Proceedings of the International Conference on Artificial Intelligence and Smart Systems (ICAIS-2021)
IEEE Xplore Part Number: CFP21OAB-ART; ISBN: 978-1-7281-9537-7

# Machine Learning based Click Bait Detection System

Ritu Agarwal
*Department of Information Technology*
*Delhi Technological University*
New Delhi, India
ritu.jeea@gmail.com

Dhruv Jain
*Department of Information Technology*
*Delhi Technological University*
New Delhi, India
dhruvjain_2k17it47@dtu.ac.in

Gagan Sharma
*Department of Information Technology*
*Delhi Technological University*
New Delhi, India
gagansharma_2k17it48@dtu.ac.in

Janardan Tiwari
Department of Information Technology
Delhi Technological University
New Delhi, India
janardantiwari_2k17it57@dtu.ac.in

2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS) | 978-1-7281-9537-7/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICAIS50930.2021.9395898

**Abstract - The state of the media and related platforms that disseminate information is currently undergoing drastic transformation. Click bait are articles or other form of information that exaggerate the content on the targeted pages. When going through such content people often end up jumping on wrong conclusions. In this research we seek to construct an efficacious computational system primarily using ensemble learning. Several Machine learning algorithms such as Logistic regression, Stochastic Gradient Descent and Random Forest have been used.**

**Keywords - Logistic regression, Stochastic Descent, Random Forest, TF IDF vectorizer, K fold Gradient cross validation, confusion matrix and ELI 5.**

## I. INTRODUCTION

This paper revolves around the detection of click baits using a combination of machine learning algorithms. With the growing usage of internet and online social media various companies try to bombard readers with advertisements, thus making revenue by the same. The content presented by such advertisement agencies is far away from being authentic and is highly misleading. Click baits are used to increase the click throughs of a website for generating revenue. When going through such content people often end up jumping on the wrong conclusions [3]. These developments have been a reason to cause enormous concern among many prominent authors and writers, because this can potentially shut down social media channels and it also defies ethical journalism. Click baits are basically those content on the internet which attract the attention of the viewer. Click baits may be graphical or textual in format, however, in this paper we'll be limiting ourselves with textual click baits and their detection [8]. Some popularly known websites that host such content are Upworthy and BuzzFeed. Besides these two there are several other hosting with similar kinds of content. Almost all click baits reside on two facts. First, they will always have incomplete or misleading information used to guarantee an emotional response which is often not served once the readers actually goes through the content. Second, they'll always try to exploit the curiosity gap in humans. Below is the list of some commonly used click baits

1. I left my kids alone at home and you will be shocked to know what happened!
2. 27 memes any procrastinator would completely agree with!
3. What these trained military dogs can do will blow your mind!
4. This Man jumped from a 20-storey building and yet he made it to the ground alive!

In this paper we have done an in-depth analysis of various machine learning techniques and try to identify click baits in most effective ways.

## II. RELATED WORK

A lot of research work employing various machine learning as well as deep learning algorithms for click bait detection / prediction is available on the internet.

TABLE I. RELATED PAPERS PUBLISHED

| Publication | Teaser Type | Annotation Scale | Article Archival | Size |
|---|---|---|---|---|
| Potthast et al. [2016] | Tweet | Binary | Yes | 2992 |
| Chakraborty et al. [2016] | Headline | Binary | No | 15000 |
| Rony et al. [2017] | Headline | Binary | No | 32000 |
| Agarwal [2016] | Headline | Binary | No | 2388 |
| Biyani et al. [2016] | Headline | Binary | No | 4073 |
| Potthast et al. [2018] | Tweet | Graded | Yes | 38517 |

As of now several machine learning techniques such as ULMFIT classifiers, Stephan Dreiseitl, Lucila Ohno-Machado [11] studies using logistic regression,A. Agrawal [2] uses deep learning, Potthast M., Köpsel S., Stein B., Hagen M [1] uses random forest and H.-T. Zheng, J.-Y. Chen, X. Yao, A. Sangaiah, Y. Jiang, and C.-Z. Zhao [4] uses Convolutional neural network, F. Kabir, S. Siddique, M. R. A. Kotwal and M. N. Huda [10] uses Gradient Descent, Recurrent neural networks and other Natural Language Processing methods [13] have also been used to identify click baits. Some papers even have generated click baits using deep learning [3]. Click bait detection is limited to textual data in our case but we also came across some other models which works well on graphical data as well [15]. Some papers even extended research to different languages like Turkish [5] and Bangla [10].

