Journal of Innovative Engineering and Research (JIER)
Vol. - 7, Issue -2, October, 2024, pp. 9-13 (5 pages)

Constructing a User-Cantered Fake News Detection Model by Using Enhance Stacking Ensemble Classification Algorithms in Machine Learning Techniques

Mr. Raut Rahul Ganpat and Dr. Sonawane Vijay Ramnath

Department of Computer Science and Engineering, Dr. A. P. J. Abdul Kalam University, Indore 452010, India
Corresponding Author Email: braut_rahul@yahoo.co.in

Abstract: In today's information-dense society, identifying false news remains an omnipresent challenge, impacting individuals and societies globally. The ever-evolving nature of misinformation and its dissemination methods demands a combined approach to detect and counter its proliferation effectively. This thesis introduces the Unified Fake News Detection System (UFNDS) - a comprehensive and integrative framework crafted by blending three pivotal phases: the Algorithm for Enhance Stacking Ensemble Classification (ES-ECA), the Optimized Machine and Deep Learning (OE-MDL).

Initially, employs. In dataset is divided into separate statements using preprocessing techniques, and n-grams are produced as features. metric, which measures feature's significance to the document, is then used to extract the features. By combining base classifiers, such as an improved version of the, an improved Naive Bayes, and an improved version of the k-Nearest Neighbors algorithm, the suggested algorithm employs an enhanced stacking ensemble approach. Next, a Random Forest method enhanced by the AdaBoostM1 technique is used to produce a meta-classifier. The stacking ensemble classifier's performance is enhanced by the Random SubSpace technique.

1. INTRODUCTION

Algorithm for Enhanced Stacking Ensemble Classification (ES-ECA). This chapter provides an exhaustive exploration of the ES-ECA phase, elucidating the preprocessing techniques, feature extraction methods, and the intricacies of the improved stacking ensemble methodology. In this phase, it leverages the power of machine learning to combat the proliferation of fake news, demonstrating how it plays a pivotal role within the overarching UFNDS framework.

1.1 Introduction to ES-ECA

The Algorithm for Enhanced is sophisticated framework developed to convey the critical issue of identification of false news. In today's digital ecosystem, fake news—defined as purposefully inaccurate or misleading material presented as true news—has become a major issue. ES-ECA offers a comprehensive solution by harnessing the potential of ensemble learning and algorithmic enhancements to spot false information accurately.

1.2 The Requirement of Sophisticated Fake News Identification

Fake news's rise has had a significant impact on society, politics, and the economy. Traditional techniques for identifying false news, which frequently depend on linguistic clues and stylistic elements, are neither very accurate nor broadly applicable. To get over these restrictions and offer a more reliable method of spotting fake news, ES-ECA was developed.

1.3 Application Areas

ES-ECA finds application in various domains where identifying false news is crucial,:

- Platforms for
- News agencies websites
- Online discussion forums
- Political campaigns
- Academic research

1.4 Enhanced stacking ensemble classification algorithm (ES-ECA)

This segment introduces an (ES-ECA) to identifying false news. It utilizes the LIAR dataset, comprising news articles, to differentiate between "fake" and "real" news articles. The algorithm employs a two-stage approach involving preprocessing the data and extracting features, which enables the conversion of the unprocessed data into a machine-learning-friendly format.

In the preprocessing phase, methods like lowercase conversion, tokenization, elimination of stop words, and stemming are used to lower the quantity of distinct remove common words and tokens lacking informative value. As for feature extraction, It comprises constructing "n-grams" out of the assertions. N-grams represent sequences of n-words found within the written word and serve as valuable features within a machine-learning model, aiding in the way the statements are categorized.

The algorithm calculates the (TF-IDF) for every n-gram within the dataset. TF-IDF is a metric that assesses the significance of a specific phrase (here, an n-gram) about a record (here, a statement) within a larger collection of documents (Here, the dataset from LIAR).

For the purpose of testing and training, the algorithm splits the dataset into two sets and saves them as different files. The technique makes use of a stacking ensemble model to solve the classification challenge. It involves creating a meta-classifier using the Random Forest approach, which uses 500 trees, and then using the AdaBoostM1 algorithm to improve its performance. Base classifiers include a modified version of an improved version of the Complement Naive Bayes algorithm, and an improved k-Nearest Neighbors (k-NN) with a neighbor count of five. Additionally, a bagging algorithm is applied to the k-NN method in order to reduce the possibility of overfitting.

