II INTERNATIONAL SCIENTIFIC CONVENTION
"II ICCUCLV 2019"

JUNE 23 th – 30 th, 2019
CAYOS DE VILLA CLARA. CUBA.

II INTERNATIONAL CONFERENCE ON INFORMATION PROCESSING
The Application of Online Classifier Ensembles in E-mail Spam Detection

Alberto Verdecia-Cabrera[1, 2], Isvani Frías-Blanco[3], Leticia Arco[4],

Yanet Rodríguez-Sarabia[2], Asiel Díaz-Benítez[2] and Agustín Ortiz-Díaz[5]

1- Universidad de Granma, Cuba. averdeciac@gmail.com

2- Universidad Central "Marta Abreu" de Las Villas, Cuba.

3- LexisNexis Risk Solutions. Sao Paulo, Brazil.

4- Computer Science Department, Vrije Universiteit Brussel, Belgium.

5- CCT-UDESC Santa Catarina State University, Joinville, Santa Catarina, Brazil.

Abstract: Internet of Things, the connection of objects such as computing machines, embedded devices, equipment, appliances, and sensors to the Internet, are generating a huge quantity of data in real time. Many of these devices connected to the internet can be used for the generation of spam emails, which can affect both companies and individual users. Because of that, e-mail servers need to be updated to detect in an efficient way these messages. To analyse these emails, there have been proposed various approaches using traditional data mining learning algorithms. But these algorithms require having the data previously stored in order to classify them. Due to the temporal dimension of the data and the dynamism of these spam emails, the target function to be learned can change over time, a problem commonly known as concept drift. In this paper we propose an approach to filter spam mails based on online ensemble classifiers. The predictive performance of these algorithms is evaluated on the benchmark corpus constructed in this work. The experimental results show that that online ensemble algorithms can be an efficient alternative for the e-mail spam detection.

Keywords: Internet of Things, online learning, concept drift, spam detection.

1. Introduction

The massive data generated by the Internet of Things (IoT) need algorithms that use an extremely small amount of time and memory resources and that are able to adapt to changes and not to stop learning (Chen et al., 2015; De Francisci Morales, Bifet, Khan, Gama, & Fan, 2016). Many application domains such as logistic, industrial process, public safety, home automation, environmental monitoring and healthcare may have significant benefits with IoT systems (Zarpelão, Miani, Kawakani, & de Alvarenga, 2017). Another example of such applications is the spam email detection (Li, Qin, Ren, & Liu, 2017; Ren & Ji, 2017). E-mail is one of the main forms of communication that exist today and used to advertise products and services of dubious quality. Spam is one of the current problems Internet is facing. These unsolicited messages affect both users and e-mail servers. The cost to corporations in bandwidth, delayed e-mail and employee productivity has become a tremendous problem for anyone who provides e-mail services (Islam, Zhou, Guo, & Xiang, 2009), since employees consume working time to distinguish between spam and non-spam messages. Spam mail is also used to spread viruses over the Internet. Due to these problems it is essential to develop efficient spam filters.

Spam mail filters are based on the analysis of the mail headers and content. Nowadays spam filtering is usually tackled by machine learning algorithms, aimed at discriminating between legitimate and spam messages. Machine learning algorithms for text mining are capable to extract knowledge from a set of e-mails and use this knowledge to train classification algorithms. There are several approaches to detect spam emails in the literature such as Naive Bayes classifier (Aski & Sourati, 2016; Metsis, Androutsopoulos, & Paliouras, 2006), support vector machines (SVM) (Sanghani & Kotecha, 2016) and neural networks (Ali, Ozawa, Nakazato, Ban, & Shimamura, 2015). Due to the temporal dimension of the e-mail data (they are constantly arriving) the target function to be learned can change over time. This situation, known as concept drift, complicates the task of estimating this target function because a previous learning model can become outdated

or even contradictory regarding the most recent data. This makes the previous approaches inadequate to deal with concept drift since they have no way to incorporate the information of the new messages.