## III. DATA USED

The data used for this research consists of 24781 tuples and 4 attributes namely index, news id, title and text. We aim to use news title and new text to assign it one of the three classes – click bait, news and other. The testing data consists of 5647 tuples and 3 attributes. Fig 1 provide some insights.

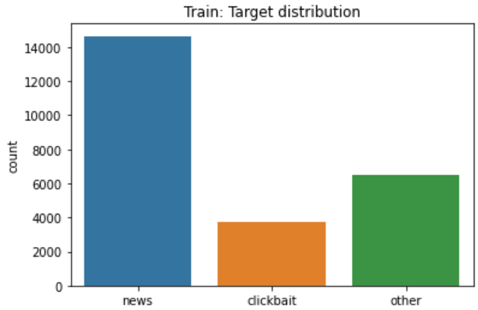


**Fig 1** shows the overall diversity of the training data. In this research we have used supervised learning and count of each label is indicated in that graphic.

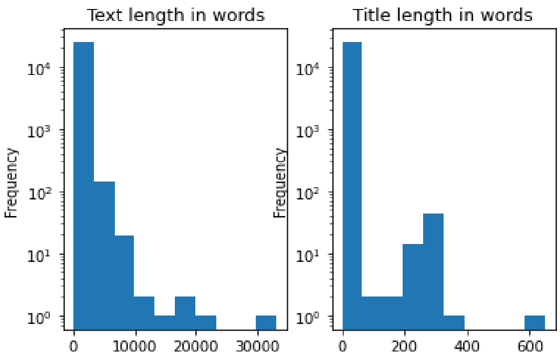


**Fig 2** depicts the frequency of the words along with their length present in the given dataset

## IV. PROPOSED WORK

In this research we have used Ensemble learning comprising of three machine learning algorithms namely logistic regression, stochastic gradient descent classifier and random forest. The following flowchart depicts the workflow of the research.

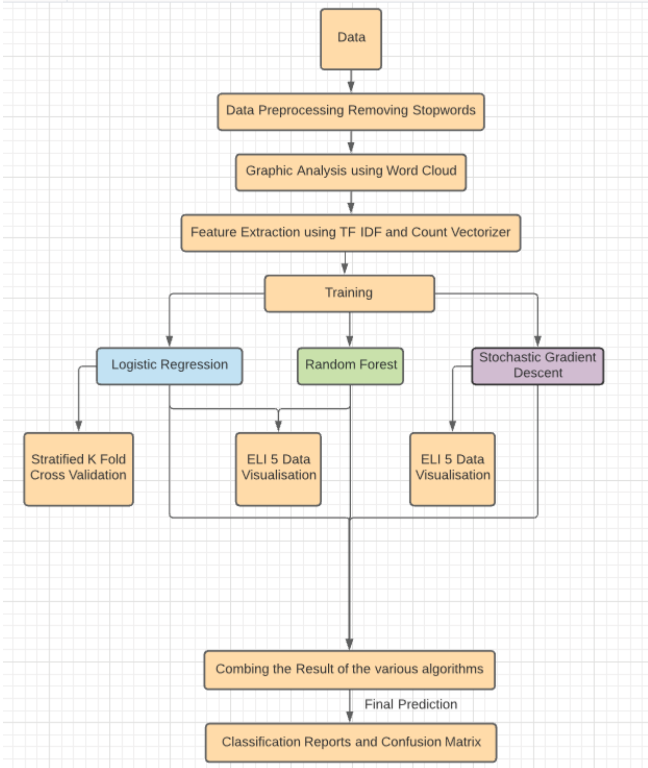


**Fig 3** shows the flowchart of the research

The various algorithms used in this research and their individual results along with the result of ensemble of all of them are depicted in the following table.

