An Effective Approach for Clickbait Detection Based on Supervised Machine Learning Technique

https://doi.org/10.3991/ijoe.v15i03.9843

Daoud M. Daoud(✉)
Higher Colleges for Technology, Sharjah, UAE
ddaoud@hct.ac.ae

M. Samir Abou El-Seoud
The British University in Egypt, Cairo, Egypt

Abstract—Clickbait is a term used to describe a deceiving web content that uses ambiguity to prompt the user into clicking a link. It aims to increase the number of online readers in order to generate more advertising revenue. In other words, Clickbait is used to describe a type of hyperlink on a web page which seduces a user to click a link to continue reading a specific article.

Typically such links will forward the visitor to a page that requires payment, registration, or lead a user to a site, which tries to sell user something or possibly extort user, by withholding the promised "bait". We use supervised machine learning to create a model that is trained on 24 features. This method achieved an F1-score of 79% and an area under the ROC curve of 0.7. Our methodology emphasises the importance of using features extracted from different elements of social media posts along with features that are extracted from the title and the article. In this research, we show that it is possible to identify Clickbaits using all parts of the post while keeping the number of features as minimum as possible.

Keywords—Clickbait, F1-score, ROC curve, Similarity, Feature extraction, Formality, Readability, Support Vector Machine (SVM), Data Modeling.

1 Introduction

A Clickbait is a deceiving headline with the aim of increasing advertisement revenue without offering adequate content or content that is close to the advertised title. A post is considered a Clickbait if it withholds information needed to understand the main theme of the article. Saying, “you will not believe what this team did!” instead of “Real Madrid wins its 12 EUFA champions league” is an example of a clickbait.

A Clickbait causes user disappointment. When the user clicks on a link and is redirected to an article, the reader finds out that the expectations were fake and he was manipulated. Clickbaits succeeded in fooling many readers. On the other hand, people are increasingly turning to social media for news. Unfortunately, they are getting abused by propaganda websites. These websites use Clickbaits with the intent not only to increase revenues but also to spread their propaganda too. Clickbait article is intentionally over-promises. Article headlines provide enough information to make readers curious, but not enough to satisfy their curiosity without clicking through to the linked content. Upon inspection of such articles, users find out that they do not necessarily match their corresponding headlines. Clickbaits articles are over-promising and under-delivering.

Identifying Clickbaits is an essential part of blocking them from the user's social media feed. Facebook, Twitter and other social media websites faced much criticism for not identifying Clickbaits and down ranking them from the user's feed. Figure 1 is an example of a Clickbait post on Facebook.

Man Tries to Hug a Wild Lion, You Won't Believe What Happens Next!
If the story was any true, either the lion would have killed the man or he would have done nothing. Well, those are the only two possibilities! But, the clickbait doesn't work like that! It makes you crave for the story.

Fig. 1. An example of a Clickbait

This research aims at solving the problem by training a model on a labelled (Clickbaits, non-Clickbaits) dataset consisting of some social media posts.

Most of the researchers are using supervised machine learning to detect Clickbaits. Though, research in this field is still active in trying to enhance the accuracy of these systems. In this paper, we will use supervised machine learning. To build our model, we will only use a minimum number of features to achieve the task. So, we select the elements from a post that are strongly related and can influence our model positively. Reducing the number of features has a positive impact on the performance of the final model.

2 Literature Review

The problem of classifying false and misleading content have existed for so long. Various attempts to solve or at least organise such material have been made in multiple applications using different approaches from spam detection to fake news detection. The methods used varied in complexity. Some were too complex to be feasibly used. Below we will discuss previous approaches used to tackle the Clickbait problem.

Yahoo research team has developed a Clickbait detection algorithm (Biyani, Tsioutsiouliklis et al. 2016). They analysed the article and the title to extract features used to detect Clickbaits. They used several text formality measures to help identify Clickbaits. Their objective was to prove that there is a relation between article informality and Clickbaits. Text formality is an index of how formal a given text is. They extracted 7000 features from 4000 articles. This approach achieved an F-1 Score of 74.9%. In our methodology, we used one of the formality measures as a feature. This method will be presented in detail in the methodology section.

Also, researchers from Bauhaus-Universität Weima used 2992 tweets on a model of 215 features and produced an F-1 score of 76% (Potthast, Köpsel et al. 2016). This research was the first paper to be published tackling the Clickbait problem. The approach relied heavily on the bag of words algorithm. Sentiment analysis and readability measures were also used.

