Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2020.013861
Article

Tech Science Press

Experimental Evaluation of Clickbait Detection Using Machine Learning Models

Iftikhar Ahmad[1,*], Mohammed A. Alqarni[2], Abdulwahab Ali Almazroi[3] and Abdullah Tariq[1]

[1]Department of Computer Science and Information Technology, University of Engineering and Technology, Peshawar, Pakistan
[2]University of Jeddah, College of Computer Science and Engineering, Department of Software Engineering, Jeddah, Saudi Arabia
[3]University of Jeddah, College of Computing and Information Technology at Khulais, Department of Information Technology, Jeddah, Saudi Arabia
[*]Corresponding Author: Iftikhar Ahmad. Email: ia@uetpeshawar.edu.pk
Received: 24 August 2020; Accepted: 25 September 2020

Abstract: The exponential growth of social media has been instrumental in directing the news outlets to rely on the stated platform for the dissemination of news stories. While social media has helped in the fast propagation of breaking news, it also has allowed many bad actors to exploit this medium for political and monetary purposes. With such an intention, tempting headlines, which are not aligned with the content, are being used to lure users to visit the websites that often post dodgy and unreliable information. This phenomenon is commonly known as clickbait. A number of machine learning techniques have been developed in the literature for automatic detection of clickbait. In this work, we consider six state of the art and classical machine learning algorithms, namely Support Vector Machine (SVM), Logistic Regression (LR), Naïve Bayes Classifier (NBC), Long Short Term Memory (LSTM), Parallel Convolutional Network (PNN), and Bidirectional Encoder Representations from Transformers (BERT) for automated clickbait detection. We also use four performance evaluation metrics, namely accuracy, precision, recall and F1-score to evaluate the performance of the selected set of machine learning algorithms on a real world data set. The results show that BERT is the best performing learning algorithm on three out of four evaluation metrics, and it achieves an average performance superiority of 3%–4% over all the other algorithms. Furthermore, it is observed that PNN has the worst performance among the selected algorithms.

Keywords: Clickbait; machine learning; BERT; social media

1 Introduction

Social networking, an integral part of our daily lives, not only enables us to effectively communicate with one another but also facilitates exchange of views and ideas from the comfort of our homes. As a matter of fact, this exposure to social media defines human relationships as well as the ways in which people interact and connect with one another. Social networking applications also help people understand various cultures and can play a key role in cross culture understanding. Besides, many users rely on

social networking sites for news, information and knowledge, thus allowing them to be a major source of referral for news media outlets.

While social media is ubiquitous and evolving with the highlighted positive aspects, its negative impact cannot go unnoticed. There are certain negative use cases as well where users post fake news, articles and videos to grab attention of people for various monetary and political purposes [1]. Clickbait is one of the examples of negative use-case of social media. A clickbait is a misleading headline of an article whose contents are not related to the headline. Generally, the headline text is false and is designed to create sensational news [2] that adds further to the curiosity in the mind of a reader. Such a use of tempting and misleading headline encourages the user to read the complete article [3]. Interestingly, such content usually spreads more frequently using social media websites such as Facebook and Twitter [3]. The objective is to increase traffic to a particular website which in turn can lead to increase in revenue. Fig. 1 is a sample example of clickbait. However, from a user perspective, the experience is a nuisance, and results in degradation of user experience. This can ultimately lead to users leaving the social networking site, or at least reduce its use. Therefore, automatic detection of clickbait is a contemporary challenge for social networking sites.

Figure 1: An example of clickbait

To address the issue of automatic detection of clickbait, a number of techniques have been developed by researchers [1,3–5]. The objective of this study is to conduct an experimental evaluation of a variety of machine learning models on a real-world data set, and report the results based on accuracy, precision, recall and F1 score.

The paper is organized as follows: Section 2 provides a concise summary of the literature addressing clickbait detection problem. Section 3 presents the experimental setup that includes methodology, data set description, set of algorithms and performance evaluation metrics. Results and discussions are presented in Section 4. Finally, Section 5 concludes the work and provides directions for future research.

2 Literature Review

In order to cope with the challenge of clickbait, social media giants like Facebook have taken significant steps against clickbait articles that show up on a user's timeline. Nonetheless, clickbait has not been eliminated and is still the undesirable aspect of social media. Similarly, twitter also has proposed to

handle clickbait introducing some useful tools. The tools mark clickbait heading on-page that is loaded in users' browser helping them to avoid clicking, and thus becoming a victim. The tools are continuously updating their clickbait database with the intent to create more awareness among the people on how to tackle this issue. In the following text, we summarize the main contributions in the domain of clickbait detection using machine learning techniques.