TABLE II. COMPARSION OF VARIOUS ALGORITHMS USED

| S. No. | Method | Accuracy | Precision | Recall |
|---|---|---|---|---|
| 1. | Logistic regression | 83.92 | 0.83 | 0.84 |
| 2. | Stochastic Gradient Descent | 86.54 | 0.85 | 0.87 |
| 3. | Random Forest | 84.88 | 0.85 | 0.85 |
| 4. | Ensemble | 86.68 | 0.86 | 0.87 |

## V. FEATURE ENGINEERING

Feature Engineering is a process that involves usage of knowledge inferred from data. The knowledge obtained is used to synthesize explanatory parameters as well as features which in turn help us build a predictive model specifically a classifier in this case. We've several motivations when talking about feature engineering.

When we identify essential parameters the overall predictive ability of the model improves significantly. Another fundamental goal of producing less complex and computationally less expensive models with impressive predictive ability is also achieved. As in the case of any other machine learning project, we always first get to deal with raw data which is a complete mess and haywire. The whole process commences with cleaning of this data which involves a lot of steps like removing missing values and also changing the data kind / types of the provided input. Further we go ahead removing some outliers or some useless features that hold little importance to our model. Once the data is cleaned, we can explore the data well and even go ahead create new features that better help in analysing the given problem statement, facilitating a smooth learning process for our machine learning model and improving the performance / accuracy.

TABLE III. DEALING WITH VARIOUS ISSUES

| Issue | Teaser Type |
|---|---|
| Missing Value | Imputed in data cleaning |
| Does not belong to same dimension | Normalisation / Standardisation |
| Information redundancy | Filtered out in feature selection |

As we discussed the feature engineering and the need to identify unique features which can improvise predictive ability of a model we would now try to get into the context of our paper and try to identify features that one can look for when dealing with identification of click bait in a news article. We are listing a few of them as follows

1. Count of stop words in the concerned content
2. Count of words in the concerned content
3. Count of contractions in the concerned content
4. Ratio of stop words to total number of words in the concerned content
5. Whether headline is a question or not

It is to be carefully noted that we haven't used all of them in our analyses; details regarding our features will be described later. That completed one of our tasks in feature engineering that is to identify key features. However, we also stand a possibility to create new attributes / features / functions. Some support is being provided by NLTK in this case for example the function process text is useful with processing of raw content and what it essentially does is that changes the case of the content to lowercase, removes extra white spaces and also omits punctuation marks. Furthermore, it replaces numerical values with a suitable string substitute.

There are two more features to look out for when dealing with click bait detection. It is a general observation that click bait texts generally are more informal in nature. Issue Solution Missing Values Imputed in data cleaning Does not belong to the same dimension Normalization / Standardization Information redundancy Filtered out in feature selection This might not hold in every case but does hold good in the majority of the cases where the sole purpose is to get the attention of the targeted user.

Another common observation is that click bait texts generally carry some specific punctuation marks. These can question marks in headings or exclamation marks which can be either in heading or repeated frequently in the body. Measuring frequency of such punctuation marks is a trivial task and can be done by looping through the test content string. Another feature to look out for is the part of speech or the overall structure of the sentence. A common observation is that non click bait headings / titles tend to contain more nouns than the click bait content. Similarly, we can also look out for the nature of adjectives and verbs used in the target content. Above tasks can be easily carried out by NLTK packages. The word cloud so obtained has been presented in figure 4.