Clickbait detection is a part of detecting fake news, but due to the complexity of identifying fake news, researchers try to find a solution to the problems in that domain hoping to get closer to solving the problem on a more significant scope. An interesting linguistic approach used to tackle the fake news problem relied on the cues present in the text that shows that the writer is lying (Bourgonje, Schneider et al. 2017). Such signs were based on the frequencies of pronouns in the post and the percentage of negative words used in the article.

A common aspect of the previously mentioned approaches is that their study was conducted based on the relevance of title and article and ignoring other elements that commonly exist in social media posts. This approach produced a good result. However, our objective is to enhance the trained model further by considering the rest of the elements in the classification process and also by considering other features. Despite this, we kept the number of features as low as possible to retain an acceptable performance without affecting the integrity of the classification. We noticed that the number of features used in the previously mentioned papers is huge which affects the performance and the ability of the model to be used in a real-time application. Some of the features that are used in our approach have been used by other researchers and proved to be relevant to the problem. We will further discuss this issue in the next section.

3 Methodology

To train our model, we use supervised machine learning. Supervised machine learning is the process of using labelled data to train a modelling algorithm to discriminate between different labelled data. The algorithm is fed by different values of a set of features and their labels. Features are a set of attributes that best describe the differences between the labels. To best explain the problem, we need to identify the elements that best represent the differences between the two classes that we have and then extract features chosen from the dataset. We then train the model on the features that were extracted. In the following sections, we will describe in details the dataset, feature extraction, elimination process and the modelling algorithm used.

3.1 Data Set

The dataset used was provided by Bauhaus-Universitat Weimar as a part of Clickbait detection challenge organised by the university (Weimar 2017). The data was annotated by five judges. The dataset contained 22,033 posts and was divided to 2495 posts for training the model, and 19538 posts used for the validation of the model. The dataset included two JSON files and a media archive in case a post includes images. One of the JSON files stores the ID of the post and its label. Below (see Table 1) is a description of the fields present in the other JSON file. The fields that are used in feature extraction is bolded out.

Table 1. Dataset description

Fields	Description
ID	The ID of the post
Post timestamp	The time it was posted
Post text	The text of the post
Post media	The path to the file in the media archive
Target title	Title of the article linked
Target description	Description tag of the target article
Target keywords	Keywords tag of the target article
Target paragraphs	text of the target article
Target captions	The caption of the images(if exists)

The ID of the post is used to search for the label in the label file. The post text is a text written by the one who shared the article as a comment or description of what is expected when reading the article. It can be considered as a title written by the one who shared the post. The target title is the title of the article and is supposed to reflect what the article is about although that does not happen with Clickbaits. The description and the keywords are data extracted from the Meta tags in the source code of the site. The target paragraph is the article text.

Another data set was used to implement the bag of words algorithm. The algorithm and the data will be discussed in the next section.

3.2 Feature extraction

The dataset used was provided by Bauhaus-Universitat Weimar as a part of Clickbait detection challenge organized by the university (Weimar 2017). The data was annotated by five judges. The dataset contained 22,033 posts and was divided to 2495 post for training the model, and 19538 posts used for the validation of the model. The dataset included two JSON files and a media archive where images were placed in the post included images. One of the JSON files included the ID of the post and its label. Below (see Table 1) is a description of the fields present in the other JSON file. Fields used in feature extraction is bolded out.

Generally, extraction of the features is a process that relies heavily on domain experience and previous researches that proved the functioning of certain features. This process is the most critical part of model training because if the features extracted have low correlation with the labels, no matter what modelling algorithm, the model will fail to classify correctly. Initially, we have extracted 28 features. In the coming sections, we will present the most common features used.

Similarity: The most common metric used was the similarity. We used similarity to measure the similarity between article's text, article's title, and post. The sentence similarity was measured by tokenizing each sentence into a set of words and then use WordNet. It is a dictionary that returns synonyms set as a graph. It was used to find out all the synonyms of each word. Then every word in sentence "A" was compared with every word and its synonyms of sentence "B" using path similarity. Path similarity calculates similarity based on the shortest path between words in the WordNet. The distance is calculated by counting the number of edges in the WordNet graph structure. We use path similarity and not subject based similarity because the subject-based similarity is context based and calculates the similarity between the two subjects, unlike path similarity that calculates the similarity between words themselves.