Agarwal [1] presented a model based on convolutional neural network for the automatic detection of clickbait. The author created his own *corpus* from social media platforms and used the concept of learning features to avoid the complicated and laborious task of feature engineering. The author reported an average accuracy of 0.9 and precision and recall score of 0.85 and 0.88 respectively. However, he did not identify the most relevant features that distinguished clickbait from non-clickbait. Bourgonje et al. [2] proposed a model for stance detection in articles by comparing the headlines with body of the text. The proposed mechanism relies on lemmatization-based n-gram matching approach to classify the headline/article-text pair into related and unrelated classes. With the suggested model, the authors claimed to achieve a weighted accuracy of 0.89. Zheng et al. [3] also proposed a model for clickbait detection utilizing word-embedding and type-related word meaning. The model also takes into consideration the loss function which affects type-related word. The dataset contains a variety of articles from four Chinese news websites. The authors reported an accuracy of 0.80, precision of 0.73, and a recall score of 0.88.

Dimpas et al. [4] considered the clickbait problem prevalent in Philippines social media users and proposed Bidirectional Long Short Term Memory architecture for automated detection of clickbait headlines. A differentiating feature of the proposed model is the effectiveness on both Filipino and English languages. The authors claimed to achieve an accuracy of 0.91. Dong et al. [5] focused on identifying the relationship between title and contents of articles to differentiate between clickbait and non-clickbait headings. The authors proposed clickbait detection based on deep similarity-aware attentive model and it has the ability to identify similarities between the article headings and contents. The model is evaluated on a real world data set and is reported to achieve an average accuracy of 0.85 and 0.88 on Clickbait Challenge and FNC data sets respectively. Shang et al. [6] considered clickbait problem in YouTube videos where title and thumbnail video differs significantly from the contents of the video. The authors proposed Online Video Clickbait Protector (OVCP) to identify clickbait contents based on the comments posted by the users. The differentiating factor of OVCP is the reliance on only user comments. For more details and related works, the reader is referred to [7–11].

3 Experimental Setup

This section presents the methodology adopted for conducting experiments, data set description, set of machine learning models selected for comparison, and the performance measures.

3.1 Methodology

The methodology utilized to conduct the experiments in this research is as follows:

1. A number of headlines were collected from major websites such as Buzzfeed, Viral Nova, Up worthy, Wiki News, New York Times, The Hindu and The Guardian.
2. After the data collection, the dataset was prepared by classifying the data as click bait and non-clickbait for the purpose of training and testing of different models.
3. Following, the desired features were extracted from the data set.
4. The data was then divided into training and test sets (70%–30% split).
5. The underlying model was further trained on the training set.
6. Lastly, the model was evaluated on the test set and record performance measure indicators.

3.2 Dataset

The dataset is collected from various online news outlets. The websites include Buzzfeed, Viral Nova, Up worthy, Wiki News, New York Times, The Hindu and The Guardian. The dataset is classified into clickbait and non-clickbait articles. A total of 32,000 clickbait and non-clickbait articles were collected for the purpose.

3.3 Clickbait Detection Algorithms

The following text discusses the working principles of the selected set of algorithms for clickbait detection problem.

3.3.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) is categorized as a supervised learning algorithm mainly used for classification. SVM differentiates between various classes by drawing a line or hyperplane among various classes of dataset. For unseen instances, it matches the data to a specific hyperplane and assigns the classes accordingly. SVM can be categorized as linear, Lagrangian and non-linear [12].

3.3.2 Logistic Regression (LR)

Logistic regression (LR) models are commonly used for classification problems where an input instance is desired to be classified among one of the pre-defined classes. Generally, LR models are used for binary classification, however, they can be extended to use for multi-class classification as well. Sigmoid function is commonly employed in LR models to achieve dichotomous/binary output.