**Fig 4** shows the word cloud where the size of the word is proportional to the frequency of its occurrence.

## VI. DATA PREPROCESSING

### A. Removing Stop words

Before we discuss their removal first let's understand stop words. Stop words are some commonly occurring words in any given language and these words don't help much in determining the nature of text or text analyses. Whenever dealing with machine learning models it becomes important to get rid of them. Generally, the most common words used in a text are "a", "I", "while", "for", "where", "when", "to", "at" etc. Consider the following example "There is a pencil in the pencil box". Now, the words "is", "a", "in", and "the" add no meaning to the sentence. Whereas words like "there", "pencil", and "pencil box" are the unique keywords and on parsing them we get useful results. NLTK generally provides us with a bunch of stop words from 16 different languages and we did use the same in the research. Just to be more accurate we even added more stop words from our side which would be evident later.

### B. TF IDF Vectorizer

When dealing with data then one must understand that machine learning algorithms can work with works from natural language. Therefore, we need to convert these words into numbers or vectors for further analysis.

Term Frequency - It is a physical quantity which can be calculated for any given document [5]. The formula for calculating Term Frequency is the number of times a word occurs in the given document.

$$tf_{i,j} = \frac{n_{i,j}}{\Sigma_k n_{i,j}} \quad (1)$$

$n_{i,j}$ : frequency of word $i$ in document $j$

$\Sigma_k n_{i,j}$: number of words in document $j$

$tf_{i,j}$ : term frequency

Inverse Data Frequency (IDF) - It is a physical quantity which can be calculated for any given document. The formula for calculating Term Frequency is the logarithm of the number of documents that contain that word divided by the total number of available documents [5].

$$idf(w) = \log\left(\frac{N}{df_t}\right) \quad (2)$$

$N$ : total number of documents

$df_t$ : document frequency

$idf(w)$ : inverse document frequency

Finally, the TF-IDF is simply the product of the above two

$$w_{i,j} = tf_{i,j^*}\ idf(w) \qquad (3)$$

$idf(w)$ : inverse document frequency
$tf_{i,j}$ : term frequency
$w_{i,j}$ : TF IDF formula

### C. Count Vectorizer

In this research too we also practiced feature engineering and used count vectorizer. Count vectorizer takes a collection of text documents as input and produces a vector of terms containing the count of each kind of token. It basically enables pre-processing of data by providing vectors instead of text words and that basically helps to make flexible representations of the vital features.

## VII. RESEARCH METHODOLOGY

### A. Ensemble Learning in Click bait detection

Ensemble Learning is a technique used to solve computational intelligence problems. Here we use several other machine learning models to solve a given problem. Each model individually generates results for that problem and then the results of these algorithms are combined which helps to produce better results [9]. This process is preferred because it reduces the probability or likelihood of an unfortunate selection of machine learning algorithm for a given dataset and a given problem statement. Here we would be using bagging technique and would assign equal priorities to the result produced by our different machine learning models. We have used Hard Voting where we give equal weightage to the results produced by the individual models and the result which is voted the most is chosen [9]. Now let's explore two more important process mentioned as follow:

1. Bootstrap Aggregation - In this method we first create some samples from the training data with replacement. These samples are completely stochastic in nature. This process is repeated several times and ensures that we don't overfit the model on the training dataset thereby minimizing variance error. Of all the samples generated voting is done for the output prediction and the most voted output among all the various models so obtained becomes our final output [9].
2. Boosting – It is another technique available for ensemble learning. The idea is to keep changing weights of the various features. Boosting essentially involves an iterative approach where if we labelled an observation incorrectly then in the next iteration, we will increase its weightage and vice versa. Boosting decreases the error due to bias and also performs better than bagging, but suffers from high variance error or overfitting. Parameter Tuning is thus a crucial step in boosting [9].

Figure 5 presents the confusion matrix and the accuracy obtained while using ensemble learning.