Context-based similarity returns high similarity if the sentences speak about different topics, but the topics are somehow related. For example, if we compare a sentence that talks about coffin prices with a sentence that talks about cancer, then it will return that the similarity is high since coffin and cancer are two related topics referring sickness and death. Path similarity will yield that the similarity is low as cancer is not a synonym of the coffin. What we needed to know is that: if the two sentences are similar and both speak about the same topic or related topics; hence path similarity will be used.

Formality: Usually, Clickbait articles are less formal than non-Clickbaits as proven by Biyani (Biyani, Tsioutsiouliklis et al. 2016). Clickbaits are low-quality pages that serve interesting content. The language used in Clickbait articles tends to be less formal than that used for professionally written news articles. For this reason, we decided to use a formality metric called *fmeasure* (HEYLIGHEN and DEWAELE 1999) to capture such informality difference. This metric determines how formal text is by measuring the frequency of a different part of speech tags in the text. The measure relies on the structure of the text to determine whether it is formal or not. *Fmeasure* is used to measure the formality of the post and the article. In other words, the amount of different part-of-speech tags in a text is used to calculate the degree of formality F in the text. The scores will be used as features. The formality F is calculated using the following formula (HEYLIGHEN, F. and J.-M. DEWAELE (1999) and Biyani, Tsioutsiouliklis et al., AAAI-16, pp. 95, 2016):

$F = (nounfreq + adjectivefreq + prepositionfreq + articlefreq - pronounfreq - verbfreq - adverbfreq - interjectionfreq + 100)/2.$

In the above formula, the frequencies are expressed as percentages of the number of words belonging to a specific category concerning the total number of words in the excerpt.

Readability: Another observation is that Clickbaits are usually written in simpler English than non-Clickbaits, and they are usually easier to read. A metric used to measure how readable is the text called Automated Readability index (Smith, E. A. and Senter, R. J. 1967) was used. This metric is used to determine the difficulty level of text in standardized examinations. The higher the value of the readability index the harder it is to read the text. It is measured as follows (Senter and Smith 1967):

Automated readability index = 4.71 (number of character/number of words) +0.5*(number of words/number of sentences)-21.43*

The first part of the formula is to calculate the average number of characters in the word. The larger the word, the harder it is to read. The next part of the formula is to calculate the average number of words in each sentence. The higher the average numbers of words in the sentence, the higher the readability index.

Bag of words: Some words are frequently used in Clickbait titles to make it look more appealing to the readers without giving any context to what is discussed in the article. To extract these words we implemented the 1-gram algorithm, which proved to be very useful with subjectivity analysis (Biyani P. et al. AAA-16, 2016) and sentiment analysis (Pang B. et al., Proceeding of the ACL, 2004). The 1-gram algorithm was implemented on a separate data set. The dataset included 6080 non-Clickbait titles and 5637 Clickbait titles (Mathura 2017). After the removal of stop words, we extracted the 100 most frequently used words in Clickbait titles and the 100 most commonly used words in non-Clickbait titles. We have extracted words that are exclusively used in Clickbait titles.

'Trump', 'review', and 'Australian' was top 3 words used in non-Clickbaits with a frequency of 1660, 1404, and 1260 respectively. The top 3 words used in Clickbaits was 'you', 'this', 'people' with a frequency of 3894, 3848, and 2760 respectively. We noticed that the usage of pronouns is much more frequent with Clickbaits. For this reason, we decided to use the number of pronouns in the title and the post as features in the model. This also proved that words are repeated much more frequently in Clickbaits. Although the number of not Clickbait titles is higher in the data set, the top 3 words in Clickbait have twice the occurrence of the top 3 words used in not Clickbait title.

Noun extraction: In Clickbaits, the subject of the title or the post usually doesn't match the theme of the article itself due to the ambiguity factor presents intentionally in Clickbaits. To measure that, we extracted nouns from the title and the post and measured the ratio of the nouns in the title mentioned in the article. The ratio was measured by finding the part of speech tag of each word, extracting nouns from both article and title and then counting the number of nouns in the title, then dividing it by the total number of nouns found.

3.3 Feature engineering

As shown above, 28 features identified in our model. The more features we consider, the more time is needed for processing. Certainly, this might negatively affect the model's performance. Although adding features may increase the accuracy this isn't always true. If a feature has a high correlation with another feature or is derived from another feature, the accuracy might decrease. The more features added, the more data is required to ensure there are enough samples for each combination of values.

