A Cognitive Framework for Detecting Phishing Websites

S. Elnagar[1], **and M. Thomas**[1]
[1]Information Systems, Virginia Commonwealth University, Richmond, Virginia, USA

Abstract - *Phishing over internet is a permeating threat that represents fifth of online business websites. Despite the extensive research in phishing websites detection, none cope with the continuous development in phishing techniques. Therefore, a cognitive, dynamic, and self-adaptive phishing detection system is needed to automatically detect new phishing strategies. Cognitive Computing techniques mimic the reasoning and learning abilities of human brain. In this paper, we propose a cognitive framework for phishing websites detection. The framework uses a cognitive network called a bidirectional long short-term memory (BLSTM) recurrent neural network (RNN). In addition, we integrated a Convolutional Neural Network (CNN) for semantically identifying objects and actions in websites' images. Existing phishing website detection systems suffer from poor image features performance as they use only statistical and structural features of images. The framework is supposed to outperform existing systems because it can learn from context continuously detect new phishing techniques.*

Keywords: Cognitive Computing, Deep Learning, Recurrent Networks, Phishing detection, Convolution Networks

1 Introduction

Phishing is illegal deception techniques that utilize a combination of social engineering and web technologies to steal sensitive personal information such as passwords and credit card details [1]. Phishing attacks have been relying on deception, diversion, and exploitation of lack of user knowledge [2]. The estimated theft through phishing attacks costs U.S. banks and credit card companies $2.8 billion annually [3]. Phishing websites are a serious problem because of the increased number of phishing websites that use intelligent strategies to deceive internet users.

Generally speaking, phishing websites fall into two groups: spoof and concocted websites [4]. Spoof sites are imitations of existing commercial websites such as: eBay, PayPal, and banking service [5], While concocted sites offer fake goods or services to internet users. They are attempting to appear as unique, legitimate commercial entities (e.g., shipping companies, investment banks, online retailers, etc.) [6].

Phishing detection systems are proactive or reactive [7]. Existing phishing detection methods can be divided into four categories: URL blacklist-based method, the visual similarity-based method, the URL and text feature-based method, and the third-party search engine-based method. Most of current methods are proactive that use a combination of features such as the URL and text features, image, linkage, and source code feature. These extended features set is called fraud cues [8, 9]. The reactive detection systems rely solely on user-reported blacklists of fake URLs.

Another reactive approach is the Anti-Phishing training for end-users [3, 10]. However, end users training is costly and requires human administration. Given the adversarial nature of phishing website detection, there has been a notable progress achieved by machine learning classifiers, yet they need constant revision to keep up with the evolving dynamic nature of Phishing websites [11].

Image features play an important role in phishing websites detection as phishing websites reuse images from original sources or other phishing websites. Spoof sites copy company logos from the original websites and concocted websites reuse images of products with the same file name and size [12]. However, image features or cues have low detection power in comparison with other fraud cues because image cues are extracted from image metadata such as: file name and size, or from statistical features such as: pixel color frequencies [13]. One approach to enhance image cues performance is to develop a tool that can recognize objects and actions inside images.

Cognitive computing aims to develop a coherent, unified, and universal mechanism inspired by human mind's capabilities [14]. Cognitive computing is the third and the most transformational phase in computing's evolution, after the Tabulating Era and Programming Era. It is inspired by human's reasoning and problem-solving mechanisms [15]. Cognitive computing is not a single thing but a compendium of capabilities, technologies, resources, and services such as deep learning, speech and vision capabilities, high-performance cloud computing, and parallel low power computing [16].

One of the powerful cognitive neural networks is the Recurrent Neural Networks (RNNs) in particular, the Bidirectional Long Short-Term Memory (BLSTM) type which overcomes the vanishing gradient problem of traditional RNNs. LSTM introduces a memory cell that is controlled by input, output, reset operations, and bidirectional processing [17]. BLSTM RNN can learn when to store or relate to context information over long periods of time. It is still one of the best regression models obtaining remarkable performance in affective computing [18].

In addition to RNN, Deep Convolutional Neural Networks (CNN) have recently shown outstanding image recognition performance in large-scale visual recognition applications. CNNs are multilayer neural networks inspired from the animal visual cortex [18]. Success of CNNs is attributed to their ability to learn rich semantic mid-level image representations [19, 20].