In this paper we propose an approach to filter spam mails based on online ensemble classifiers. Classifier ensembles have been successfully used for mining non-stationary data streams (Bifet, Holmes, & Pfahringer, 2010; Bifet, Holmes, Pfahringer, Kirkby, & Gavalda, 2009; Frías-Blanco, Verdecia-Cabrera, Ortiz-Díaz, & Carvalho, 2016). Ensemble methods combine the predictions from base classifiers aiming at improving the predictive accuracy obtained by a single classifier. In order to deal with concept drift, these methods use performance measures to monitor the ensemble consistency regarding new data. Significant variations in the performance values are interpreted as a concept drift and these ensemble methods eliminate, reactivate or add new base classifiers dynamically in response to these variations. To use classifier ensembles for filtering spam mails is necessary to preprocess the message corpus to remove stop-words and HTML tags. We use a bag-of-word representation where documents (e-mails) are represented by the words occurring in it without attention to their ordering.

2. Methodology

2.1 Online learning ensemble methods for classification e-mail messages

Online classifier ensembles have been successfully used for mining data streams (Bifet, Holmes, & Pfahringer, 2010; Bifet et al., 2009; Frías-Blanco et al., 2016). Ensemble methods combine the predictions from base classifiers aiming at improving the predictive accuracy obtained by a single classifier. In this section we review three ensemble methods for spam filter.

The algorithm Fast Adaptive Stacking of Ensemble (FASE) (Frías-Blanco et al., 2016) is designed to deal with concept drift. To train the base classifiers, FASE uses previous online bagging method (Oza & Russell, 2001) to work in non-stationary environments. FASE uses adaptive learners both in the voting procedure and in the base classifiers (see

Figure 1). Each adaptive learner uses the Hoeffding-based Drift Detection Method (HDDM) (Frias-Blanco et al., 2015) for drift detection and error estimates, which monitors error rates in order to trigger three different drift signals during the learning process.

The meta-learner of FASE receives *meta-instances* as input, where each attribute is nominal. FASE uses a test-then-train approach (Bifet, Holmes, Kirkby, & Pfahringer, 2010) to generate meta-instances (see Figure 2). Thus, for each original training instance $I = (\vec{a}, c)$, FASE generates a training meta-instance $M = (\hat{c}_1, \hat{c}_2 \ldots, \hat{c}_n, c)$, where each $\hat{c}_i$ is an attribute value and c is its corresponding class label. Each attribute value $\hat{c}_i$ of the meta-instance M corresponds to the predictions of the base classifier i for the instance I. For this meta-instance M, the value $\hat{c}_i$ is the class label predicted by the classifier i. The class label of the meta-instance M is the same as the original training instance.The experimental results show that FASE can be an efficient alternative for learning from data streams.