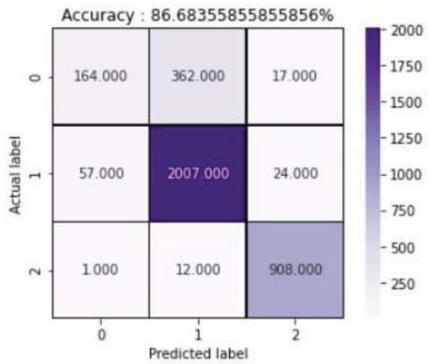


**Fig 5** shows the confusion matrix of the Ensemble learning used in the research

### B. Logistic Regression in Click bait detection

Logistic function is basically used to obtain the probability of a certain class. We can also use it to get the probability of events like whether something will pass or fail, someone will win or lose, something is dead or alive or maybe healthy or sick. It is also known as the Logit Model. Logistic regression derives its root from the Logit Model or logistic function. It's a statistical model which has one or many independent variables of various kinds but essentially a dependent binary variable although more complex alterations do exist [11]. We can also have regression analysis here; logistic regression estimates the parameters of a logit model. This task is similar to a form of binary regression. Mathematically, a binary logistic model will always have a single dependent variable with two possible values and we generally label them as "True" and "False" or sometimes as 0 or 1. When dealing with the logistic model, the logarithm of the odds for the value labelled as true / 1 comprises a linear combination of one or more variables which are independent and are also called predictors and these variables can each be a binary kind variable or a continuous kind variable. Figure 6 presents the confusion matrix and the accuracy obtained.

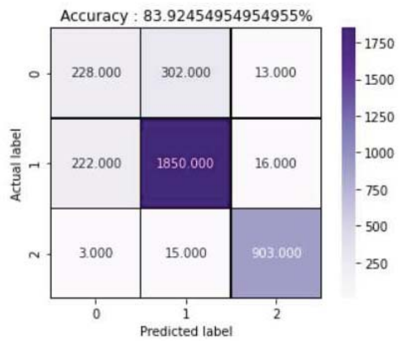


**Fig 6** shows the confusion matrix for Logistic regression

### C. Stochastic Gradient Descent in click bait detection

Stochastic Gradient Descent (SGD) is a machine learning algorithm commonly used to improve performance of the standard gradient descent algorithm [10]. This algorithm is commonly used in other machine learning algorithms and it also underpins neural networks. Gradient means slope or the slant of a surface at a given point. So gradient descent as the name suggests means descending slope to reach the lowest point on the surface. Stochastic Gradient descent is similar to Gradient descent except the fact that it randomly chooses a point and then starts executing in an iterative fashion trying to find the lowest point. We decided to use SGD because there are a few drawbacks of the gradient descent algorithm and that would become evident once we take an example to show the number of computations involved in each iteration of the gradient descent algorithm.

Let us assume that we have 5,000 data points and 10 features. Now Gradient descent algorithm demands computation of derivative of this function with respect to each and every feature, so in total we end up doing 5000 * 10 = 50,000 computations and that is just for 1 iteration. Generally, we end up with 1000 iterations and that will be an absolutely safe and realistic assumption so in total we have 50,000 * 1000 = 50000000 computations to complete the algorithm. The number of computations involved are too large and thus gradient descent is computationally expensive whereas SGD reduces all these computations by a large amount. Figure 7 presents the confusion matrix and the accuracy obtained.

