dimensionality. An empirically determined threshold $\theta$ was then applied to build the binary input to WiSARD: if the term weight was greater or equal to $\theta$ than the value considered would be 1 and 0 otherwise. The idea was to distinguish the terms that would be considered important to the piece of news in question from the others bigrams.

TABLE I
ACCURACY(ACC) AND STANDARD DEVIATION(SD)

| Model | Unigrams | | Bigrams | | Trigrams | |
|---|---|---|---|---|---|---|
| | Acc | SD | Acc | SD | Acc | SD |
| Logistic Regression | 88% | 4% | 95% | 2% | 94% | 3% |
| WiSARD 4 bits with bleaching | 88 | 5 | 93% | 3% | 95% | 2% |
| WiSARD 8 bits with bleaching | 88 | 5 | 93% | 3% | 94% | 3% |
| SVM (kernel linear) | 87 | 4 | 93% | 4% | 92 | 4 |
| WiSARD 16 bits with bleaching | 89 | 6 | 93% | 6% | 93 | 4 |
| WiSARD 16 bits | 89 | 4 | 92% | 5% | 93 | 3 |
| WiSARD 8 bits | 90 | 6 | 91% | 3% | 94 | 3 |
| WiSARD 4 bits | 91 | 4 | 91% | 4% | 94 | 3 |
| GB(estimators=150, learning rate=0.01, depth=5) | 83 | 8 | 86% | 4% | 85 | 5 |
| Bernoulli Naive Bayes | 80 | 5 | 85% | 5% | 86 | 5 |
| WiSARD 32 bits with bleaching | 83 | 6 | 84% | 4% | 93 | 3 |
| WiSARD 32 bits | 82 | 6 | 82% | 6% | 93 | 4 |

Table I depicts a comparison between the performance of some variations of WiSARD architectures (having 4, 8, 16 or 32 address bits), with and without bleaching, and some classifiers present in the literature: Logistic Regression [8], Gradient Tree Boosting [9], SVM [3] and Naive Bayes [4]. WiSARD showed comparable results to the other techniques both in accuracy and in standard deviation. The most successful WiSARD architecture, 4 address bits with bleaching, outperformed Naive Bayes and Gradient Tree Boosting, had similar performance to SVM. Only Logistic Regression modeled input/output pairs better than WiSARD.

Although logistic regression presented best accuracy results, WiSARD is also an important candidate to address the bias detection problem. WiSARD is naturally flexible to alternate between training and classification phases. This allows for online learning, i.e., the training set does not have to be retrained as a whole when new observations arrive. Besides, this flexibility facilitates adjustment mechanisms. For example, the training of misclassified events is straightforward under the WiSARD model. Semi-supervised learning is also feasible [35]. These features, that combine flexibility and speed, make WiSARD a suitable and promising paradigm to tackle tendency changes in news feeds and social networks.

## V. CONCLUSION

This article studied the possibility of bias identification in Brazilian political news with the weightless neural model WiSARD as well as with other models present in the literature. WiSARD with 4 and 8 address bits with bleaching had better accuracy than most of the techniques used for comparison, being outperformed only by Logistic Regression. Therefore WiSARD presented itself as a good alternative for the task of bias identification in news, specially in dynamic contexts, due to its online learning ability.

It is important to note that the construction of the training set requires that the features collected from each party belong to the same time period. This is justified by the possibility that positions from a political party change according to changes of the country's context. Besides, it is desirable that training sets of different classes be balanced, to avoid bias in the classification, even though some techniques may be employed to diminish this problem.

Work on further dimensionality reduction and the insertion of more than two classes constitute next steps of our research. Another topic under investigation is the application of the method to other news subjects such sports, economy and entertainment.

## REFERENCES


[1] "Pope denounces virus of polarization," http://www.csmonitor.com/World/2016/1119/Pope-denounces-virus-of-polarization-while-welcoming-new-cardinals, accessed: 2016-11-25.

[2] "Obama nails why the political climate is so polarized," https://boingboing.net/2016/11/25/beyond-fake-news-the-constr.html, accessed: 2016-11-25.

[3] V. N. Vapnik and V. Vapnik, *Statistical learning theory*. Wiley New York, 1998, vol. 1.

[4] H. Zhang, "The optimality of naive Bayes," *AA*, vol. 1, no. 2, p. 3, 2004.

[5] M. A. d. R. d. Santos *et al.*, "Detecção automática da polaridade em notícias sobre política," *Technical report*, 2012.