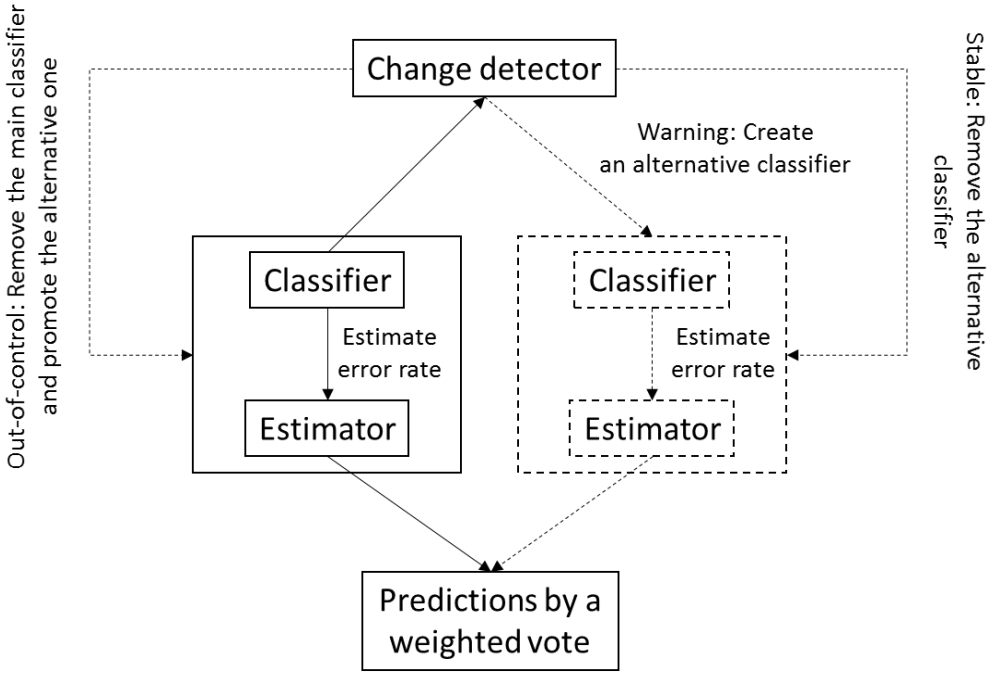

Figure 1. Learning mechanism used in the adaptive learners.

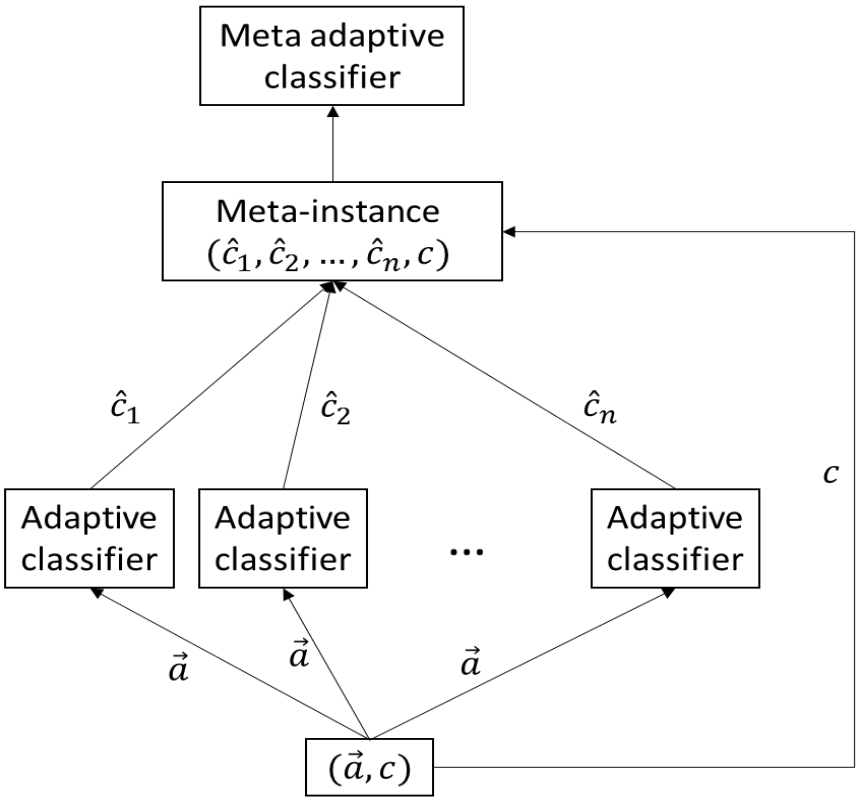

Figure 2. Scheme of FASE.

Online bagging and boosting algorithms have also been extended to be able to deal with non-stationary data streams; e.g., LeveragingBag (Bifet, Holmes, & Pfahringer, 2010) and OzaBagADWIN (Bifet et al., 2009). Basically, when a concept drift is detected they use the ADWIN change detector to replace the base classifiers with the lowest predictive performance in the ensemble by a new base classifier. LeveragingBag, different from OzaBagADWIN, leverage the performance of bagging with two randomization improvements: increasing resampling and using output detection codes.

2.2 Corpus collection

In this paper we start from the SpamAssassin public mail corpus[1] and Enron-Spam datasets[2], as described in (Metsis et al., 2006), to construct our benchmark corpus. The Enron data set is divided into six sub-collections, one for each employee. In (Metsis et al., 2006) spam emails are obtained from different sources and they are grouped into three sources. The first collection (S1) contains messages sent between May 2001 and July 2005, the second one (S2) between August 2004 and July 2005 and the third one (S3) contains spam messages received between December 2003 and September 2005. The six ham message collections (six Enron employees) were each paired with one of the three spam collections (S1, S2, S3). As in (Metsis et al., 2006), in three of the resulting benchmark datasets, we used a ham-spam ratio of approximately 3:1 while in the other three we inverted the ratio to 1:3. To take into account the concept drift problem, all spam messages were ordered chronologically. Table 1 summarizes the characteristics of the six datasets represented by DS1,..., DS6.