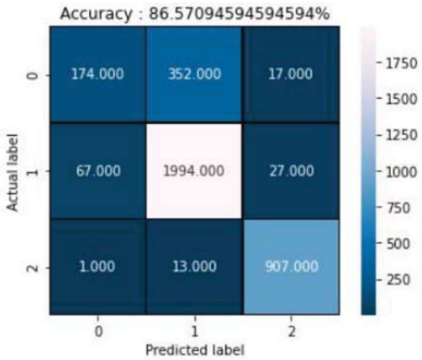


**Fig 7** shows the confusion matrix for stochastic gradient descent

### *D. Random Forest in click bait detection*

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. A Decision tree is a rooted tree used as a predictive model. We express a decision tree by partitioning the feature space recursively into many subspaces and each of these subspaces form a basis of prediction. In a decision tree a set of features are used in a hierarchical fashion in such a way that after each split the entropy of the system minimizes. We can have a split criterion for the internal nodes. Finally, each leaf is assigned to one class or its probability. Decision tree suffers from Overfitting as even small variation in the dataset lead to large errors and to deal with the same methods like pruning of trees is adopted. A Random forest basically constitutes an ensemble of a large number of these decision trees [1]. Every Decision tree in random forest provides an output to the given input data utilizing all the available features and finally a voting is done and class occurring the most frequently becomes the final output of the random forest. In any random forest a large number of trees operate and it is ensured that these trues and uncorrelated models thereby ensuring that they don't get influenced by each other and naturally such a combination is expected to produce better results than any individual decision tree. Individual trees in our random forest are protected from other decision trees because they are not corrected or their coefficient of correlation is very low and consequently each tree gets protected from the errors of other trees. The prerequisites for efficacious use of random forests include:

1. Features in the model must be well researched and should actually influence the predictive ability. This will prevent the model from randomly making any guesses thereby providing better results.
2. It is essential that the individual decision trees in random forests are not correlated or have a very low correlation amongst each other.

Figure 8 presents the confusion matrix and the accuracy obtained while using random forest.

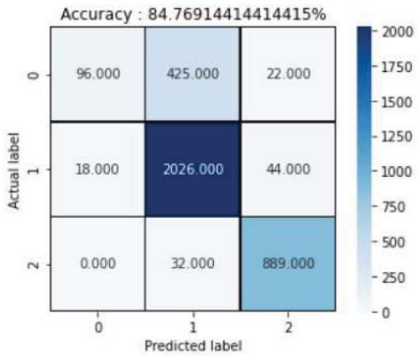


**Fig 8** shows the confusion matrix for random forest

### *E. Explain like I am 5 (ELI 5)*

It is a python library used for visualising and debugging machine learning models. ELI 5 has excellent build in support for numerous machines learning frameworks and also supports unified API. ELI 5 provides a lucid way to explain black box models. In this research we used the Eli 5 library to visualise how much a word contributes in making a text as a click bait. A positive value indicates that the word contributes in making a text click bait while a negative value on the other hand states indicates that the given word drives the text towards not being a click bait. Figure 9 shows the obtained ELI 5 visualisation.

| **y=clickbait** top features | | **y=news** top features | | **y=other** top features | |
|---|---|---|---|---|---|
| **Weight?** | **Feature** | **Weight?** | **Feature** | **Weight?** | **Feature** |
| +4.716 | 2017 | +4.507 | 2017 | +5.736 | <BIAS> |
| +3.532 | buzzfeed | +4.296 | breitbart | +4.575 | hn |
| +3.467 | buzzfeed inc | +3.733 | trump | +4.156 | etc |
| +2.726 | 2016 | +3.371 | cnn | +3.779 | ask hn |
| +2.651 | inc | +3.014 | buzzfeed | +3.375 | ask |
| +2.638 | 2017 buzzfeed | +2.805 | day | +3.037 | seems |
| +2.585 | video | +2.726 | shows | +2.676 | hn |
| +2.519 | things | +2.611 | inc | +2.496 | anything |
| +2.456 | quiz | +2.586 | linkedin | +2.478 | 2012 |
| +2.431 | know | +2.492 | tuesday | +2.384 | startup |
| *… 9928 more positive …* | | *… 11164 more positive …* | | *… 5084 more positive …* | |
| *… 10058 more negative …* | | *… 8822 more negative …* | | *… 14902 more negative …* | |
| -1.954 | represents | -2.076 | tools | -4.925 | buzzfeed inc |
| -1.976 | audio | -2.294 | drafted | -5.262 | inc |
| -1.980 | seventh | -2.375 | ask hn | -5.325 | cnn |
| -3.168 | breitbart | -2.472 | etc | -6.546 | buzzfeed |
| -4.164 | <BIAS> | -2.895 | hn | -9.223 | 2017 |

