applied sciences

MDPI

Article

An Improved Multiple Features and Machine Learning-Based Approach for Detecting Clickbait News on Social Networks

Mohammed Al-Sarem [1,*], Faisal Saeed [1,2,*], Zeyad Ghaleb Al-Mekhlafi [3,*], Badiea Abdulkarem Mohammed [3], Mohammed Hadwan [4,5], Tawfik Al-Hadhrami [6], Mohammad T. Alshammari [3], Abdulrahman Alreshidi [3] and Talal Sarheed Alshammari [3]

[1] College of Computer Science and Engineering, Taibah University, Medina 42353, Saudi Arabia
[2] Institute for Artificial Intelligence and Big Data, Universiti Malaysia Kelantan, City Campus, Pengkalan Chepa, Kota Bharu 16100, Kelantan, Malaysia
[3] College of Computer Science and Engineering, University of Ha'il, Ha'il 81481, Saudi Arabia; b.alshaibani@uoh.edu.sa (B.A.M.); md.alshammari@uoh.edu.sa (M.T.A.); ab.alreshidi@uoh.edu.sa (A.A.); talal.alshammari@uoh.edu.sa (T.S.A.)
[4] Department of Information Technology, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia; m.hadwan@qu.edu.sa
[5] Department of Computer Science, College of Applied Sciences, Taiz University, Taiz 6803, Yemen
[6] School of Science and Technology, Nottingham Trent University, Nottingham NG11 8NS, UK; tawfik.al-hadhrami@ntu.ac.uk
[*] Correspondence: msarem@taibahu.edu.sa (M.A.-S.); fsaeed@taibahu.edu.sa (F.S.); ziadgh2003@hotmail.com (Z.G.A.-M.)

check for updates

Citation: Al-Sarem, M.; Saeed, F.; Al-Mekhlafi, Z.G.; Mohammed, B.A.; Hadwan, M.; Al-Hadhrami, T.; Alshammari, M.T.; Alreshidi, A.; Alshammari, T.S. An Improved Multiple Features and Machine Learning-Based Approach for Detecting Clickbait News on Social Networks. *Appl. Sci.* **2021**, *11*, 9487. https://doi.org/10.3390/app11209487

Academic Editors: Giovanni Randazzo, Anselme Muzirafuti and Dimitrios S. Paraforos

Received: 10 September 2021
Accepted: 6 October 2021
Published: 13 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

Abstract: The widespread usage of social media has led to the increasing popularity of online advertisements, which have been accompanied by a disturbing spread of clickbait headlines. Clickbait dissatisfies users because the article content does not match their expectation. Detecting clickbait posts in online social networks is an important task to fight this issue. Clickbait posts use phrases that are mainly posted to attract a user's attention in order to click onto a specific fake link/website. That means clickbait headlines utilize misleading titles, which could carry hidden important information from the target website. It is very difficult to recognize these clickbait headlines manually. Therefore, there is a need for an intelligent method to detect clickbait and fake advertisements on social networks. Several machine learning methods have been applied for this detection purpose. However, the obtained performance (accuracy) only reached 87% and still needs to be improved. In addition, most of the existing studies were conducted on English headlines and contents. Few studies focused specifically on detecting clickbait headlines in Arabic. Therefore, this study constructed the first Arabic clickbait headline news dataset and presents an improved multiple feature-based approach for detecting clickbait news on social networks in Arabic language. The proposed approach includes three main phases: data collection, data preparation, and machine learning model training and testing phases. The collected dataset included 54,893 Arabic news items from Twitter (after pre-processing). Among these news items, 23,981 were clickbait news (43.69%) and 30,912 were legitimate news (56.31%). This dataset was pre-processed and then the most important features were selected using the ANOVA F-test. Several machine learning (ML) methods were then applied with hyper-parameter tuning methods to ensure finding the optimal settings. Finally, the ML models were evaluated, and the overall performance is reported in this paper. The experimental results show that the Support Vector Machine (SVM) with the top 10% of ANOVA F-test features (user-based features (UFs) and content-based features (CFs)) obtained the best performance and achieved 92.16% of detection accuracy.

Keywords: ANOVA-test; clickbait news; feature selection; social network

1. Introduction

Currently, social networks have become the main environment for communicating, sharing, and posting news on the Internet. Twitter, Facebook, and Instagram are the main social networks that are used to share our opinions and news. With this development, a huge amount of textual data are posted on these media, which increasingly become difficult to process manually. Although the social networks provide an easy way to express our opinions, this platform also can be used to share misinformation in the form of news and advertisements. This is a very serious issue, because this misinformation has the power to influence individuals and sway their opinions. Therefore, finding a way to protect users of social networks from the spread of this misinformation and develop a reliable mechanism to detect it is very important. This misinformation can take the form of clickbait, which aims at enticing the users into clicking a link to news items or advertisements, whose titles (headlines) do not completely reflect the inside contents. According to Chen et al. [1], clickbait is defined as "Content whose main purpose is to attract attention and encourage visitors to click on a link to a particular web page".

The automatic detection of clickbait headlines from the huge volume of news on social networks has become a difficult research issue in the field of data science. Some previous efforts have utilized machine learning to detect clickbait headlines automatically. For instance, Biyani et al. [2] applied Gradient Boosted Decision Trees (GBDT) on a dataset drawn from news sites such as Huffington Post, New York Times, CBS, Associated Press and Forbes. The dataset contains 1349 clickbait and 2724 non-clickbait webpages. The best results achieved were an F1-score of 61.9% with five-fold cross-validation for the clickbait class and an F1-score of 84.6% for the non-clickbait category. Potthast et al. [3] applied linear regression, Naïve Bayes, and random forest methods on a dataset gathered from Twitter. The dataset contained 2992 data points. The results recorded were relatively close, with an approximate precision of 75%.