However, CNNs is trained by millions of parameters and requires a large number of annotated image sample [20].
The main contribution of this research is to build a cognitive framework that dynamically learn from domain knowledge to detect phishing websites using a cognitive classifier that employ human cognitive behavior. It uses a set of fraud cues to train a bidirectional long short-term memory recurrent neural network (BLSTM-RNN). In addition, Convolutional Neural Networks (CNN) is used to enhance the performance of image cues in phishing websites detection. CNN generates sematic image cues by detecting objects and actions in website images.

2 Related work

From classification methodology perspective, Phishing detection systems can be categorized into Lookup, Rule-based heuristics, Visual similarity, and Machine Learning-based classifiers. However, the best performing anti-phishing tools use Machine Learning techniques, as they achieve high detection accuracy for analyzing similar data parts to those of rule-based heuristic techniques [21]. A comparison of phishing detection systems in terms of detection methodologies is shown in Table 1.
For machine learning based detection systems, Abbasi et al. [13] proposed the AZProtect classifier system which uses Support Vector Machine (SVM) to detect fraud for both spoof and concocted website. They used a set of heuristics and meta heuristic cues to train the SVM. Mao et al. [22] created a prototype that used cascading style sheet (CSS) as the basis to accurately quantify the visual similarity of each website page element as attackers usually reuse some or all CSS properties in the original CSS. This system targets spoof websites and wasn't applied to concocted websites.
Zhang et al. [4] proposed a model for detecting phishing in e-Business websites which uses unique domain features of Chinese e-Business websites in addition to a set of URLs based features. They built the system model with four different machine learning algorithms. In their experiment, the Sequential Minimal Optimization (SMO) model achieved the best performance. Moreover, sensitivity analysis demonstrated that the domain-specific features have the best detection performance.
Williams et al. [2] developed a computer model that simulates human behavior with respect to phishing website detection based on the ACT-R cognitive architecture which possesses strong capabilities that map well onto the phishing use case. The Feed Phish [23] detects phishing sites based on automation of human behavior for submitting sensitive information. The system uses fake credentials to log into the system before using real ones. It uses URL based features in addition to heuristic features. It neither depends on third-party services nor needs any prior knowledge of websites.

Table1. a comparison of phishing detection systems.

Category	Examples	Strength	Weakness
Lookup Black List	-Google's safe browsing API	-Easy to implement	-High false negative rates. -Detection Limited to websites on the list
Based Detection	-EarthLink Toolbar -FirePhish	-Low cost	
URL And Content Based Detection	-SpoofGuard -CANTINA -GoldPhish	-Reasonable detection rate	-Noises can be added to web page texts -Not dynamically updated
End user training	-APTIPWD -Anti-Phishing Phil	-Easily implemented	-Costly -User perception may be biased. -Require human administration

3 The Phishing Detection Framework

Based on our review of the existing phishing detection systems, we have identified the crucial characteristics a phishing detection system must possess. It should [13]: exhibit the ability to generalize across diverse phishing websites, leverage important domain-specific features such as: stylistic similarities and content duplication. In addition, it should provide long-term sustainability against dynamic adversaries by adapting to changes in the properties exhibited by phishing websites. The first step in building the framework is the feature selection. Selecting a representative set of features has the greatest impact on the accuracy of detection.

3.1 Feature selection

In early works, the researchers identified many fraud cues for detecting phishing websites which can depart into two main types: page information features and external resource features [24]. The page information features use all page related information to verify whether the page is a phishing or not. On the other hand, the external resource features consult a third party to recognize evidences of phishing. Selecting strong fraud cues that depend mainly on domain knowledge will enhance phishing detection performance.
In developing the framework, we used some of the widely used content-based features such as: analyzing Java script, source code, and word phrases (e.g. outdated copyrights and "pay by phone"). Phishing websites usually have many spelling and grammatical mistakes and long URLs [3]. Another page content feature is the similarity score between pages' contents. Phishing websites usually uses similar or even the same text content to its target webpage in order to lure their visitors.
For external resources features, we have added some new features that are expected to effectively enhance the detection performance. Any website contains "About us" page which contains emails, phone and location information. We can check the validity of the phone number, the domain name and the business address using phone directories, maps, and search engines. Moreover, as most of businesses now have online ratings and reviews, the existence for reviews may prove the business legitimacy.
Finding out the business focus of a website by text mining of website pages and meta tags, will help the RNN classifier to rapidly recognize its authenticity by comparing them to websites of similar focus. There are many cognitive computing services that can simply and accurately identify the focus of a