**Fig 9** shows the weight of each word in determining whether an article is a click bait or not.

### *F. Model Validation – Stratified K Fold*

In order to anticipate the skill of a machine learning model on unlabelled data, cross validation is fundamentally utilized in applied machine learning. In K fold validation we have an essential parameter k which denotes the number of subgroups the original data can be split into. The reason we apply cross validation is because it tests the ability of our machine learning model to predict when data is completely unseen to it. Subsequently, cross validation allows us to anticipate the performance of our machine learning model when used on test data for actual predictions by ensuring the data that we use to validate the model isn't the same as training data [7]. Stratified K Fold validation is similar to K fold cross validation except the fact that Stratified K Fold validation doesn't immediately split data but first shuffles our data and then splits the data. This happens every time before actually splitting the data. So, among the various segments of the data so obtained 1 segment is used for testing while other segments are used for training in this procedure. The value of parameter K in Stratified K fold cross validation plays a very essential role. If we don't choose a good value of k then it might lead to complete misrepresentation of our data and accuracy will be compromised by humongous amounts. Model can end up suffering from high bias where we overestimate a particular skill or high variance where the model is too much dependent on data and has also absorbed noise. A little change in data in such cases produces a significant drop in the overall accuracy of the model. Here are some common techniques when choosing the value of K.

1. The value for k is chosen such that each group of data samples, be it training or testing, is large enough to be statistically representative of the dataset.
2. We can make k equal to 10. Most of the experiments prefer this value and it's a promising value resulting in a low bias model with modest variance.
3. We can make k equal to n. In doing so we provide every segment of our data to act as a training data as well as treating data which might help in improving the overall accuracy of the model.

## VIII. Result And Conclusion

In this paper we used Ensemble learning to detect click baits. This paper provides an intuitive method of the combining the results by giving preference to the model with highest accuracy in case of ambiguity. First, we resorted to hard voting but in order to avoid clashes we have more preference to Stochastic Gradient Descent classifier due to its higher accuracy owing to the fact that it implements mini batch technique to find the optimal parameters. Feature extraction was done using TF IDF vectorizer as well as Count vectorizer. Various machine learning algorithms were used and a combine function was defined to create ensemble using hard voting mechanism. Model Validation has been done with Stratified K Fold cross validation. This research paper focuses on ensemble learning to ensure that algorithms are faster and can be applied in various social media platforms as well as news articles. The final accuracy so obtained was 86.68%.

## IX. Future Work

There were couple of difficulties and future extensions that work confronted and should be addressed.

- A leap from Ensemble learning to deep learning and inclusion of neural networks is also a possible idea which might improve accuracy [2].
- Besides training the better pre-processing of the dataset and identification of some other vital elements might even enhance the overall performance of the machine learning model.
- We also believe better validation methods can also bring about major differences in the accuracy of the system. In this research we adopted K fold Stratified validation but many other validations can be tried.
- A better method of encoding the dataset might also improvise the existing models.
- The research was using news articles for training the model however we strongly believe that other sources of click bait like tweets, Facebook posts and answers floating A2A platforms like Quora might serve as better sources thereby enhancing the overall performance of the model.
- The research was also limited to textual click baits however graphics which majorly includes images and videos are also common.
- We can also extend the current research to various other language besides English [14].
- Content can be classified into more precise subclasses. In our research we resorted to three classes i.e., clickbait, news and other. However, the "other" section can be further extended [12].
- Advanced computer vision techniques might also further open more domains of research in this particular problem statement of detecting click baits [6].