Based on all those disadvantages of having a model with high dimensionality, we decided to perform recursive feature elimination to decrease the dimensionality of the model. Recursive feature elimination works by recursively considering smaller and smaller sets of features. We used recursive feature elimination because of the several factors affecting feature selection. Statistically, features with a correlation close to zero should be eliminated, but that is not the only factor to be considered. A feature with high correlation with another feature can decrease the performance of the model. To ensure that we chose the best discriminatory features, we decided to implement the recursive method. After applying the algorithm, four features were removed.

3.4 Modelling the data

Even though feature elimination decreased the dimensionality of the data, Our data is still high dimensional, leading to a low accuracy of the results for some types of algorithms. To avoid this problem, we used a support vector machine (SVM) (Corttes and Vapnick 1995).

SVM is an algorithm that takes each instance in the data set as a vector and plots it in a high dimensional space and then constructs a hyperplane to separate each class from the other. The separator can be a straight plane or a curve depending on the linearity of the data. The linear SVM performed better on the data hence the linearity of the data available. The hyperplane is selected so that the distance between the plane and the nearest data point of each class is maximized. The model is designed to handle high dimensional data and has a high noise tolerance. Due to the high dimensionality of the data, we were unable to plot it, but below is a linear SVM hyperplane separating a three-dimensional data into two classes (see Figure 2).

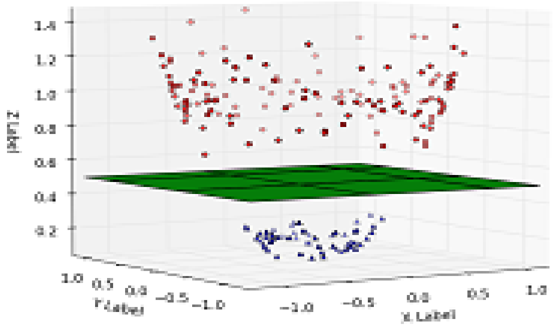

Fig. 2. 3-dimensions plot

4 Results and Discussion

4.1 Results using different modelling algorithms

Our model was trained on 2495 posts that consist of 762 Clickbait posts and 1697, not Clickbait posts. To validate the model we used 19,487 posts that include 14774

Clickbait posts and 4713 non-Clickbait posts. Also, to using SVM with linear kernel, we tried using logistic regression. It is an algorithm that is known to work well for binary classification problems and linearly separable data. The logistic algorithm tries to separate between the two classes using probability, unlike SVM that uses Euclidean distance. SVM attempts to find the broadest possible separating margin, while Logistic Regression uses probabilities modelled by the sigmoid function to discriminate between the two classes. We achieved 79% accuracy in both models. The results of the validation process were presented in table 2.

Table 2. Results

Algorithm	Precision	Recall	F1-Score
Logistic regression	0.79	0.79	0.79
Linear SVM	0.78	0.79	0.79

Precision is calculated by dividing the number of positives predicted correctly over the total true positive results in the data. The recall is the ratio of true positive results predicted correctly over the total number of all relevant samples that should have been identified as positive. F1-score is the weighted average of both precision and recall, and is measured by using the following formula

$$\mathrm{F1} = 2 * \frac{\mathrm{precision} * \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \quad (1)$$

In a statistical analysis of binary classification, the F1-score is a measure of a test`s accuracy. It considers both the recall and precision of the test to evaluate the score. F1-score reaches its best value at (perfect precision and recall) and worst at 0.

The results obtained were very similar using both algorithms. The two algorithms are known to perform similarly. The reason for using SVM is because of its capability on handling noise in the data. The content of the web evolves, and Clickbaits might change its style which makes necessary to increase data and retrain the model at some point. To make our algorithm practical, we decided to use SVM that uses margin instead of a line to separate between the two classes. The margin is the gap between the two types. Unlike logistic regression that needs all the data in the training method, SVM only uses the data closest to the margin. The margin is placed in a point such that the distance between the margin and the points of the two classes are maximized. The size of the margin and the linearity of the data were determined by using a genetic algorithm called Grid Search. This method makes SVM much faster to train than logistic regression. Another reason on why SVM was selected is that it deals better with noise. Data parsed from the web can be messy, and much noise might be presented. SVM is known to perform well on noisy and missing data(Janardhanan, L. et al. 2015).