Dataset	DS1	DS2	DS3	DS4	DS5	DS6
ham	1966	3669	4363	4012	2364	2714
spam	3675	1499	1496	1500	4500	4496

Table 1. Main characteristics of the constructed data streams.

Text mining techniques were applied for pre-processing our benchmark corpus. Specifically, we use the Vector Space Model (VSM) (Salton, Wong, & Yang, 1975) representation to represent documents because it is widely recognized as an effective representation for documents in the text mining community. First, the documents are transformed into a sequence of word tokens, from which a sequence of indexes of possible terms is created for each document during the term extraction step. The extracted terms

[1]http://spamassassin.apache.org/old/publiccorpus/

[2] http://csmining.org/index.php/enron-spam-datasets.html

are used as the corpus vocabulary V. One problem here is that the number of extracted terms is extremely large; therefore, a vocabulary reduction is necessary. For each document d_j under consideration, the term frequency vector $d_{tf} = \left(tf_d(t_1), \ldots, tf_d(t_m)\right)^T$ is created, where $tf_d(t)$ denotes the number of times that the term $t \in V$ appears in the document d. After creating this vector for each document in the collection, we reduce the number of words by means of a feature selection algorithm. In our case we use information gain since it has been successful for feature selection in text mining (Liu, 2004).

3. Results and discussion

The experimental study described in this section evaluates the algorithms in our benchmark corpus. All the experiments were performed over MOA (Bifet, Holmes, Kirkby, et al., 2010), a framework for online analysis.

The experiments followed the test-then-train approach, which is derived from the predictive sequential (prequential) error. In this approach, when a new instance arrives, first, a classification model predicts its class label (test step). Next, the instance is used by a learning algorithm to update a classification model (train step).

At each new instance, the classifier was first tested and then updated. During the learning process, predictive performance was assessed using a sliding window of size 100. Taking into account that in our benchmark corpus exists certain grade of class imbalance, and the accuracy metric does not provide adequate information on a classifier's performance over imbalanced data, we also include precision and recall metrics.

In the selected algorithms we used the default configuration proposed by the authors. The number of base classifiers was set to 10 for all the ensemble algorithms, which is the default configuration adopted by MOA. FASE used Naive Bayes as base classifier and meta-classifier in all the experiments. We also included the Naive Bayes algorithm in the experimental study because it is one of the most successful algorithms for learning from data streams.

The predictive performances of the algorithms under consideration were evaluated using our benchmark corpus. The concept drift problem can be adverse in real world situations such as spam data. For our constructed datasets, there is no strong claim about the presence or type of change. The algorithms were evaluated processing the instances in their temporal order and the test-then-train approach was adopted to calculate the predictive accuracy. The Tables 2 and 3 reports the accuracy of the algorithms for the six created datasets. The highest levels of predictive accuracy are in bold. These tables show that FASE outperformed the rest of the algorithms regarding predictive accuracy. In addition, all the algorithms obtain good results with 500 and 600 features. Additionally, the Tables 4 and 5 shows precision and recall of the algorithms.

Algorithm	FASE	LeveragingBag	NaiveBayes	OzaBagAdwin
DS1	**99,91 ± 0,54**	99,40 ± 4,07	94,88 ± 11,54	97,00 ± 11,17
DS2	**99,02 ± 3,64**	98,69 ± 5,21	91,50 ± 11,14	98,04 ± 9,85
DS3	**99,92 ± 0,53**	99,39 ± 4,14	92,29 ± 10,73	98,34 ± 8,83
DS4	**99,91 ± 0,54**	99,43 ± 3,98	91,61 ± 10,58	97,73 ± 11,87
DS5	**99,93 ± 0,49**	99,54 ± 3,59	94,12 ± 11,09	96,65 ± 12,55
DS6	**99,44 ± 3,09**	99,17 ± 3,48	95,54 ± 11,09	97,10 ± 12,08

Table 2. Performance of the algorithms with 500 features.