Chakraborty et al. [4] built a browser extension that used support vector machine (SVM), decision tree, and random forest to automatically detect the clickbait headlines. For training purpose, they collected a well-balanced dataset which contains 30,000 headlines (clickbait and non-clickbait) from ViralStories, Upworthy, BuzzFeed, Wikinews, Scoopwhoop, and ViralNova. In addition, for each data point in the dataset, they extracted sentence structure, clickbait language, word patterns, and n-gram features. The results they achieved are as follows: SVM: an accuracy rate of 93% with 95% precision, 90% recall, 93% F1-score, and 97% ROC-AUC values; Decision Tree: 90% accuracy rate with 91% precision, 89% recall, 90% F1-score, and 90% ROC-AUC values; Random Forest: 92% accuracy rate, 94% precision, 91% recall, 92% F1-score, and finally; ROC-AUC values of 97% using a combination of all extracted features.

Khater et al. [5] proposed the use of logistic regression and linear SVM. They extracted 28 features from a dataset provided by Bauhaus-Universität Weimar at the time of a clickbait detection challenge. The most commonly extracted features were Bag of Words (BOW), noun extraction, similarity, readability, and formality. The best results achieved were 79% and 78% precision for logistic regression and linear SVM respectively. Since the methods of the first category require extracting and labeling each feature before feeding the data into the machine learning tool, researchers have found that deep learning techniques are useful to overcome the feature engineering phase. For instance, López-Sánchez et al. [6] combined metric learning with a CNN deep learning algorithm by integrating them with case-based reasoning methodology. For feature selection, they used TF-IDF, n-gram, and 300 dimensional Word2Vect using the dataset provided by [4]. The proposed approach achieved average areas of 99.4%, 95%, and 90% under the ROC curve using Word2vec, TF-IDF, and n-gram count. Agrawal [7] also used a CNN model to classify a manually constructed news corpus obtained from Reddit, Facebook, and Twitter social networks into clickbait and non-clickbait. As feature selection methods, they used Click-Word2vec and Click-scratch. The highest results that they achieved were 89% accuracy with 87% ROC-AUC score for Click-scratch features and 90% when the Click-Word2vec was used.

Kaur et al. [8] also proposed a hybrid model where a CNN model is combined with LSTM. They found that the CNN-LSTM model when implemented with pre-trained GloVe embedding yields the best results, based on accuracy, recall, precision, and F1-score performance metrics. They also identify eight other types of clickbait headlines: reaction, reasoning, revealing, number, hypothesis/guess, questionable, forward referencing, and shocking/unbelievable. They also found that shocking/unbelievable, hypothesis/guess, and reaction clickbait types to be the most frequently occurring types of clickbait headlines published online.

Although several machine learning approaches have been proposed to detect clickbait headlines, most of these recent methods are not very robust. The previous studies used hybrid categorization techniques such as Gradient Boosted Decision Trees, linear regression, Naïve Bayes and random forest methods, SVM, decision tree, logistic regression, and convolutional neural network deep learning. Most of these studies used datasets with headlines written in English. However, this paper uses an Arabic language dataset and proposes a comprehensive approach that includes three main phases: data collection, data preparation, and machine learning model training and testing phases. This dataset was pre-processed and then the most important features were selected using ANOVA F-test. Several machine learning methods were then applied which include random forest (RF), stochastic gradient descent (SGD), Support Vector Machine (SVM), logistic regression (LR), multinomial Naïve Bayes (NB), and k-nearest neighbor (k-NN). Hyper-parameter tuning methods were applied to ensure finding the optimal settings. Finally, the ML models were evaluated and the overall performance is reported here. The key contributions of this paper are as follows:

- We constructed the first Arabic clickbait headline news dataset. The raw dataset is available publicly for research purpose.
- We extracted a set of user-based features and content-based features for the constructed Arabic clickbait dataset.
- We implemented six machine learning-based classifiers, including Random Forest (RF), Stochastic Gradient Descent (SGD), Support Vector Machine (SVM), Logistic Regression (LR), Multinomial Naïve Bayes (NB), and k-Nearest Neighbor (k-NN).
- We proposed an effective approach for enhancing the detection process using a feature selection technique, namely a one-way ANOVA F-test.
- We conducted extensive experiments, and the results show that the proposed model enhances the performance of some classifiers in terms of accuracy, precision, and recall.

2. Related Works

2.1. Characteristics of Clickbait News

Biyani et al. [2] define eight types of clickbait, which include exaggeration, teasing, inflammatory, formatting, graphic, bait-and-switch, ambiguous, and wrong. In exaggeration, the title overdraws the content on the target page. Teasing means hiding the details from the title to build more suspense. In the inflammatory type, inappropriate or vulgar words are phrased. Formatting means overusing the capitalization/punctuation in the headlines, for instance ALL CAPS or exclamation points are used. In graphic types, the subject matter is disturbing or unbelievable. Bait-and-switch means the news included in the title is not found at the target page. Ambiguous means the title is unclear or confusing, while wrong means using a plainly incorrect article. Kaur et al. [8] also identify eight other types of clickbait headlines: reaction, reasoning, revealing, number, hypothesis/guess, questionable, forward referencing, and shocking/unbelievable. They also found that shocking/unbelievable, hypothesis/guess, and reaction clickbait types to be the most frequently occurring types of clickbait headlines published online.

According to Zheng et al. [9], different ways of attracting users' attention are used by the headlines of different article types, which means the characteristics of clickbait vary between article types. This is different from traditional text-analysis issues. For instance, the headlines of forums or blogs are more colloquial than the headlines of other traditional news. The main difference between these two types of headlines is the use of functional linguistic characteristics such as wondering, exaggerating, and questioning. In [9], two types of characteristics were used: general clickbait, and the type-related characteristics, while the main characteristics used by Naeem et al. [10] for detection of clickbait were sensationalism, mystery, notions of curiosity, and shock.

In another approach, Potthast et al. [3] used three types of features for clickbait headlines, which are: the teaser message, the linked web page, and meta information. The first type includes basic text statistics and dictionary features, while the second type analyses the web pages linked from a tweet, and the third type includes meta information about the tweet's sender, medium, and time.