3.4 Naïve Bayes Classifier (NBC)

Naïve Bayes classifier (NBC) is a simplified Bayesian probability model where the probability of one feature/variable is not affected by any other feature/variable. Let us assume that $X = (x_1, x_2, \ldots, x_n)$ represents n features/variables of a problem instance for a K class classification problem. Naïve Bayes assigns X a probability $p(C_k|x_1, x_2, \ldots, x_n)$, where $k \in \{1, 2, \ldots, K\}$. NBC uses maximum a-posterior decision rule to assign a class $\hat{y}$ to an instance X with n features/variables as following:

$$\hat{y} = \underset{k \in \{1,2,\ldots,K\}}{\arg\max}\ p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k) \tag{1}$$

3.5 Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) belongs to Recurrent Neural Networks (RNNs) and were proposed by Hochreiter and Schmidhuber [13] and later refined by many other researchers [14,15]. By design, RNNs suffers from two main problems, i.e., vanishing and exploding gradients. Thus, they are unable to learn long term dependencies. Just like other neural network models, LSTM consists of input layers; one or more hidden layer, and an output layer. The differentiating feature of the hidden layers of LSTM is the presence of memory cells. Each memory cell includes 3 gates, namely forget gate, input gate, and output gate. Forget gate is responsible for the decision of removal of unnecessary information from the memory. Input gate decides which information should be included or added to the memory, and output gate defines which information should be considered as output memory. For more details about the working on LSTM, the reader is referred to Graves [16].

3.6 Parallel Convolutional Network (PNN)

Parallel Neural Network (PNN) is a type of deep neural network architecture that parallelizes a number of individual neural networks in combination. PNNs are able to solve more complex problems and thus

achieve more robust results on many real-world problems [17]. The architecture of PNN consists of two parts. The first part is a set of neural networks. The job of each neural network is to identify the salient features using the predefined architecture. The second module of the PNN is responsible for identifying the optimal weights. The optimal weights are calculated in the training phase by minimizing the sum of the squared estimated error over the training set. A comprehensive working mechanism of PNN is provided in Chen et al. [17].

3.7 Bidirectional Encoder Representations from Transformers (BERT)

During the last decade, researchers have shown the utility of "transfer learning". Transfer learning is a concept in machine learning where knowledge gained in solving one problem can be utilized in another related problem. A well-known example of such model is ImageNet. Although, the transfer learning technique was successfully applied in many computer vision problems, the gap still existed for natural language processing. Bidirectional Encoder Representations from Transformers (BERT) is a type of transfer learning model suitable for natural language processing. Transformers are used as building blocks of BERT, which is an attention mechanism to identify contextual relations between two words. In its simplistic form, transformers include two separate models. In the first place, it uses an encoder for reading the input text, and then a decoder is used to produce a prediction for the task. Unlike the traditional language processing models, which reads and processes the text in one direction (left to right or vice versa), transformer encoder has the ability to read and process the complete sentence at once. For detailed architectural overview and working of BERT, the reader is referred to [18].

3.8 Performance Evaluation Metrics

Accuracy, precision, recall, and F1 score are used, in this study, as performance evaluation metrics to evaluate our selected set of techniques. However, it is significant to consider the concept of confusion matrix that will help in comprehending the need to use the performance evaluation metrics, selected for this study. Confusion matrix presents summary of accurate and mis-classified instances for a classification problem. Fig. 2 is a graphical depiction of a confusion matrix. It has four key entries namely, True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). Each of the term is defined as follows;

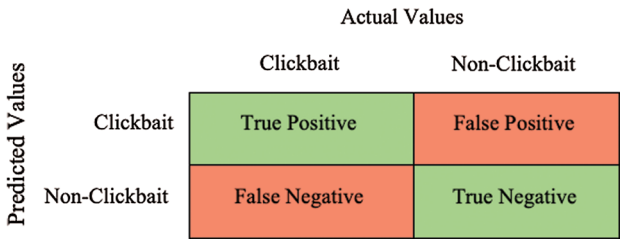

Figure 2: Confusion matrix

True Positive (TP): A positive problem instance is classified as positive. For instance, when a clickbait article is classified as clickbait. The objective is to increase the count of TP.

False Positive (FP): A negative problem instance is classified as positive. For instance, when a non-clickbait article is classified as clickbait. The aim is to minimize the count of FP.

False Negative (FN): A positive problem instance is classified as negative. For instance, when a clickbait article is classified as non-clickbait. The count of FN should be minimized.

True Negative (TN): A negative problem instance is classified as negative. For instance, when a non-clickbait article is classified as non-clickbait. It is desired to maximize the count of TN.

Although confusion matrix is a good indicator of the performance of a learning algorithm, the four values cannot be taken independently and must be considered as a set. Therefore, we use accuracy, precision and F1-score to examine the effectiveness of various learning algorithms. The terms are defined as follows.