website such as the text analytics APIs of Microsoft cognitive service or IBM Watson. These services are trained with millions of documents and achieve high text mining accuracy. In addition, Site maps can be a useful verification of website legitimacy. The deeper a sitemap hierarchy, the less probability that the site is fraud. A recent page content feature is CSS similarity as attackers usually reuse some or all CSS properties in the original CSS [22]. One of the widely used semantic cues is TF-IDF which assesses document's words importance by assigning them weights and counting their frequency [25]. However, this method efficacy depends on the precision of the top five keywords selected. Therefore, we will slightly replace this TF signature with the business focus feature. Target Identification (TID) algorithm [26] not only identify fraud, but also identify the phishing target. The algorithm identifies all the direct and indirect links associated with the webpage under scrutiny to identify the target domain. Adding the three features of: webpage content similarity, TID, and TF-IDF forms the Semantic Link Network (SLN) of the suspicious webpage. Reasoning of SLN is to discover the implicit semantic relations of any two resources [27] which is the summation of the indirect relations for all possible paths between the two resources.

3.2 Image features using CNNs

Images are essential content in any website, and semantic image recognition will definitely enhance phishing detection. Attackers use the original site's images with slight modifications. Unfortunately, image cues have the worst performance in phishing detection systems because attackers change the image's gradient, colors, resolution, and size to make it hard to recognize by detection systems. However, they can't change the content of the images. Cognitive image recognition services use content features to recognize images, while traditional image recognition techniques use structural and statistical features such as SIFT descriptors, HoG, and moments. However, they don't provide objects representation of image [28].

Cognitive image recognition algorithms recognize image's objects, faces, actions, and give semantic description to the image. One of the most powerful semantic image recognition algorithms is CNN. It can be trained with millions of images with much fewer connections, preprocessing, and parameters than ordinary neural network [29]. Most of CNNs require GPU computation to support high dimensional parallel processing. There are many deep learning open source CNN libraries, which are already trained with millions of images such as Cuda, ConvNet, Torch, Theano, and Caffe [30]. Since the proposed phishing detection system targets different commercial activities, we will use a universal image recognition network. Moreover, since image cues are an integral part of our system, we would reuse existing CNN libraries or APIs such as Clarifai, IBM Watson, and Microsoft cognitive services to retrieve image semantic features.

3.3 Preprocessing

Features extracted from the previous stage are of different formats and length. Therefore, we need to preprocess the input features before they are passed to later stages. Extracted features can be in the form of text in case of text mining features, or in binary format such as the SLN feature, or in numeric value such as the depth of website sitemap or page content similarity. Although RNN can handle multifaceted features without any preprocessing or very little feature engineering, preprocessing will decrease the training time and the complexity of the network. For text-based features, some additional preprocessing steps are needed.

1. Cleaning: by removing spaces, special marks, and unfamiliar words.
2. Vector representations of words: text words are converted to a vector representation. There are many tools to get a vector of words such as word2vec[1] and GloVe[2]. Intuitively, they are telling the network which words are similar so it needs to learn less about the language. Using pre-trained vectors will help the network to generalize to unseen words [31].
3. When data is unscaled, has a range of values, (e.g., quantities in 10s to 100s) it is possible for large inputs to slow down the convergence of the framework. However, standardizing the inputs will accelerate the training time and reduce the chances of getting stuck in local optima. Gaussian distribution is used to normalize input to zero mean and unit variance from 0 to 1.
4. Finally, labeling the webpages with meta-data tags would be useful for training. The meta-data labels are not fed into the neural network model as an input feature but they are used to stratify or balance the data set for training and testing purposes. The feature extraction and preprocessing phases are shown in Figure 1.