Bazaco, Redondo, and Sánchez-García [11] describe the characteristics of clickbait using six variables under two categories: presentation variables and content variables. The first category includes incomplete information, appealing expressions, repetition and serialisation, and exaggeration, while the second type includes the use of soft news and sensationalist content and striking audiovisual elements. According to [1], the characteristics of curiosity used in clickbait are: its intensity, tendency to disappoint, transience, and association with impulsivity. These lead to a knowledge gap that are exploited by the clickbait headlines to encourage readers to click through to read the whole article.

2.2. Machine Learning and Deep Learning Methods for Clickbait Detection

Several machine learning and deep learning methods have been applied to detect clickbait headlines from different social networks, including Twitter, Facebook, Instagram, Reddit, and others. Table 1 summarizes recent studies on clickbait detection methods. The results in the table show that the performance of machine learning methods still needs to be improved. In the best cases, the highest accuracy obtained reached 0.87 by [12]. In contrast, the use of deep learning showed a good improvement in performance, where the accuracy obtained by [13] reached 0.97. Most of the existing studies used headlines written in English or other languages. Only a few studies focused on clickbait headlines in Arabic. Although Arabic and English scripts have some similarities, there are a number of characteristics that specify the uniqueness of Arabic script. These include: the direction of Arabic, which is written from right to left, and the fact that neither upper nor lower cases exist in Arabic, which is written cursively. In Arabic, all letters are connected from both sides, except six letters that can be connected from the right side only. Each of the 28 letters of Arabic script has different shapes, depending on its position in the word, and some letters are very similar, differing only in the number and/or the position of dots [14,15]. In addition, there are other special features which are unique to Arabic script such as elongation, morphological characteristics, word meters, and morphemes [16].

Table 1. Summary of recent studies on clickbait detection methods.

Study	Dataset	Classificatio Method(s)	Accuracy of the Model(s)	Issues/Future Directions
[2]	The dataset includes 1349 clickbait and 2724 non-clickbait websites from different news websites whose pages surfaced on the Yahoo homepage.	Gradient Boosted Decision Trees (GBDT)	0.76	(1) Include the non-textual features (example: images and videos) and the comments of users on articles. (2) Find the most effective types of clickbait that can attract clicks and propose methods to block them. (3) Deep learning is proposed to be applied to obtain more indicators for clickbaits.The obtained performance needs to be improved.

Table 1. *Cont.*

Study	Dataset	Classificatio Method(s)	Accuracy of the Model(s)	Issues/Future Directions
[3]	The dataset includes 2992 tweets from Twitter, 767 of which are clickbait.	Logistic regression, naive Bayes, and random forest	0.79	The first evaluation corpus was proposed with baseline detection methods. However, this task needs more investigation to detect clickbait between different social media, and improving the performance of detection. The obtained performance needs to be improved
[17]	Clickbait Challenge 2017 Dataset includes over 21,000 headlines.	Random Forest Regression	0.82	Future works can be: (1) Extract more features; (2) apply other machine learning methods; (3) collect more high-quality data. The obtained performance needs to be improved.
[12]	CLDI dataset from Instagram includes 7769 instances and WCC dataset from Twitter includes 19538 instances.	KNN, LR, SVM, GNB, XGB, MLP,	0.87	Future works: Develop the model as a website or mobile application for Twitter and Instagram. The obtained performance needs to be improved.
[9]	The dataset contains 14,922 headlines, where half of them are clickbait. These headlines are taken from four famous Chinese news websites	Clickbait convolutional neural network (CBCNN)	0.80	The maximum length of the headline is limited. If the headlines are long, this might cause information loss. This needs more investigation to solve information-loss problem and including user-behavior analysis. The obtained performance needs to be improved.
[10]	Dataset of head-lines from Reddit,. The datasets includes 16,000 legitimate news and 16,000 clickbait samples.	LSTM using word2vec word embedding	0.94	The good accuracy was obtained due to the loop back approach that was employed by the LSTM that allows for a better understanding of the context and then better classification of headlines.
[6]	The dataset was collected from Reddit, Facebook and Twitter. It includes 814 clickbait samples and 1574 nonclickbait samples.	Convolutional neural network	0.90	Future works include (1) Find the most important features needed for learning process. (2) Gather more data to develop better models (3) Develop web application that can utilize this model and can alert the user to the clickbait websites.
[13]	The dataset includes 32,000 headlines that includes 16,000 clickbait and 16,000 non-clickbait titles.	Recurrent Convolutional Neural Network (RCNN) + Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU)	0.97	A larger dataset can be used.

Table 1. *Cont.*

Study	Dataset	Classificatio Method(s)	Accuracy of the Model(s)	Issues/Future Directions
[18]	The three datasets (A, B and C) from Clickbait Challenge 2017 were used. It includes 2495, 80,012 and 19,538 respectively.	Self-attentive RNN	0.86	The obtained performance needs to be improved.
[19]	Clickbait Challenge datasets include 20,000 pairs of training and validation posts. FNC dataset includes 49,972 pairs of training and validation posts.	Deep Semantic Similarity Model (DSSM)	0. 86	The other features like image information were not considered in this work. Also, the obtained performance needs to be improved.

To address the lack of study of clickbait detection in Arabic texts, this paper focuses on improving the performance of machine learning methods for detecting clickbait headlines on social networks in the Arabic language.

2.3. Problem Formulation for Clickbait Detection

The clickbait detection problem is a subset of natural language processing that can be represented as a binary classification as follows:

Given a set of shared posts via social networking platforms (tweets) $T = \{t_1, t_2, \ldots, t_n\}$, let $t \in T$ denote a post that is classified into a class $\mathbb{C} = \{\mathbb{C}+, \mathbb{C}-\}$ where $\mathbb{C}+$ is a class of the tweets $t_i \in T$ that are considered as legitimate news, and $\mathbb{C}-$ is the class of the clickbait news $t_j \notin \mathbb{C}+$.