3.8.1 Accuracy

Accuracy reflects the ratio of true prediction (TP and TN) against the overall predictions and is calculated as following:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

3.8.2 Precision

Precision reflects the ratio of correctly predicted true instance over the total number of positively predicted instances, and calculated as:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

3.8.3 Recall

Recall reflects the ratio of currently predicted true instance over the total number of positive instances, and is calculated as:

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

3.8.4 F1-Score

F1-score is the harmonic mean of precision and recall, and is calculated as:

$$\mathrm{F1} = 2\,\frac{\mathrm{Precision}\ .\ \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{5}$$

4 Results and Discussions

This section discusses the results based on the aforementioned four performance evaluation metrics.

4.1 Accuracy

The results of this study show that the best accuracy is achieved by BERT (0.977), whereas the worst accuracy is recorded by PNN (0.92). It has been observed that BERT achieves 3% better accuracy than the average of the remaining learning algorithms. The study also reveals that logistic regression (LR) achieves better accuracy (0.969) than the more sophisticated learning algorithms such as LSTM and PNN. Furthermore, the results determine SVM as the second-best performance learning algorithm while achieving an accuracy of 0.97.

Fig. 3 summarizes the accuracy achieved by various learning algorithms on the dataset.

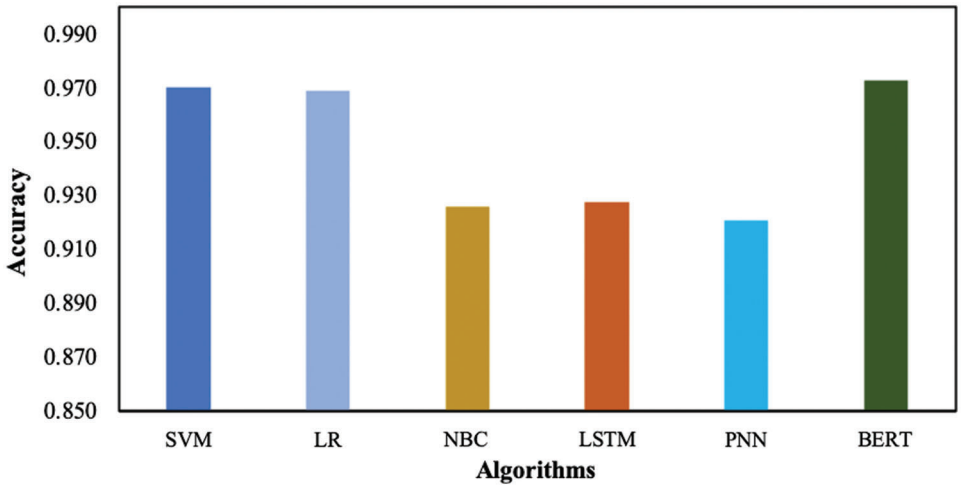

Figure 3: Accuracy achieved by learning algorithms

4.2 Precision

In terms of precision as performance evaluation measure, both BERT and SVM are found to be the best performing algorithms. BERT and SVM achieve a precision score of 0.979 which is slightly better than the average of the rest of the algorithms. The difference is 1% only. It has also been shown that PNN is the worst performing algorithm, whereas LR, NBC and LSTM perform on a comparable level. The performance is summarized in Fig. 4.

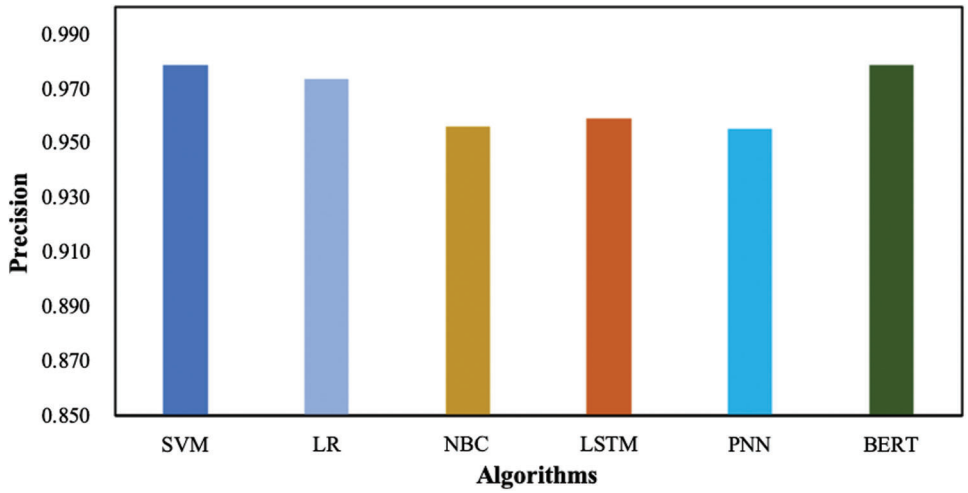

Figure 4: Precision achieved by learning algorithms

4.3 Recall

Measuring the performance of learning algorithms on recall, it has been observed that LR achieves a recall score of 0.98 which is the best among the considered set of algorithms. The second-best performance is observed for BERT; however, the difference is less than 1%. The results (Fig. 5) indicate that the worst performing algorithm in term of recall score is PNN.