3.4 Classification using BLSTM RNNs

3.4.1 Recurrent networks

Recurrent networks (RNN) is a class of artificial neural networks that utilizes sequential information and maintains history of data through its intermediate layers. They are distinguished from other neural networks by having a feedback loop connected to their past decisions, and memory cells. The decision a recurrent net reached at time step $t-1$ affects the decision it will reach one moment later at time step t [32]. For input xt and a previous output $ht-1$, the mathematical representation of RNN is:

$$ht = \sigma\,(W \cdot xt + R \cdot ht-1 + b) \quad (1)$$

Where W, R, b are input weight, hidden weight, and bias respectively. σ is a non-linear function which is sigmoid by default. These weights need to be adjusted to minimize the total loss on training data. There are commonly used learning algorithms such as (Nesterov) Momentum Method, AdaGrad, AdaDelta and ADAM. We will use ADAM because it can converge a lot faster than other methods and it gives a quick idea of the capability of a given network topology. However, recurrent networks cannot keep memory for a long time, and the nonlinear gradient of sigmoid vanish or explode by running the network so many times. To overcome these problems, more advanced RNN should be used.

3.4.2 Bidirectional LSTM RNN

Bidirectional RNN can be trained simultaneously in the positive and the negative time direction h_{t-1} and h_{t+1} [33]. Adding memory cells to recurrent networks enable them to

memorize data for long time and so solve the vanishing gradient problem. Long Short-Term Memory or LSTM is specifically designed to model long term dependencies in RNNs [34] by adding a memory cell (C_t) and three gates called as input (I_t), output (O_t) and forget (ft) gates. These gates make decisions about what to store, and when to read, write and erasure. So, the network can be retrained easily as it dynamically learns from new data inputs. Studies have showed that Bidirectional LSTM is capable of fast and effective relearning [35].

One essential factor to successful RNN is selecting the activation functions. They are needed for the hidden units to introduce nonlinearity into the network [36]. Since phishing website detection is a multi-class classification problem and the outputs of the network are needed to be interpretable as posterior probabilities, we will use the SoftMax activation [37]. SoftMax gets the probability of each class, so the outputs of the function lie between zero and one, and to sum to one.

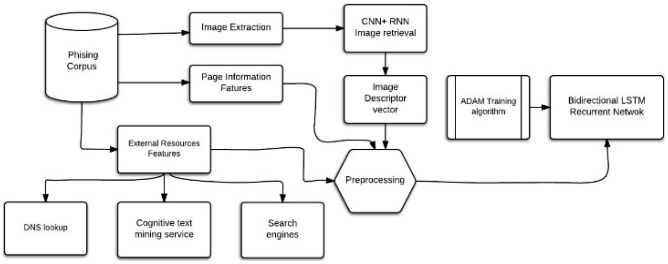

Figure 1: Feature Extraction and Preprocessing Phases.

3.5 BLSTM-RNN System Architecture

The proposed system composed of two phases: the training phase and the recognition or validation phase. In both phases, features are extracted and then preprocessed as explained in the previous steps. In the training phase, preprocessed features are split into training data and testing. The training data represent 75 % of total data, testing data represents 15 %, and the remaining 10 % is for validation.

The BLSTM-RNN is trained by ADAM learning algorithm and SoftMax activation function. The BLSTM-RNN will save the output of this phase into a trained model. Additionally, the system will save the related data used while training in a lookup database data as shown in Figure 2 a. This trained model contains the logic, rules, and weights of the BLSTM-RNN, and will be recalled during the recognition phase. The lookup database has two roles. The first is to elicit the common fraud cues in each industry which help speeding up the recognition process and identifying spoof websites. The second role is to enhance the recognition system performance by updating false detection records and retrain the recognition logic.

In the validation or recognition phase, the system extract and preprocess features from the suspected site. Then, it uses the trained model to detect the authenticity of the site. In addition, websites that has the same business focus are retrieved from the lookup database to recognize if the domain name exists or not and give higher weight to the business focus common cues. Moreover, a retraining loop is returned to the BLSTM-RNN to add the detected website to the trained model and accumulate phishing detection system logic.

The output of the BLSTM-RNN is recognizing whether the website is phishing or real with a confidence level (e.g. the x website is 0.6 real). Then, the website records are stored in the lookup database, so other systems can use it as a reference lookup. The architecture of the recognition phase is shown in Figure 2b.