To solve the problem, let D be a dataset of all posts $D = \{V1, V2, \mathbb{C}\}$ where $V1 = \{v_1^1, v_2^1, v_3^1, \ldots, v_n^1\}$ a vector of extracted features from user portfolio (user-based features (UFs)) and $V2 = \{v_1^2, v_2^2, v_3^2, \ldots, v_n^2\}$ is a vector of extracted features from the post/tweet content (content-based features (CFs)). Let also v_i^1 and v_i^2 be the points of a specific feature I and $v_i^1 \in V1$ and $v_i^2 \in V2$.

Let D' be a training set and D'' be a testing set, where D' and $D'' \in D$. Let ξ be a function that generates I from D' and D'' based on the feature space $V : \xi : T \times V \rightarrow I$. As the vector space can be high-dimensional, the clickbait detection problem is now formulated as follows:

Let χ be a function that maps post $t_i \in T$ to $\mathbb{C} = \{\mathbb{C}+, \mathbb{C}-\}$, C: χ: $T \rightarrow C$, where C = $\langle \mathbb{C}, r \rangle$ and r is a binary relation which takes value 1 if a post $t_i \in T$ is a legitimate post and $t_i \in \mathbb{C}$+, and 0 otherwise.

The function χ can now be set as an optimization problem as follows:

optimize $f\chi(\mathrm{V1}, \mathrm{V2})$ subject to $c(V1, V2)$ where c is a constraint set on the search space.

3. Materials and Methods

The proposed multiple-feature-based approach for detecting clickbait news is presented in this section. Since the difference between clickbait and normal news can be distinguished directly by analysis of the linguistic character of news content [20], the proposed approach takes into consideration both the headlines and the content of the news features (CFs). In addition, to overcome the limitations of such approach, they are combined with news content features.

Figure 1 presents the methodology followed in this study, which consists of the following phases: data collection, data preparation, and machine learning model training and testing. For detecting clickbait news on social networks, both of the investigated news and profile of the user who shared the post are collected. We first constructed a baseline dataset from the raw dataset by labelling the news as clickbait or legitimate. Since the amount of collected data was huge and for building a sufficiently satisfactory dataset, we used a pseudo labelling learning (PLL) technique [21]. In the next phase, both of the news headlines and contents are pre-processed, including text cleansing, normalization, stemming, stop word removal, and tokenization. These steps are necessary to enhance the overall performance of the ML-based model. We concatenated the processed text with user-based features and then applied the feature reduction using a one-way ANOVA test. The selected features were fed to the ML model. A set of ML models was tested, and their hyper-parameters were tuned to ensure finding the optimal settings. Finally, the ML model was evaluated, and the overall performance reported.

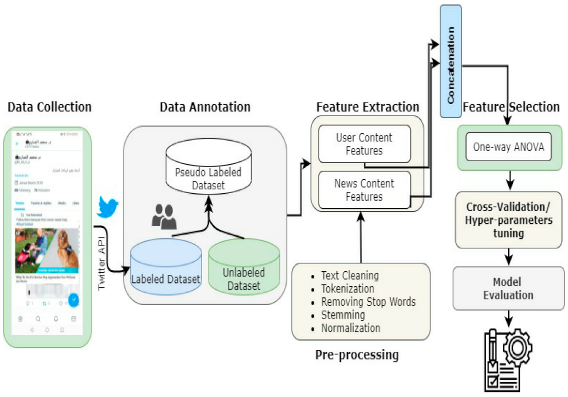

Figure 1. The proposed multiple feature approach for detection of clickbait news.

3.1. Data Collection

We collected 72,321 Arabic news items from Twitter. The dataset can be obtained from github.com (https://github.com/Moh-Sarem/Clickbait-Headlines#clickbait-headlines) (accessed on 1 October 2021). For this purpose, we implemented a special crawler that can access breaking news on social networks by feeding the name of the public breaking news agencies. Often, Twitter APIs return tweets in JSON format. However, because many features are not helpful for the proposed model, the used crawler filters out and saves all the collected information from user profile and shared content in comma-separated values (CSV) format. The details of the collection process through multiple feature analysis are shown in Algorithm 1. In addition, the full description of the features used is presented in Tables 2 and 3.

Algorithm 1 Pseudocode of dataset collection process for extracting UFs and CFs

```
Input: A list of public Twitter breaking news agencies' profiles $N$
Output: Unlabelled dataset with UFs and CFs
For each profile $p \in N$ do:
    Access public page of $p$
    Retrieve all shared tweets $t^p$
    Pull out using Twitter APIs tweet's features (USs)
    If $t^p$ contains an external URL Then:
        Visit the external webpage $p^e$
        For all html tags in $p^e$ do:
            Find html tag that contains news full text (CFs)Compute similarity score between $t^p$ and $p^e$
        End
    End if
    Store the extracted features in csv format
End
```

Table 2. User-based features.

# Feature	Feature Name	Description
UF^1	User ID	Every user has one unique ID.
UF^2	Name	The name of the user who post news on Twitter
UF^3	Screen name	The screen name that this user identifies themselves with.
UF^4	Date of join	The creation time of the user account
UF^5	#Url	Number of URL provided by the user in association with their profile
UF^6	Profile description	A text that shows how the user describs his/her account
UF^7	Verified	A boolean indicator shows whether the user has a verified account or not
UF^8	Count of followers	Total number of followers
UF^9	Count of friends	Numeric value indicates how many friends that the user has
UF^{10}	Count of favorites' accounts	Numeric value indicates how many tweets this user has liked
UF^{11}	Count of public lists	Total number of public lists that this user is a member of.
UF^{12}	Location	The geographical location
UF^{13}	Hashtage	The associated hashtag with the post
UF^{14}	Lang	The post language
UF^{15}	Number of post shared	Total number of content shared by the user

Table 3. Content-based features.