## X. References

[1] Potthast M., Köpsel S., Stein B., Hagen M. (2016) Click bait Detection. In: Ferro N. et al. (eds) Advances in Information Retrieval. ECIR 2016. Lecture Notes in Computer Science, vol 9626. Springer, Cham. https://doi.org/10.1007/978-3-319-30671-1_72

[2] A. Agrawal, "Click bait detection using deep learning," 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), Dehradun, 2016, pp. 268-272, doi: 10.1109/NGCT.2016.7877426.

[3] K. Shu, S. Wang, T. Le, D. Lee and H. Liu, "Deep Headline Generation for Click bait Detection," 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 2018, pp. 467-476, doi: 10.1109/ICDM.2018.00062.

[4] H.-T. Zheng, J.-Y. Chen, X. Yao, A. Sangaiah, Y. Jiang, and C.-Z. Zhao, "Click bait Convolutional Neural Network," *Symmetry*, vol. 10, no. 5, p. 138, May 2018.

[5] A. Geçkil, A. A. Müngen, E. Gündogan and M. Kaya, "A Click bait Detection Method on News Sites," 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Barcelona, 2018, pp. 932-937, doi: 10.1109/ASONAM.2018.8508452.

[6] S. Chawda, A. Patil, A. Singh and A. Save, "A Novel Approach for Click bait Detection," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 1318-1321, doi: 10.1109/ICOEI.2019.8862781.

[7] A. Chakraborty, B. Paranjape, S. Kakarla and N. Ganguly, "Stop Click bait: Detecting and preventing click baits in online news media," 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), San Francisco, CA, 2016, pp. 9-16, doi: 10.1109/ASONAM.2016.7752207.

[8] Chen, Y. and Rubin V. L. (2017). Perceptions of Click bait: A Q-Methodology Approach. In the Proceedings of the 45th Annual Conference of The Canadian Association for Information

Science/L'Association canadienne des sciences de l'information (CAIS/ACSI2017), Ryerson University, Toronto, May 31 - June 2, 2017

[9] Gang Wang, Jianshan Sun, Jian Ma, Kaiquan Xu, Jibao Gu, Sentiment classification: The contribution of ensemble learning, Decision Support Systems, Volume 57, 2014, Pages 77-93, ISSN 0167-9236, https://doi.org/10.1016/j.dss.2013.08.002.

[10] F. Kabir, S. Siddique, M. R. A. Kotwal and M. N. Huda, "Bangla text document categorization using Stochastic Gradient Descent (SGD) classifier," 2015 International Conference on Cognitive Computing and Information Processing(CCIP), Noida, 2015, pp. 1-4, doi: 10.1109/CCIP.2015.7100687.

[11] Stephan Dreiseitl, Lucila Ohno-Machado, Logistic regression and artificial neural network classification models: a methodology review, Journal of Biomedical Informatics, Volume 35, Issues 5–6, 2002, Pages 352-359, ISSN 1532-0464, https://doi.org/10.1016/S1532-0464(03)00034-0

[12] Y. Sun, K. Tang, L. L. Minku, S. Wang and X. Yao, "Online Ensemble Learning of Data Streams with Gradually Evolved Classes," in IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 6, pp. 1532-1545, 1 June 2016, doi: 10.1109/TKDE.2016.2526675.

[13] S. Manjesh, T. Kanakagiri, P. Vaishak, V. Chettiar and G. Shobha, "Clickbait Pattern Detection and Classification of News Headlines Using Natural Language Processing," 2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), Bangalore, 2017, pp. 1-5, doi: 10.1109/CSITSS.2017.8447715.

[14] P. Klairith and S. Tanachutiwat, "Thai Clickbait Detection Algorithms Using Natural Language Processing with Machine Learning Techniques," 2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST), Phuket, 2018, pp. 1-4, doi: 10.1109/ICEAST.2018.8434447.

[15] J. Fu, L. Liang, X. Zhou and J. Zheng, "A Convolutional Neural Network for Clickbait Detection," 2017 4th International Conference on Information Science and Control Engineering (ICISCE), Changsha, 2017, pp. 6-10, doi: 10.1109/ICISCE.2017.11.