[6] G. D. d. Arruda, "Análise de viés em notícias na língua portuguesa," Ph.D. dissertation, Universidade de São Paulo, 2015.

[7] R. C. C. Zaccara, "Anotação e classificação automática de entidades nomeadas em notícias esportivas em português brasileiro," Ph.D. dissertation, Universidade de São Paulo, 2012.

[8] D. W. Hosmer Jr and S. Lemeshow, *Applied logistic regression*. John Wiley & Sons, 2004.

[9] T. G. Dietterich, A. Ashenfelter, and Y. Bulatov, "Training conditional random fields via gradient tree boosting," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 28.

[10] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," in *Encyclopedia of database systems*. Springer, 2009, pp. 532–538.

[11] H. C. Carneiro, F. M. França, and P. M. Lima, "Multilingual part-of-speech tagging with weightless neural networks," *Neural Networks*, vol. 66, pp. 11–21, 2015.

[12] I. Aleksander, W. Thomas, and P. Bowden, "Wisard: a radical step forward in image recognition," *Sensor review*, vol. 4, no. 3, pp. 120–124, 1984.

[13] D. de Oliveira Cardoso, "Uma arquitetura para agrupamento de dados em fluxo contínuo baseada em redes neurais sem pesos," Ph.D. dissertation, Universidade Federal do Rio de Janeiro, 2012.

[14] L. C. Bandeira, "Nc-wisard: Uma interpretação sem pesos do modelo neural neocognitron," Ph.D. dissertation, Universidade Federal do Rio de Janeiro, 2010.

[15] C. V. Figueira, "Modelos de regressão logística," *Technical report*, 2006.

[16] D. HOSMER and S. Lemeshow, "Applied logistic regression, new york, john willy & sons, 1989," *GIMENO, SG A; SOUZA, JMP Utilização de estratificação e modelo de regressão logística na análise de dados de estudos casocontrole. Rev. Saúde Pública*, vol. 29, pp. 283–9, 1995.

[17] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[18] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of machine learning research*, vol. 9, no. Aug, pp. 1871–1874, 2008.

[19] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[20] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[21] M. Pontil and A. Verri, "Properties of support vector machines," *Neural Computation*, vol. 10, no. 4, pp. 955–974, 1998.

[22] S. Haykin and N. Network, "A comprehensive foundation," *Neural Networks*, vol. 2, no. 2004, p. 41, 2004.

[23] A. McCallum, K. Nigam *et al.*, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, vol. 752. Citeseer, 1998, pp. 41–48.

[24] D. D. Lewis and M. Ringuette, "A comparison of two learning algorithms for text categorization," in *Third annual symposium on document analysis and information retrieval*, vol. 33, 1994, pp. 81–93.

[25] V. T. d. M. Mayrink, "Avaliação do algoritmo gradient boosting em aplicações de previsão de carga elétrica a curto prazo," *Technical report*, 2015.

[26] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[27] R. Baeza-Yates and B. Ribeiro-Neto, *Recuperação de Informação: Conceitos e Tecnologia das Máquinas de Busca*. Bookman Editora, 2013.

[28] H. Dong, F. K. Hussain, and E. Chang, "A survey in semantic web technologies-inspired focused crawlers," in *Digital Information Management, 2008. ICDIM 2008. Third International Conference on*. IEEE, 2008, pp. 934–936.

[29] K. S. Jones, *Readings in information retrieval*. Morgan Kaufmann, 1997.

[30] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.

[31] W. Cavnar, "Using an *n*-gram-based document representation with a vector processing retrieval model," *NIST Special Publication*, pp. 269–269, 1995.

[32] E. T. Matsubara, C. A. Martins, and M. C. Monard, "Pretext: Uma ferramenta para pré-processamento de textos utilizando a abordagem bag-of-words," *Relatório Técnico (available at Research Gate)*, vol. 209, 2003.

[33] "Python weightless artificial neural network," https://github.com/firmino/PyWANN, accessed: 2016-01-20.

[34] V. M. Orengo and C. R. Huyck, "A stemming algorithmm for the portuguese language." in *SPIRE*, vol. 8, 2001, pp. 186–193.

[35] F. Rangel, F. Firmino, P. M. V. Lima, and J. Oliveira, "Semi-supervised classification of social textual data using WiSARD," *Technical report (UFRJ)*, 2016.