# Feature	Feature Name	Description
CF^1	#Url	Number of external URLs provided by the user in association with news
CF^2	Source	Name of the source of the news article.
CF^3	Headline	The headline of the news article for catching the reader's attention
CF^4	Tweet Text	The body of the tweet news
CF^5	Body Text	The full news in readable format, often with external content
CF^6	Retweet count	Total number of times this tweet has been retweeted.
CF^7	media	Boolean value indicates whether there are associated images or videos
CF^8	Similarity score	The score for similarity between headline text and body text.
CF^9	Creation date	The posted date of the news content

3.2. Data Annotation

Once we obtained the final dataset by using the implemented crawler, we prepared a baseline dataset from the retrieved dataset. Every shared tweet was labelled as a clickbait or legitimate by asking three media professionals to volunteer in judging 12,321 tweets and their associated news. They were asked to access the external webpage by following the URL link provided with tweet and comparing the tweet's body and headline with the full text in the destination webpage. To facilitate this job, we provided them with examples showing what clickbait news looks like. Table 4 shows a guideline for how to classify the content of the shared tweets.

Table 4. Example of clickbait news.

Type	Definition	Example of Arabic Clickbait News	Translation of the Arabic Clickbait News
Ambiguous	Title unclear or confusing to spur curiosity.	هذا الأمر لم يحدث في المملكة؟	This matter did not happen in the kingdom.
Exaggeration	The title exaggerates the content of the landing page.	الراجل ده كريم أوى يا بابا.. زبون يأكل بـ20 دولار ويترك 1400 بقشيش بأمريكا	This man is kind father. In America, the customer eats for $20 and leaves 1400 tips
Inflamm-atory	Either phrasing or use of inappropriate/vulgar words.	اطباء تحت مسمى الطب " الاطباء المجرمين "	Doctors under the name of medicine “criminal doctors”
Teasing	Omission of details from title to build suspense: teasing.	بين ليلة وضحاها... أمريكي يربح مليار دولار	Overnight... an American wins a billion dollars
Formatt-ing	Excessive use of punctuation or exclamation marks.	كيف أنتِ عمياء ومصوِّرة"؟!.. هذه " العبارة السلبية كانت انطلاقة "المطيري"	“How are you blind and a photographer”?!.. This negative phrase was the launch of “Al-Mutairi”
Wrong	Just plain incorrect article: factually wrong.	10 أمور يقوم بها الأغنياء ولا تقوم بها انفسك!	10 things rich people do that you don’t do yourself!
URL redirection	The thing promised/implied from the title is not on the landing page: it requires additional clicks or just missing.	كندا: ينمو الناتج المحلي الإجمالي الحقيقي بنسبة 0.7٪ في نوفمبر مقابل 0.4٪ المتوقعة	Canada: Real GDP grows 0.7% in November vs. 0.4% expected
Incomplete	The title is incomplete	عاجل :تطور في أرامكو و مدينة صناعية... -	Urgent: An improvement in Aramco and an industrial city

As shown in Table 4, there are seven categories that the volunteers could use to label each post as clickbait news. In case of unclearness or doubt about which class the post belongs to, the post is labelled as “incomplete”. Every content text in the baseline dataset has three labels, one provided by each annotator. To assign the final class label, we applied the majority voting algorithm and labelled the content as clickbait or legitimate news. Table 5 shows the details of the baseline dataset, which includes 4325 items of clickbait news and 6743 legitimate items. The news items that are labelled as incomplete were later removed from the dataset. The remaining baseline dataset contained 11,068.

Table 5. Details of baseline dataset.

Parameter	# of Treated Data
Total news in dataset	12,321
Remaining baseline dataset	11,068
% of treated news in respect to the whole dataset	17%
Clickbait news items, %	4325, 35.1%
Legitimate news items, %	6743, 54.72%
Incomplete posts, %	1253, 10.16%
Number of external URLs	4862
Number of breaking news sources	7

As the size of our final baseline dataset was quite small (17% of the original dataset), we decided to apply a pseudo-labelling learning technique to enhance the performance of the ML model. PLL is an efficient semi-supervised technique that can be applied to utilize unlabeled data while training ML models. As shown in Figure 1, the ML model is trained first on the labeled data (in this case: the baseline dataset). The model then predicts the labels of unlabeled data. The predicted pseudo-labels are assigned as target classes for unlabeled data and combined with the original baseline dataset (labeled data). Finally, the produced new dataset is then used to train the proposed ML-models. After applying PLL technique, the size of the labeled dataset was increased to around 54893 instances. Table 6 shows the details of the final dataset after applying the PLL technique on 71.54% of the remaining unlabeled data.

Table 6. Final dataset after applying PLL technique.

Parameter	# of Treated Data
Total no. of news items in the dataset,	54,893
% of treated news items with respect to the whole dataset	75.90%
No. of clickbait items, % of the total news items	23,981, 43.69%
No. of legitimate news items, % of the total items	30,912, 56.31%
Number of external URLs	14,518
Number of breaking news sources	22

3.3. Pre-Processing and Numeric Representation

Beside the UFs and CFs described above in Tables 2 and 3, the "headline" CF^3, "tweet text" CF^4, and "body text" CF^5 features from CFs required additional treatment.

3.3.1. Pre-Processing

For many text classification systems, pre-processing is considered as an essential step to improve the quality of data as well as the efficiency and accuracy of ML models [22,23]. The common pre-processing steps include text cleansing, tokenization, removing stop words, stemming, and normalization. Since the obtained data is pulled out from Twitter and by accessing the external web pages following the URL links associated with the body of the tweets, additional pre-processing steps were performed, such as deletion of unnecessary, insignificant items from texts (e.g., digits, punctuation marks, URLs, special characters, non-Arabic characters, diacritics), and removal of emojis and hashtags.