2. PREPROCESSING

Preprocessing, involves cleaning up raw data and putting it in a format that will allow for further analysis. Preprocessing in Algorithm 3.1 refers to a set of steps intended to smooth and transform the original news article dataset before moving on to feature extraction and model training. The subsequent steps are included in these preprocessing operations:

1. **Taking the dataset lowercase:** At this point, the dataset's text is converted to lowercase throughout. Here operation ensures uniform treatment of terms spelled similarly, but differing capitalization, such as "Apple" and "apple," which would be considered identical words following the conversion to lowercase.
2. **Making the dataset tokenized:** Tokenization separates the text into discrete words, also known as tokens. Using spaces or other predefined delimiters, the dataset is divided into distinct words in order to accomplish this. To tokenize a statement like "The food was amazing, but the service was terrible," the word list that results would be ["the", "food", "was", "amazing", "," "but", "the", "service", "was", "terrible"] would result.
3. **Eliminating stop phrases:** Terms with little or no meaning in a certain context are frequently employed as stop words. Words like "the," "and," "a," "an," "in," "of," and so on are instances of stop words. These terms are commonly found in written works and do not make a substantial difference between documents. For this reason, removing stop words can help reduce noise in the model and lessen irrelevant information.
4. **Stemming the dataset:** The stem is the procedure of simplifying phrases to their core or root form. It entails the removal of prefixes and suffixes from words to extract their foundational form. For instance, the words "jumping," "jumps," and "jumped" would all be boiled down to the common stem "jump" through stemming. This technique aids in decreasing data dimensionality and enhancing model accuracy by diminishing the count of unique words. It occurs because numerous terms used in a language share a common root structure and can, therefore, be regarded as identical words.

3. FEATURE EXTRACTION

A technique called identify and extract a dataset's most important features or qualities. The main goal of feature extraction is to transform unprocessed data into a set of useful features that can be used to build prediction models or carry out more analysis.

High-dimensionality refers to the fact that the data used in machine learning and data analysis typically contains a large number of variables or characteristics. complexity, might difficult build accurate features might be redundant or insignificant. This problem is resolved by feature extraction, which lowers the dimensionality of the data and finds the most important features.

N-grams

A combination of statistical and computational techniques is typically used in feature extraction. One popular method for identifying word sequences of varying lengths in text data is to use n-gram generation. An n-gram, as used in natural language processing, is a group of two or three words that appear together frequently in a text. These n-grams are used as features for analysis since they identify recurring patterns or themes in the text data.

N-grams can be applied to the detection of fake news in order to identify common word sequences associated with it, such as "unverified sources," "conspiracy theory," or "hoax debunked." By recognizing these frequently occurring n-grams, a more powerful model for identifying and classifying bogus news stories can be built.

Generally speaking, the problem and dataset at hand determine the proper n-gram size. For instance, there are circumstances in which it could be advantageous to use only unigrams, or single words. However, there are other circumstances in which using bigrams (two-word sequences) or trigrams (three-word sequences) may be more beneficial. The statement "The sun is shining, and the birds are singing in the park" has the following instances of unigrams, bigrams, and trigrams:

Unigrams

1. "The"

2. "sun"

Bigrams

1. "The sun"

2. "sun is"

Trigrams

1. "The sun is"

2. "sun is shining"

TF-IDF

Using (TF-IDF) vectors is another popular feature extraction method for detecting fake news. Term frequency, or the frequency of each word in a document, is calculated using this method. The word's frequency throughout the entire corpus is then used to alter it.

The first step in determining TF-IDF scores is figuring out the term frequency (TF), which is a measure of how frequently a word appears in a document. Then, for every word, the inverse document frequency (IDF) is calculated, which indicates how uncommon the word is throughout the corpus.

The logarithm of divided by the number of documents containing the particular word yields the IDF. Then, each word in the text's TF-IDF rating is obtained by multiplying its TF by its IDF. Equation (3.1) provides the formula for determining the TF-IDF scores for a word or term in

$$\textbf{TF-IDF} = \textbf{TF} * \textbf{IDF} \quad \textbf{(3.1)}$$

Here:

Inverse document computed follows:

$$\textbf{IDF} = \textbf{log} (\textbf{N} / \textbf{n}) \quad \textbf{(3.2)}$$

In provided formula, "N" represents the entire quantity of records within the corpus, while "n" signifies the count of documents in the corpus containing the specific term.

4. ENHANCED J48 CLASSIFIER

Decision trees are highly popular and extensively employed algorithms in machine learning, particularly for classification tasks. They construct a tree-like model representing decisions and their associated potential outcomes. Among these algorithms, the J48 algorithm holds prominence as an algorithm for decision trees developed within the software for machine learning, Weka.

With additional features designed to improve the classifier's accuracy and efficiency, the improved J48 algorithm iteration is an expansion of the initial strategy. The ability to produce unpruned trees is one of these improvements that stands out. Since every tree was cut in the original J48 algorithm, there was a chance of overfitting and reduced accuracy. The improved J48 algorithm can produce more accurate models since it allows unpruned trees to be created, which makes it more capable of handling particular dataset features.

The confidence factor parameter in the improved J48 method is a useful tool for controlling the amount of pruning that is done to the tree. A more aggressive trimming technique is triggered by a greater confidence factor, whereas a more cautious approach is prompted by a lower confidence factor. This parameter allows you flexibility in adjusting the balance between accuracy and simplicity of the model.

Reduced error pruning with backfitting, which involves repeatedly pruning and regrowing the decision tree, is another feature of the improved J48 algorithm. The algorithm's overall efficacy is boosted by this iterative process, which allows for increased accuracy and more effective use of training data.