4.2 Validation and Training accuracy

Training accuracy is the accuracy of the model to classify the same data it was trained on. Validation accuracy is the accuracy of the model to organise different data. The Training accuracy of the model was found to be 74.5%, and the validation accuracy is 79.4%. This measure is useful for better understanding the model behaviour. Usually, the training accuracy is slightly higher or equal to the validation accuracy. The model's validation accuracy is 4.9% higher than the training accuracy because the training data's distribution is different from the distribution of the validation data.

4.3 Results analysis

The distribution of the data affected the results as shown in table 3. The better performance in identifying the non-Clickbait is a result of the difference in the number of instances in both classes.

Table 3. Precision, Recall & F1-score

Class	Precision	Recall	F1-score	Instances number
Not-Clickbait	0.85	0.88	0.87	14774
Clickbait	0.58	0.50	0.54	4713
Weighted average	0.78	0.79	0.79	19487

To better evaluate the model, the Receiver Operating Characteristic (ROC) curve was plotted. In a ROC curve, the true-positive percentage (sensitivity) is plotted in function of the false-positive rate for different cut-off points of a parameter.

The ROC curve is used to illustrate the diagnostic ability of a binary classifier. It's the plot of true positive rate against false positive rate. The area underneath the ROC curve (AUC) is used as a metric to measure the feasibility of the model. AUC represents the ability of the algorithm to classify data correctly. In general, AUC is a measure of the usefulness of a test. Greater area means a more useful test. The AUC measures test accuracy. An area of 1 denotes a perfect test, and an area of 0.5 represents a worthless test. In our model, the AUC was 0.70 (see Figure 3).

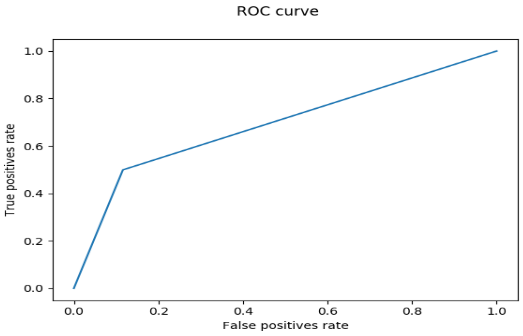

Fig. 3. ROC Curve

These numbers show that our model has a pretty good ability to classifying the posts. It also proves that it is possible to extract relevant features from all parts of the post and keep the dimensionality as low as possible. As mentioned in the literature review, most of the previous researches ware based on the title and article. These approaches achieved good performance in some cases but in a short amount of time, the topics of the articles change.

Consequently, the style of the titles will make the model completely useless. The usage of all parts of the post was an attempt to extract features that generally describe the problem and maintain the functionality of the model for a long period. 6 out of 10 of the top features in our model don't use the title and article which proves that other parts of a post are relevant in identifying Clickbaits posts.

5 Conclusion

In this research, we presented a different approach to the Clickbait classification problem. In the method, we showed the relevance of a social media post elements. Moreover, we showed the importance of new features that ware extracted from different elements of social media as well as features extracted from the title and the article. We demonstrated that the detection of Clickbaits could be done using a minimal number of features. Based on the above we can conclude the following:

- Clickbait detection is possible on social media platforms with better performance if elements of posts on such platforms are correctly used.
- A low number of features can still be sufficient to classify Clickbaits, which helps in building a real-time classifier moving this idea from theory to application.

6 Future Work

Future work may include different learning approach and different methods for features extractions and some modifications that can be more useful in a product than in research. Also experimenting with the Arabic language could be of interest. However, currently, it will not be feasible due to the lack of the appropriate tools for processing text in that language. However, we are planning to extend our work by:

- Limiting the training of the model to the features extracted from all parts of a post excluding the article should make the classifier faster and should also reduce data storage and processing required for fetching and saving the article.
- Determining the features using unsupervised machine learning techniques could result in higher accuracy but can't be done currently due to the need for a larger and more diverse dataset and the longer time needed for the completion of each run in the learning process. Additionally, the unsupervised learning techniques might replace the SVM model. However, this needs to be investigated.
- Porting this approach to Arabic would be a challenging problem. The challenges are stemmed from the limited available resources for this morphologically rich language.

7 References

[1] Biyani, P.; K. Tsioutsiouliklis; et al. (2016). "8 Amazing Secrets for Getting More Clicks": Detecting Clickbaits in News Streams Using Article Informality. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence AAAI-16.