3.3.2. Numeric Representation

By numeric representation, we mean converting the textual content into a form that could be fed into the ML model in treatable format. In this work, the term frequency-inverse document frequency (TF-IDF) is used as a numeric representation. Mathematically, the TF-IDF can be calculated as in Equations (1)–(3):

$$tf\_idf_{t,D} = TF_{t,D} \times IDF_t \tag{1}$$

where

$$TF_{t,D} = \frac{Number\ Of\ Repetitions\ of\ Term\ t\ In\ a\ Document\ D}{\#\ Of\ terms\ In\ a\ Document} \tag{2}$$

and

$$IDF_t = log \frac{Number\ Of\ Documents}{Number\ Of\ Documents\ Containing\ The\ term\ t} \tag{3}$$

After applying the TF-IDF technique on the final dataset, the training time of ML models was long because of high dimensionality, where the number of extracted features reached 10,230.

3.4. Feature Selection

Feature selection (FS) is an effective way to reduce large data [23]. The main purpose of FS is to delete irrelevant and noisy data. It also enables a representative subset of all data to be chosen to minimize the complexity of the classification process. Several FS techniques can be found in the literature. These include: Mutual Information (MI), Information Gain (IG), improved Chi-square, and the one-way ANOVA F-test [24] (referred to, hereafter as FV-ANOVA). This paper proposes to use FV-ANOVA as a feature selection method that is used to statistically select the important features according to the F-values. The features are sorted in ascending order, where the most relevant features appear on the top. Finding the best cut-point value is a challenge. Thus, we divided the features into a set of groups based on a given percentile (p%) of the original number of features. This step allows us to find the top-scoring features. Later, only the p% top-scoring features were used to train the ML classifiers. The process of selecting features for FV-ANOVA is presented in Algorithm 2.

Algorithm 2 Pseudocode for selecting features-based FV-ANOVA method.

```
Input: $D$-dataset, $V$ features extracted as numeric representation by TF-IDF, $C$-class label and $p$% percentile.
Output: $D_{FS}$ subset of top-scoring features based on the given $p$%
$k \leftarrow$ number of classes in $D$
$N \leftarrow$ number of features in $D$
For each pair $f_j \in (V, C)$ do:
Count number of samples per class
Compute (mean, standard deviation, standard error) of each $f_j$ with respect to $C_i$
Compute degree of freedom between/within classes ($SS_B$, $SS_w$)
Compute sum of square of $(SS_B, \ SS_w)$
Find mean square $MS_B$ between groups as $MS_B = SS_B \ / \ (k-1)$
Find mean square $MS_W$ between groups as $MS_W = SS_W \ / \ (N-k)$
End for
                    $F\_value \leftarrow \frac{MS_B}{MS_W}$
Sort $F\_value$ in ascending order
$D_{FS} \leftarrow$ Select the top-scoring features based on $p$%
Return $D_{FS}$
```

3.5. Feature Selection

Six ML classifiers were implemented: Random Forest (RF), Logistic Regression with Stochastic Gradient Descent (SGD), Support Vector Machine (SVM), Logistic Regression (LR), Multinomial Naïve Bayes (NB), and k-Nearest Neighbor (k-NN). To explore the effectiveness of the proposed feature selection method, we conducted different experiments and employed these classifiers on different subsets of features based on F-values.

For tuning hyper-parameters of the used ML classifiers, the grid search algorithm with k-fold cross-validation is used. Subsequently, the values of hyper-parameters that yield the highest performance measure are set to be the final values of hyper-parameters for each classifier. The set of values of hyper-parameters used in this work is presented in Table 7.

Table 7. List of optimized hyper-parameters of each classifier.

ML Classifier	Hyper-Parameters Used for Tuning the Model	Best Values of Hyper-Parameters

RF	Criterion = [entropy, gini]
	max_depth = [10–1200] + [None]
	min_samples_leaf = [3–13]
	min_samples_split = [5–10]
	n_estimators = [150–1200]
		Criterion = gini
		max_depth = 142
		min_samples_leaf = 3
		min_samples_split = 7
		n_estimators: 300

SGD	alpha = [1×10^{-5}, 1×10^{-4}, 1×10^{-3}, 1×10^{-2}, 0.1, 1]
	loss = [log, hinge]
	max_iter= [10–1000]
	Penalty= [l2′, ‘l1′, ‘elasticnet’]
		alpha= 1×10^{-4}
		loss = log
		max_iter = 1000
		Penalty = l2

SVM	C = [0.1, 1, 10, 1×10^{2}, 1×10^{3}]
	Gamma = [1×10^{-4}, 1×10^{-3}, 1×10^{-2},0.1, 1, 10, 1×10^{2}]
	Kernel = [sigmoid, linear, rbf]
		C = 10
		Gamma = 1×10^{-3}
		Kernel = rbf

LR	C = [1×10^{-3}, 1×10^{-2}, 0.1, 1, 10, 100], fit_intercept = [True, False]	C = 1×10^{-3}
		fit_intercept = True

NB	alpha = [1×10^{-5}, 1×10^{-4}, 1×10^{-3}, 1×10^{-2}, 0.1, 1]fit_prior = [True, False]	alpha = 0.1
		fit_prior = True

K-NN	n_neighbors = [1, 40]	Number of neighbours = 7

3.6. Model Evaluation

To evaluate the performance of classifiers, we computed the accuracy (*Acc*), recall (*R*), precision (*P*), and f1-score (*F*1) metric of each classifier with those features that were selected by the proposed F-values of the one-way ANOVA test. The descriptions of these metrics are shown in Equations (4)–(7) respectively.

$$Acc. = \frac{TP + TN}{D} \tag{4}$$

where (TP + TN) is the accurately predicted content either clickbait or not, D is the total number of samples in the dataset.

$$P = \frac{TP}{TP + FP} \tag{5}$$

where (TP + FP) is the total number of predicted clickbait content.

$$R = \frac{TP}{TP + FN} \tag{6}$$

where $(TP + FN)$ is the total number of actual clickbait content.

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{7}$$

4. Experimental Design

The experiments in this study were performed on Python 3.8 with Windows 10 operating system. We used numerous Python packages including sklearn 0.22.2 for implementing the classifiers, nltk 3.6.2 for pre-processing Arabic text and Beautiful soup 4.9.0 for scraping data from external web pages. The user-based features and content-based features were fed into classifiers separately. Later, we merged both types and measured the performance of ML classifiers based on top-scoring features p% that were selected based on f-values of one-way ANOVA. For ensuring fair comparison between classifiers, the same pre-processing steps and the same set of features were used for each classifier. In addition, we considered four experimental scenarios per feature type, as illustrated in Table 8.