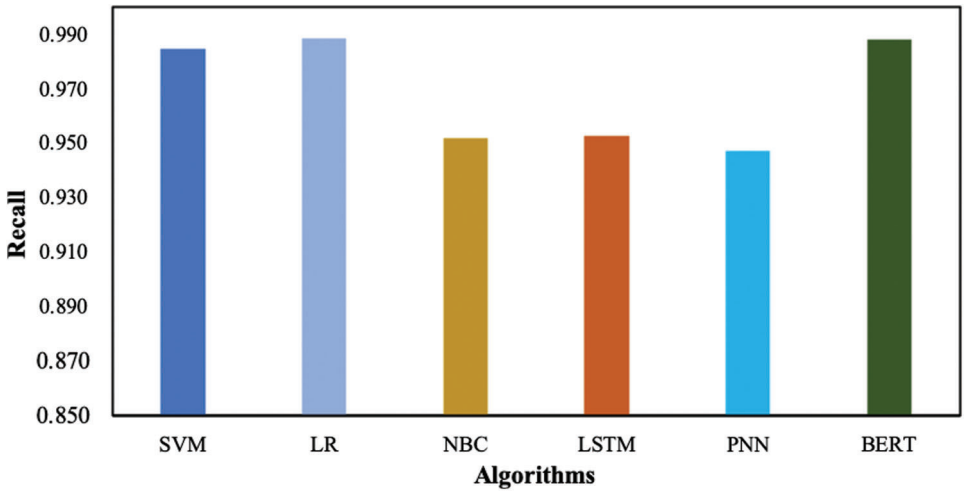

Figure 5: Recall achieved by learning algorithms

4.4 F1-Score

Fig. 6 is a pictorial representation of the algorithms' performance in term of F1-score. The results suggest that BERT has exceeded other performance learning algorithms having 0.983 in terms of F1-score. Moreover, SVM and LR both have achieved F1-score of 0.98, following BERT. The worst performing algorithm is PNN that achieved F1-score of 0.94.

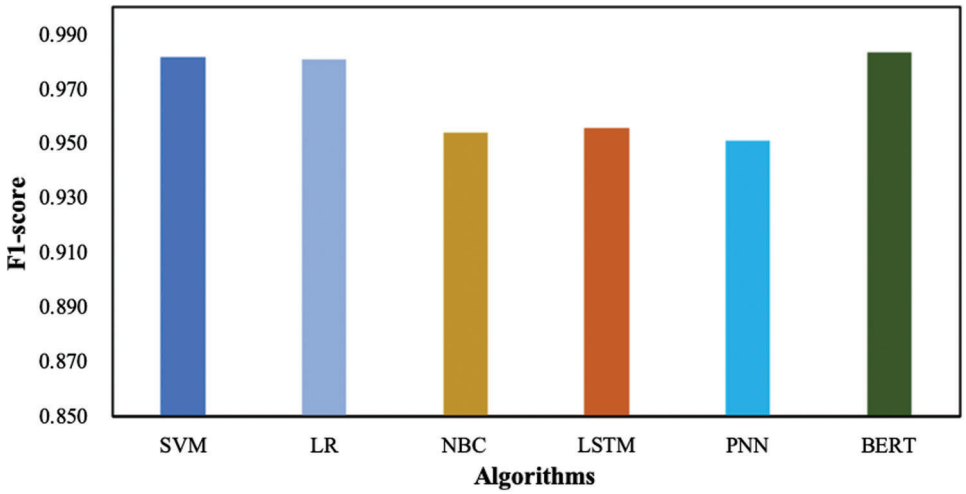

Figure 6: F1-score achieved by learning algorithms

The comparison of the overall results indicates that BERT has achieved the finest results among the selected set of learning algorithms. The findings further reinforce the widely known notion about BERT as the performing technique relatively better for natural language processing. However, the results do not meet the generally believed status of PNN as its performance is not up to the par. s. Interestingly, the results further highlight that the performance of LR and NBC algorithms achieved a comparable level of performance better than LSTM and PNN.