[2] Bourgonje, P.; Schneider, J. M; et al. (2017). "From Clickbait to Fake News Detection: An Approach based on Detecting the Stance of Headlines to Articles". The second workshop on Natural Language Processing meets Journalism, Copenhagen, Denmark (2017). https://doi.org/10.18653/v1/W17-4215

[3] Corttes, C. and V. Vapnick (1995). "Support-Vector Networks". Machine Learning, 20, pp. 273-297, Kluwer Academic Publishers, Boston (1995). https://doi.org/10.1007/BF00994018

[4] HEYLIGHEN, F. and J.-M. DEWAELE (1999). "Formality of Language: definition, measurement and behavioral determinants". Internal Report, Center "Leo Apostle", Free University of Brussels (1999).

[5] Janardhanan, P., H. L., et al. (2015). "Effectiveness of Support Vector Machines in Medical Data mining." Journal of Communications Software and Systems, Vol. 11, No. 1, (2015). https://doi.org/10.24138/jcomss.v11i1.114

[6] Mathura, S. (2017). "Clickbait Detector." 2018. github.com/saurabhmathur96/clickbait-etector/blob/master/data.

[7] Pang Bo and Lee L. (2004). A Sentimental Education: Sentimental Analysis Using Subjectivity Summarization Based on Minimum Cuts. Proceedings of the ACL, pp. 271-278 (2004).

[8] Potthast, M., S. Köpsel, S, et al. (2016). "Clickbait Detection". The 38th European Conference on IR Research (ECIR 16), Padua, Italy, March 20-23, 2016. Published in the series: Advances in Information Retrieval, part of the Lecture Notes in Computer Science book series (LNCS, volume 9626), pp 810-817, Springer (2016). https://doi.org/10.1007/978-3-319-30671-1_72

[9] Smith, E. A.; Senter, R. J. (1967). "Automated Readability Index". Aerospace Medical Research Laboratories and University of Cincinnati, USA, AMRL-TR-66-220 (1967). http://www.dtic.mil/dtic/tr/fulltext/u2/667273.pdf

[10] "Clickbait Challenge 2017". Workshop at Bauhaus-Universität Weimar, Germany, November 2017 http://www.clickbait-challenge.org/

8 Authors

Daoud M. Daoud received his B.Sc. degree in Electrical and Computer Engineering from Kuwait University in 1988, his M.Sc. in Computing Science from Glasgow University, UK, and his PhD in Computing Science from Joseph Fourier University, France. Dr Daoud is currently serving as an Associate Professor at PSUT. He is also a faculty member at Higher colleges for technology, UAE. In the period between 1996 to1999, Dr Daoud worked as a principal investigator for the Arabic section of the Universal Networking Language project. He then worked at the Institute of Advanced Studies, United Nations University (1998-1999). He served as a director for Next Generation, in the Services department at Paltel (1999-2001). His main research interests are Natural Language Processing, machine translation, Information Extraction, Information Retrieval and analysis of Arabic Social Media and Big Data.

M. Samir Abou El-Seoud: Professor Samir Abou El-Seoud received his BSc degree in Physics, Electronics and Mathematics from Cairo University in 1967, his Higher Diplom in Computing from Technical University of Darmstadt (TUD) /Germany in 1975 and his Doctor of Science from the same University (TUD) in 1979. His field of study inlcudes Scientific Computations and Parallel Algorithms. His research interests are Computer Aided Learning, Parallel Algorithms, Numerical Scientific Computations and Computational Fluid Mechanics

Professor El-Seoud held different academic positions at TUD Germany. Letest Full-Professor in 1987. Outside Germany, Professor El-Seoud spent different years as a Full-Professor of Computer Science at SQU – Oman, Qatar University, and PSUT-Jordan and acted as a Head of Computer Science for many years. At industrial institutions, Professor El-Seoud worked as Scientific Advisor and Consultant for the GTZ in Germany and was responsible for establishing a postgraduate program leading to M.Sc. degree in Computations at Colombo University / Sri-Lanka (2001 – 2003). He also worked as Application Consultant at Automatic Data Processing Inc., Division Network Services in Frankfurt/Germany (1979 – 1980). Professor El-Seoud joined The British University in Egypt (BUE) in 2012. Currently, he is Basic Science Coordinator at the Faculty of Informatics and Computer Science (ICS) at BUE. Professor El-Seoud has more than 150 publications in international proceedings and reputable international journals.

Article submitted 10 October 2018. Resubmitted 11 November 2018. Final acceptance 16 January 2019. Final version published as submitted by the authors.