Algorithm	FASE	LeveragingBag	NaiveBayes	OzaBagAdwin
DS1	**99,91 ± 0,54**	99,39 ± 4,20	94,09 ± 12,61	96,58 ± 11,81
DS2	**98,98 ± 2,98**	98,44 ± 5,90	90,29 ± 12,32	97,77 ± 10,69
DS3	**99,92 ± 0,53**	99,39 ± 4,14	91,08 ± 11,85	98,29 ± 9,06
DS4	**99,91 ± 0,54**	99,43 ± 3,98	89,79 ± 11,96	97,64 ± 12,31
DS5	**99,93 ± 0,49**	99,55 ± 3,47	93,43 ± 12,23	96,19 ± 12,94
DS6	**99,46 ± 2,89**	99,15 ± 3,95	95,10 ± 11,89	96,60 ± 13,05

Table 3. Performance of the algorithms with 600 features.

Algorithm	Measure	DS1	DS2	DS3	DS4	DS5	DS6
FASE	precision	0.99	0.97	0.99	0.99	0.99	0.99
	recall	0.99	0.99	0.99	0.99	0.99	0.99
LeveragingBag	precision	0.98	0.99	0.99	0.99	0.99	0.98
	recall	0.99	0.96	0.98	0.98	0.99	0.99
NaiveBayes	precision	0.99	0.77	0.78	0.77	0.96	0.99
	recall	0.86	0.99	0.98	0.99	0.86	0.88
OzaBagAdwin	precision	0.92	0.99	0.99	0.99	0.91	0.93
	recall	0.99	0.93	0.94	0.92	0.99	0.99

Table 4. Precision and recall of the algorithms with 500 features.

Algorithm	Measure	DS1	DS2	DS3	DS4	DS5	DS6
FASE	precision	0.99	0.97	0.99	0.99	0.99	0.99
	recall	0.99	0.99	0.99	0.99	0.99	0.99
LeveragingBag	precision	0.98	0.99	0.99	0.99	0.99	0.98
	recall	0.99	0.95	0.98	0.98	0.99	0.99
NaiveBayes	precision	0.99	0.75	0.75	0.73	0.97	0.99
	recall	0.84	0.99	0.98	0.99	0.84	0.87
OzaBagAdwin	precision	0.91	0.99	0.99	0.99	0.90	0.92
	recall	0.99	0.92	0.93	0.91	0.99	0.99

Table 5. Precision and recall of the algorithms with 600 features.

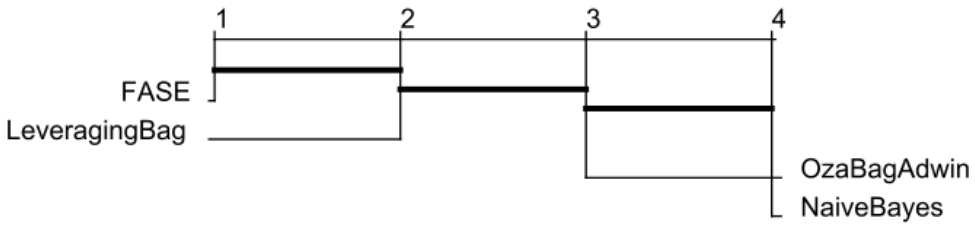

Figure 3. Comparison of all classifiers against each other with the Friedman test and the Holm procedure for the post hoc analysis. The ranks were computed in accordance with Tables 2 and 3.

4. Conclusions

In this work, we experimentally evaluated three online ensemble algorithms that are able to deal with concept drift. These algorithms have not been previously used in the context of e-mail spam detection. Specifically, we included in our study the algorithms FASE, LeveranginBag and OzaBagAdwin. We also included the Naive Bayes, because it is one of the most successful algorithms for learning from data streams and text mining. The paper also presented empirical results of the online ensemble algorithm in the constructed benchmark corpus. The experimental results show that FASE and LeveragingBag obtained the best predictive performance. Additionally, these results show that text mining techniques in combination with online ensemble algorithms can be an efficient alternative for e-mail spam detection. We plan to continue with this research by using other learning algorithms and methods for feature selection.