5 Conclusion

Clickbait detection, as a result of a surge in the use of social media, is a contemporary problem faced by its ever-increasing users. The highlighted challenge is a matter of concern for both the service providers as well as the social media users. In addition, the rise of clickbait makes the users vulnerable to unreliable content, and thus negatively affects the user experience with the social networking websites. In this study, we experimentally evaluated a number of machine learning algorithms for automated detection of clickbait. We collected the data set from renowned websites and performed extensive experiments. Our study identified BERT as the most apt learning algorithm performing well on various performance evaluation measures. Rather surprisingly, PNN is observed to be the one with subpar performance.

There is still a considerable gap in the quest for the design of a universally applicable clickbait detection model that can perform well across a variety of datasets of different languages. It will be interesting to see how the current models behave when tested on datasets from various languages whose underlying syntax and semantics are different from English.

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Agrawal, "Clickbait detection using deep learning," in *Proc. 2nd Int. Conf. on Next Generation Computing Technologies*, Dehradun, pp. 268–272, 2016.

[2] P. Bourgonje, J. M. Schneider and G. Rehm, "From clickbait to fake news detection: An approach based on detecting the stance of headlines to articles," in *Proc. EMNLP Workshop: Natural Language Processing Meets Journalism*, Copenhagen, Denmark, pp. 84–89, 2017.

[3] H. T. Zheng, J. Y. Chen, X Yao, A. K. Sangaiah, Y. Jiang *et al.,* "Clickbait convolutional neural network," *Symmetry*, vol. 10, no. 5, pp. 1–12, 2018.

[4] P. K. Dimpas, R. V. Po and M. J. Sabellano, "Filipino and english clickbait detection using a long short term memory recurrent neural network," in *Proc. Int. Conf. on Asian Language Processing*, Singapore, pp. 276–280, 2017.

[5] M. Dong, L. Yao, X. Wang, B. Benatallah and C. Huang, "Similarity-aware deep attentive model for clickbait detection," in *Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, Cham: Springer, pp. 56–69, 2019.

[6] L. Shang, D. Y. Zhang, M. Wang, S. Lai and D. Wang, "Towards reliable online clickbait video detection: A content-agnostic approach," *Knowledge-Based Systems*, vol. 1820, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0950705119303260.

[7] A. Pujahari and D. S. Sisodia, "Clickbait detection using multiple categorisation techniques," *Journal of Information Science*, vol. 1999, no. 5, 016555151987182, 2019.

[8] D. S. Sisodia, "Ensemble learning approach for clickbait detection using article headline features," *Informing Science: The International Journal of an Emerging Transdiscipline*, vol. 22, pp. 31–44, 2019.

[9] D. M. Daoud and S. A. El-Seoud, "An effective approach for clickbait detection based on supervised machine learning technique," *International Journal of Online and Biomedical Engineering*, vol. 15, no. 3, pp. 21–32, 2019.

[10] F. Liao, H. H. Zhuo, X. Huang and Y. Zhang, "Federated hierarchical hybrid networks for clickbait detection." *arXiv preprint*, *arXiv:1906.00638*, 2019.

[11] B. Naeem, A. Khan, M. O. Beg and H. Mujtaba, "A deep learning framework for clickbait detection on social area network using natural language cues," *Journal of Computational Social Science*, vol. 3, no. 1, pp. 231–243, 2020.

[12] S. Suthaharan, *Machine Learning Models and Algorithms for Big Data Classification*. Boston, MA: Springer, 2016. [Online]. Available: https://link.springer.com/book/10.1007/978-1-4899-7641-3.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] F. A. Gers, J. Schmidhuber and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Proc. 9th Int. Conf. on Artificial Neural Networks*, Edinburgh, UK, pp. 850–855, 1999.

[15] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5–6, pp. 602–610, 2005.

[16] A. Graves, "Generating sequences with recurrent neural networks." *arXiv preprint*, *arXiv: 1308.0850*, 2013.

[17] Y. Chen, E. W. Sun and Y. Lin, "Machine learning with parallel neural networks for analyzing and forecasting electricity demand," *Computational Economics*, vol. 56, no. 2, pp. 569–597, 2020.

[18] J. Devlin, M. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint*, *arXiv:1810.04805*, 2018.