4 Conclusion

Phishing websites are a serious threat to the economy resulting in billions of dollars loss for internet users. Challenges for phishing websites emerge not only from their increasing number, but also from the intelligent strategies used by designers to give them the legitimate appearance and make them hard to detect. Current phishing detection systems are limited in their ability to adapt to the continuous changes in phishing strategies. Additionally, they lack generalization across different business focuses. This study proposed a cognitive framework that use domain knowledge features combined with semantic text and image features to detect phishing websites. The framework uses the powerful deep learning network of Bidirectional LSTM RNN for phishing detection in combine with Convolution Networks (CNN) for semantic images feature extraction. The system can relearn from newly detected websites and keep records of them in a lookup database. In addition, the domain knowledge stored in the Lookup database will help design engineers to detect new phishing techniques as they emerge.

Future work will include phishing detection system implementation and empirical evaluation. Case studies are developed to ensure that the framework will be a useful phishing website detection application for different business focuses. Another related future research is to expand the system to detect spam and online fraud advertisements. It will include building a large-scale business system and provide it as a web service through browser extensions or API calls.

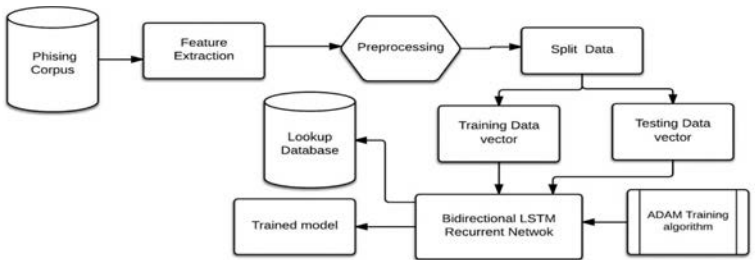

Figure 2 a: Training phase of the BLSTM-RNN system

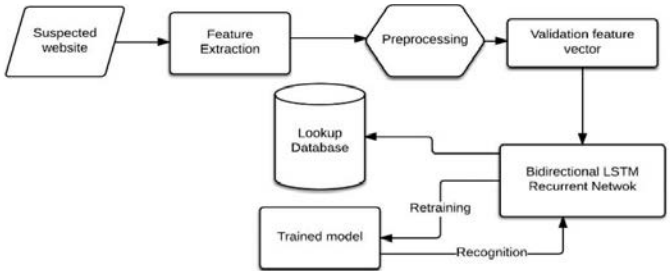

Figure 2 b: Recognition phase of the BLSTM-RNN system

5 References

[1] R.B. Basnet, S. Mukkamala, A.H. Sung, Detection of Phishing Attacks: A Machine Learning Approach, Soft Computing Applications in Industry, 226 (2008) 373-383.
[2] N. Williams, S. Li, Simulating Human Detection of Phishing Websites: An Investigation into the Applicability of the ACT-R Cognitive Behaviour Architecture Model, Cybernetics (CYBCONF), 2017 3rd IEEE International Conference on, IEEE, 2017, pp. 1-8.
[3] N. Abdelhamid, A. Ayesh, F. Thabtah, Phishing detection based associative classification data mining, Expert Systems with Applications, 41 (2014) 5948-5959.
[4] D. Zhang, Z. Yan, H. Jiang, T. Kim, A domain-feature enhanced classification model for the detection of Chinese phishing e-Business websites, Information & Management, 51 (2014) 845-853.
[5] C.E.H. Chua, J. Wareham, Fighting internet auction fraud: An assessment and proposal, Computer, 37 (2004) 31-37.
[6] Y. Zhang, S. Egelman, L. Cranor, J. Hong, Phinding phish: Evaluating anti-phishing tools, ISOC, 2006.
[7] A. Abbasi, H. Chen, A comparison of tools for detecting fake websites, Computer, 42 (2009).
[8] P. Kolari, T. Finin, A. Joshi, SVMs for the Blogosphere: Blog Identification and Splog Detection, AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, 2006, pp. 92-99.
[9] Z. Yan, S. Liu, T. Wang, B. Sun, H. Jiang, H. Yang, A Genetic Algorithm Based Model for Chinese Phishing E-commerce Websites Detection, International Conference on HCI in Business, Government and Organizations, Springer, 2016, pp. 270-279.
[10] A. Alnajim, M. Munro, An Approach to the Implementation of the Anti-Phishing Tool for Phishing Websites Detection, Intelligent Networking and Collaborative Systems, 2009. INCOS'09. International Conference on, IEEE, 2009, pp. 105-112.
[11] T. Dinev, Why spoofing is serious internet fraud, Communications of the ACM, 49 (2006) 76-82.
[12] A. Abbasi, H. Chen, Detecting fake escrow websites using rich fraud cues and kernel based methods, arXiv preprint arXiv:1309.7261, (2013).