5. References

Ali, S., Ozawa, S., Nakazato, J., Ban, T., & Shimamura, J. (2015). An autonomous online malicious spam email detection system using extended RBF network. In *2015 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–7). https://doi.org/10.1109/IJCNN.2015.7280826

Aski, A. S., & Sourati, N. K. (2016). Proposed efficient algorithm to filter spam using machine learning techniques. *Pacific Science Review A: Natural Science and Engineering*, *18*(2), 145–149. https://doi.org/10.1016/j.psra.2016.09.017

Bifet, A., Holmes, G., Kirkby, R., & Pfahringer, B. (2010). Moa: Massive online analysis. *The Journal of Machine Learning Research*, *11*, 1601–1604.

Bifet, A., Holmes, G., & Pfahringer, B. (2010). Leveraging bagging for evolving data streams. In *Machine learning and knowledge discovery in databases* (pp. 135–150). Springer. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-15880-3_15

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., & Gavalda, R. (2009). New ensemble methods for evolving data streams. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 139–148). ACM. https://doi.org/10.1145/1557019.1557041

Chen, F., Deng, P., Wan, J., Zhang, D., Vasilakos, A. V., & Rong, X. (2015). Data mining for the internet of things: literature review and challenges. *International Journal of Distributed Sensor Networks*, *11*(8), 431047.

De Francisci Morales, G., Bifet, A., Khan, L., Gama, J., & Fan, W. (2016). Iot big data stream mining. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 2119–2120). ACM.

Frias-Blanco, I., Campo-Avila, J. del, Ramos-Jimenez, G., Morales-Bueno, R., Ortiz-Diaz, A., & Caballero-Mota, Y. (2015). Online and Non-Parametric Drift Detection Methods Based on Hoeffding Bounds. *IEEE Transactions on Knowledge and Data Engineering*, *27*(3), 810–823. https://doi.org/10.1109/TKDE.2014.2345382

Frías-Blanco, I., Verdecia-Cabrera, A., Ortiz-Díaz, A., & Carvalho, A. (2016). Fast adaptive stacking of ensembles. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing* (pp. 929–934). ACM. Retrieved from http://dl.acm.org/citation.cfm?id=2851655

Islam, M. R., Zhou, W., Guo, M., & Xiang, Y. (2009). An innovative analyser for multi-classifier e-mail classification based on grey list analysis. *Journal of Network and Computer Applications*, *32*(2), 357–366. https://doi.org/10.1016/j.jnca.2008.02.023

Li, L., Qin, B., Ren, W., & Liu, T. (2017). Document representation and feature combination for deceptive spam review detection. *Neurocomputing*, *254*, 33–41. https://doi.org/10.1016/j.neucom.2016.10.080

Liu, Y. (2004). A Comparative Study on Feature Selection Methods for Drug Discovery. *Journal of Chemical Information and Computer Sciences*, *44*(5), 1823–1828. https://doi.org/10.1021/ci049875d

Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006). Spam filtering with naive bayes-which naive bayes? In *CEAS* (Vol. 17, pp. 28–69).

Oza, N. C., & Russell, S. (2001). Online Bagging and Boosting. In T. Jaakkola & T. Richardson (Eds.), *Eighth International Workshop on Artificial Intelligence and Statistics* (pp. 105–112). Key West, Florida. USA: Morgan Kaufmann. https://doi.org/10.1109/ICSMC.2005.1571498

Ren, Y., & Ji, D. (2017). Neural networks for deceptive opinion spam detection: An empirical study. *Information Sciences*, *385–386*, 213–224. https://doi.org/10.1016/j.ins.2017.01.015

Salton, G., Wong, A., & Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. *Commun. ACM*, *18*(11), 613–620. https://doi.org/10.1145/361219.361220

Sanghani, G., & Kotecha, K. (2016). Personalized spam filtering using incremental training of support vector machine. In *Computing, Analytics and Security Trends (CAST), International Conference on* (pp. 323–328). IEEE. Retrieved from http://ieeexplore.ieee.org/abstract/document/7914988/

Zarpelão, B. B., Miani, R. S., Kawakani, C. T., & de Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things. *Journal of Network and Computer Applications*, *84*, 25–37. https://doi.org/10.1016/j.jnca.2017.02.009