Table 8. Number of features per each experiment.

#	Type of Experiment	Number of Features
UCFs	Baseline:	15
	F_values_5%: 5% of features	4
	F_values_10%: 10% of features	7
	F_values_15%: 15% of features	9
NCFs	Baseline: Including (TF-IDF) extracted features	10,236
	F_values_5%: 5% of features	732
	F_values_10%: 10% of features	2187
	F_values_15%: 15% of features	5867
UCFs + *NCFs*	Baseline:	10,251
	F_values_5%: 5% of the extracted features	736
	F_values_10%: 10% of the extracted features	2194
	F_values_15%: 15% of the extracted features	5876

5. Results and Findings

This section describes and discusses the results for each experiment shown in Table 8. First, we present the findings that were obtained when only the user-based features were used. The accuracy of each classifier is presented in Table 9. The second type of features, content-based features, were then investigated, as shown in Table 10. Finally, we combined both types of features and the performance of classifiers is presented in Table 11.

Table 9. Accuracy of different experiments with user-based features only.

ML Classifier	Experiment			
	Baseline	F_Values_5%	F_Values_10%	F_Values_15%
RF	61.24	61.73	**63.93**	62.34
SGD	59.04	52.65	51.86	57.62
SVM	64.40	62.76	**66.54**	61.08
LR	61.87	61.87	61.87	61.87
NB	61.75	62.03	60.12	61.49
k-NN	60.57	46.83	42.72	41.33

Table 10. Accuracy of different experiments with content-based features only.

ML Classifier	Experiment			
	Baseline	F_Values_5%	F_Values_10%	F_Values_15%
RF	77.97	86.83	90.21	83.67
SGD	74.72	85.72	84.59	79.19
SVM	75.89	89.31	**91.83**	90.37
LR	77.65	75.43	75.76	75.09
NB	74.65	87.35	90.24	89.46
k-NN	76.99	73.77	65.08	65.08

Table 11. Accuracy of different experiments with combination of UFs and CFs.

ML Classifier	Experiment			
	Baseline	F_Values_5%	F_Values_10%	F_Values_15%
RF	77.18	86.94	88.13	85.02
SGD	74.83	82.52	87.02	85.39
SVM	75.00	88.92	**92.16**	90.65
LR	76.77	76.73	75.00	75.87
NB	75.30	89.27	90.74	89.62
k-NN	77.05	74.41	71.02	71.61

Based on the results presented in Table 9–11, the following findings are observed and can be summarized as follows:

- When the content-based features were used, the classifiers performed well and SVM, NB, and RF achieved notable results using 10% of top-scoring features compared to their results in the baseline experiment. Among these methods, SVM obtained the best accuracy (91.83%) for content-based features.
- In most cases of experiments with content-based features, all classifiers showed good results when the one-way ANOVA method was used as feature selection, except k-NN and LR. It is notable that k-NN had the worst performance when the number of selected features increased to 10% and 15%.
- Increasing the percentage of the top-scoring features to more than 10% leads to a reduction in the performance of the ML classifiers.
- RF and SVM benefited more when the user-based features were used, compared to their results in the baseline experiment.
- The result for LR remained constant, and no change was observed when user-based features were fed into the classifier.
- The k-NN and SGD do not benefit from the ANOVA-based feature selection at all for user-based features.

- Combining user-based features and content-based features shows an improvement in the performance of ML classifiers and only LR and k-NN classifiers did not show any improvement.
- The SVM outperforms all other classifiers and benefited more when the proposed feature selection method was used for the combination of user-based features and content-based features. The highest accuracy achieved was 92.16%.
- As the total number of features for the combination of user-based and content-based features is 10,251, selecting the top 10% of these features (2194) was more suitable for SVM, which performed well with low dimensionality data.
- As shown in the results, using the user-based features achieved lower performance than using the content-based features for all ML methods. Therefore, the proposed model relies more on the content-based features and the combined ones.

6. Conclusions

This paper has proposed a comprehensive approach that includes three main phases: data collection, data preparation, and machine learning modeling phases. After collecting the dataset, which is considered the first Arabic clickbait headline news dataset, the pre-processing tasks were performed, which included text cleansing, normalization, stemming, stop words removal, and tokenization. The features of the processed text (content-based features) were then combined with the user-based features and the feature selection was then applied using one-way ANOVA test. Finally, the ML models were applied, which included Random Forest (RF), Stochastic Gradient Descent (SGD), Support Vector Machine (SVM), Logistic Regression (LR), Multinomial Naïve Bayes (NB), and K-nearest Neighbor (k-NN). Hyper-parameter tuning methods were applied to ensure finding the optimal settings. The experimental results showed a great enhancement when the CFs were used and also when a combination of UFs and CFs was used. The accuracy achieved reached 92.12% using 10% of the top-scoring features, which is better than that reported in many previous studies (discussed in the related works). This enhancement is particularly interesting, as we are dealing with Arabic contents. Future work will investigate the application of several deep learning methods on this Arabic dataset in order to enhance the detection performance. Moreover, collecting more Arabic content to add to the dataset will be a beneficial addition to conducting the analysis.

Author Contributions: Conceptualization, M.A.-S., F.S., T.A.-H.; methodology, M.A.-S., F.S.; software, M.A.-S.; validation, T.A.-H., A.A.; formal analysis, F.S., M.H.; investigation, M.T.A., T.S.A.; resources, Z.G.A.-M., B.A.M., M.H., A.A., T.S.A.; data curation, B.A.M.; writing—original draft preparation, M.A.-S., F.S.; writing—review and editing, M.A.-S., F.S.; visualization, F.S.; supervision, M.A.-S., F.S.; project administration, M.A.-S., Z.G.A.-M.; funding acquisition, Z.G.A.-M., B.A.M., M.T.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been funded by the Scientific Research Deanship at the University of Ha'il, Saudi Arabia, through project number RG-20 023.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset can be obtained from https://github.com/Moh-Sarem/Clickbait-Headlines#clickbait-headlines (accessed on 1 October 2021).