[13] A. Abbasi, Z. Zhang, D. Zimbra, H. Chen, J.F. Nunamaker Jr, Detecting fake websites: the contribution of statistical learning theory, Mis Quarterly, (2010) 435-461.
[14] G.A. Miller, The cognitive revolution: a historical perspective, Trends in cognitive sciences, 7 (2003) 141-144.
[15] G. Wang, Data-Driven Granular Cognitive Computing, International Joint Conference on Rough Sets, Springer, 2017, pp. 13-24.
[16] W. Hildesheim, Cognitive Computing–the new Paradigm of the Digital World, Digital Marketplaces Unleashed, Springer2018, pp. 265-274.
[17] F. Weninger, J. Bergmann, B. Schuller, Introducing currennt: The munich open-source cuda recurrent neural network toolkit, The Journal of Machine Learning Research, 16 (2015) 547-551.
[18] L. He, D. Jiang, L. Yang, E. Pei, P. Wu, H. Sahli, Multimodal affective dimension prediction using deep bidirectional long short-term memory recurrent neural networks, Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge, ACM, 2015, pp. 73-80.
[19] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556, (2014).
[20] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 1717-1724.
[21] M. Khonji, Y. Iraqi, A. Jones, Phishing detection: a literature survey, IEEE Communications Surveys & Tutorials, 15 (2013) 2091-2121.
[22] J. Mao, W. Tian, P. Li, T. Wei, Z. Liang, Phishing-Alarm: Robust and Efficient Phishing Detection via Page Component Similarity, IEEE Access, 5 (2017) 17020-17030.
[23] R. Srinivasa Rao, A.R. Pais, Detecting Phishing Websites using Automation of Human Behavior, Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security, ACM, 2017, pp. 33-42.
[24] N. Sanglerdsinlapachai, A. Rungsawang, Using domain top-page similarity feature in machine learning-based web phishing detection, Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on, IEEE, 2010, pp. 187-190.
[25] Y. Zhang, J.I. Hong, L.F. Cranor, Cantina: a content-based approach to detecting phishing web sites, Proceedings of the 16th international conference on World Wide Web, ACM, 2007, pp. 639-648.
[26] G. Ramesh, I. Krishnamurthi, K.S.S. Kumar, An efficacious method for detecting phishing webpages through target domain identification, Decision Support Systems, 61 (2014) 12-22.
[27] L. Wenyin, N. Fang, X. Quan, B. Qiu, G. Liu, Discovering phishing target based on semantic link network, Future Generation Computer Systems, 26 (2010) 381-388.
[28] A. Vedaldi, K. Lenc, Matconvnet: Convolutional neural networks for matlab, Proceedings of the 23rd ACM international conference on Multimedia, ACM, 2015, pp. 689-692.
[29] A. Sharif Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN features off-the-shelf: an astounding baseline for recognition, Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2014, pp. 806-813.
[30] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, Proceedings of the 22nd ACM international conference on Multimedia, ACM, 2014, pp. 675-678.
[31] D. Britz, Recurrent neural network tutorial, part 4-implementing a GRU/LSTM RNN with python and theano, URL http://www. wildml. com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano, (2015).
[32] P. Liu, S.R. Joty, H.M. Meng, Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings, EMNLP, 2015, pp. 1433-1443.
[33] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing, 45 (1997) 2673-2681.
[34] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation, 9 (1997) 1735-1780.
[35] N. Beringer, A. Graves, F. Schiel, J. Schmidhuber, Classifying unprompted speech by retraining LSTM nets, Artificial Neural Networks: Biological Inspirations–ICANN 2005, (2005) 575-581.
[36] P. McCullagh, Generalized linear models, European Journal of Operational Research, 16 (1984) 285-292.
[37] M.I. Jordan, Why the logistic function? A tutorial discussion on probabilities and neural networks, Citeseer, 1995.