Acknowledgments: We would like to acknowledge the Scientific Research Deanship at the University of Ha'il, Saudi Arabia, for funding this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, Y.; Conroy, N.J.; Rubin, V.L. Misleading Online Content: Recognizing Clickbait as "False News". In Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection, Washington, DC, USA, 13 November 2015; pp. 15–19.
2. Biyani, P.; Tsioutsiouliklis, K.; Blackmer, J. "8 Amazing Secrets for Getting More Clicks": Detecting Clickbaits in News Streams Using Article Informality. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
3. Potthast, M.; Kopsel, S.; Stein, B.; Hagen, M. Clickbait Detection. In *Advances in Information Retrieval. 38th European Conference on IR Research (ECIR 16)*; Springer: Cham, Switzerland, 2016; pp. 810–817.
4. Chakraborty, A.; Paranjape, B.; Kakarla, S.; Ganguly, N. Stop clickbait: Detecting and preventing clickbaits in online news media. In Proceedings of the 2016 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), Davis, CA, USA, 18–21 August 2016; pp. 9–16.
5. Khater, S.R.; Al-sahlee, O.H.; Daoud, D.M.; El-Seoud, M. Clickbait detection. In Proceedings of the 7th International Conference on Software and Information Engineering, Cairo, Egypt, 4–6 May 2018; pp. 111–115.
6. López-Sánchez, D.; Herrero, J.R.; Arrieta, A.G.; Corchado, J.M. Hybridizing metric learning and case-based reasoning for adaptable clickbait detection. *Appl. Intell.* **2017**, *48*, 2967–2982. [CrossRef]
7. Agrawal, A. Clickbait Detection Using Deep Learning. In Proceedings of the 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India, 14–16 October 2016; pp. 268–272.
8. Kaur, S.; Kumar, P.; Kumaraguru, P. Detecting clickbaits using two-phase hybrid CNN-LSTM biterm model. *Expert Syst. Appl.* **2020**, *151*, 113350. [CrossRef]
9. Zheng, H.T.; Chen, J.Y.; Yao, X.; Sangaiah, A.K.; Jiang, Y.; Zhao, C.Z. Clickbait convolutional neural network. *Symmetry* **2018**, *10*, 138. [CrossRef]
10. Naeem, B.; Khan, A.; Beg, M.O.; Mujtaba, H. A deep learning framework for clickbait detection on social area network using natural language cues. *J. Comput. Soc. Sci.* **2020**, *3*, 231–243. [CrossRef]
11. Bazaco, A.; Redondo, M.; Sánchez-García, P. Clickbait as a strategy of viral journalism: Conceptualisation and methodst. *Rev. Lat. de Comun. Soc.* **2019**, *74*, 94–115.
12. Jain, M.; Mowar, P.; Goel, R.; Vishwakarma, D.K. Clickbait in Social Media: Detection and Analysis of the Bait. In Proceedings of the 2021 55th Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 24–26 March 2021; pp. 1–6.
13. Chawda, S.; Patil, A.; Singh, A.; Save, A. A Novel Approach for Clickbait Detection. In Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 23–25 April 2019; pp. 1318–1321.
14. Aburas, A.A.; Gumah, M.E. Arabic Handwriting Recognition: Challenges and Solutions. In Proceedings of the 2008 International Symposium on Information Technology, Kuala Lumpur, Malaysia, 26–28 August 2008.
15. Al-Nuzaili, Q.A.; Hashim, S.Z.M.; Saeed, F.; Khalil, M.S.; Mohamad, D.B. Pixel distribution-based features for offline Arabic handwritten word recognition. *Int. J. Comput. Vis. Robot.* **2017**, *7*, 99–122. [CrossRef]
16. Al-Sarem, M.; Cherif, W.; Wahab, A.A.; Emara, A.H.; Kissi, M. Combination of stylo-based features and frequency-based features for identifying the author of short Arabic text. In Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications, Rabat, Morocco, 24–25 October 2018; pp. 1–6.
17. Cao, X.; Le, T. Machine learning based detection of clickbait posts in social media. *arXiv* **2017**, arXiv:1710.01977.
18. Zhou, Y. Clickbait detection in tweets using self-attentive network. *arXiv* **2017**, arXiv:1710.05364.
19. Dong, M.; Yao, L.; Wang, X.; Benatallah, B.; Huang, C. Similarity-aware deep attentive model for clickbait detection. In *Advances in Knowledge Discovery and Data Mining*; Springer: Cham, Switzerland, 2009; Volume 11440, pp. 56–69.
20. Sahoo, S.R.; Gupta, B.B. Multiple features based approach for automatic fake news detection on social networks using deep learning. *Appl. Soft Comput.* **2021**, *100*, 106983. [CrossRef]
21. Lee, D.-H. Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. In *Workshop on Challenges in Representation Learning*; ICML: Atlanta, GA, USA, 2013; Volume 3, p. 2.
22. Al-Sarem, M.; Saeed, F.; Alsaeedi, A.; Boulila, W.; Al-Hadhrami, T. Ensemble methods for instance-based Arabic language authorship attribution. *IEEE Access* **2020**, *8*, 17331–17345. [CrossRef]
23. Bahassine, S.; Madani, A.; Al-Sarem, M.; Kissi, M. Feature selection using an improved Chi-square for Arabic text classification. *J. King Saud Univ.-Comput. Inf. Sci.* **2020**, *32*, 225–231. [CrossRef]
24. Elssied, N.O.F.; Ibrahim, O.; Osman, A.H. A novel feature selection based on one-way ANOVA F-test for e-mail spam classification. *Res. J. Appl. Sci. Eng. Technol.* **2014**, *7*, 625–638. [CrossRef]