5. ENHANCED NAÏVE BAYES CLASSIFIER

A development of the traditional Naïve Bayes algorithm, the Enhanced Naïve Bayes classifier, also known as the Complement Naïve Bayes classifier, was created to address the problem of class imbalance in classification problems. Class probabilities in the classic Naïve Bayes method are computed by multiplying the probabilities of individual features assigned to a specific class. However, when dealing with imbalanced datasets—that is, when one class has a significantly higher number of instances than the other—this method may display bias in favor of the majority class.

This issue is resolved by the Enhanced Naïve Bayes classifier, which takes into account the complement of the class distribution—that is, the proportion of examples that do not fall into any class. More specifically, the algorithm determines the likelihood of each feature given the complement of the class, as opposed to computing the likelihood of each characteristic given a class. This change aids in addressing the problems that the dataset's class imbalance presents. The Complement Naïve Bayes classifier is particularly useful for improving Naïve Bayes performance in text classification problems where there is a significant difference in the number of positive and negative occurrences.

6. KNN CLASSIFIER WITH ENHANCEMENT

The K-Nearest Neighbors (KNN) algorithm is a simple yet effective technique for regression and classification problems. Within the training dataset, the KNN method finds the K closest neighbors of a given query instance. It then allocates the query instance to either the class that occurs the most frequently (class classification tasks) or the K nearest neighbor average (regression tasks).

The K-Nearest Neighbors (KNN) technique works well, but it has certain limitations. These include being sensitive to unimportant and noisy features, being computationally demanding in high-dimensional data, and performing poorly in datasets that are unbalanced. Improvements to the KNN algorithm are applicable to overcome these issues. Using the IBk algorithm is one method that can assist reduce the influence of features that are irrelevant or noisy. Furthermore, by using the Bagging technique, which combines several KNN models, the algorithm's robustness and performance can be improved in a range of situations.

An improved version of the KNN algorithm, the IBk method provides more versatility by allowing for the selection of features, different weighting schemes, and distance metrics that may be customized. A query instance's K closest neighbors are determined by the IBk algorithm. These neighbors are then given weights according to how far away they are from the query instance. After that, the weighted contributions from the neighbors are combined to create a weighted total that is used to predict values for the

Journal of Innovative Engineering and Research (JIER)
Vol. - 7, Issue -2, October, 2024, pp. 9-13 (5 pages)

query instance in regression tasks or class labels in classification tasks.

7. IMPROVED RANDOM FOREST CLASSIFIER

The Random Forest algorithm's performance and accuracy have been improved with the creation of the Enhanced Random Forest Classifier. By mixing several decision trees, the Random Forest algorithm is an ensemble learning technique that improves the robustness and accuracy of classification problems. A randomly chosen portion of the training data and features are used to build each tree in the Random Forest. The predictions from each individual tree in the ensemble are combined to determine the final classification.

AdaBoostM1 is used to strengthen the Enhanced Random Forest Classifier in the ES-ECA. In order to produce a strong classifier, this boosting approach combines several rounds of a weak classifier, in this case the Random Forest algorithm. Iterations are the way the AdaBoostM1 algorithm works. It finds the cases that were incorrectly classified in the previous iteration, gives them greater weights, and uses the modified dataset to train a new weak classifier instance. In the end, a more reliable and accurate overall classification model is produced by combining the predictions of each individual weak classifier to establish the final classification choice. In addition, the Enhanced Random Forest Classifier has a maximum depth restriction of 10 and is designed with a larger ensemble of 500 trees. These configurations help to improve classification accuracy while reducing overfitting.

8. IMPROVEMENTS TO STACKING CLASSIFIER

A machine learning system called the enhanced stacking classifier combines the predictions made by many basic classifiers to improve prediction accuracy. This technique uses an advanced stacking ensemble approach that uses many basis classifiers. It then uses a meta-classifier to combine the basic classifiers and generate the final prediction.

A meta-classifier harmonizes the base classifiers' predictions after they have been trained on the dataset. The Random Forest algorithm is used in the suggested algorithm to build the meta-classifier, and the AdaBoostM1 algorithm is then used to further boost it. Because of its resilience, capacity to handle noisy data, and ability to reduce overfitting, the Random Forest method was chosen for the meta-classifier, making it an appropriate fit for this ensemble technique.

Random Subspace algorithm

A machine-learning method for feature selection in classification tasks is the Random Subspace algorithm. It is commonly used in conjunction with ensemble techniques such as Random Forests and is designed to reduce the likelihood of overfitting while optimizing model performance. For the purpose of the classification task, the algorithm chooses at random a subset of features from the original dataset. The model is trained on a smaller subset of features since this subset is usually less than the original feature set.

9. FINDINGS FROM THE EXPERIMENT AND CONVERSATIONS

The Liar dataset, which is used to detect fake news, is used in this part to evaluate the performance of the ES-ECA. Statements made by politicians are included in the publicly available Liar dataset. The following truth categories are applied to these statements: "true," "mostly true," "half true," "barely true," "false," and "pants on fire." The collection consists of textual elements like statements and metadata,

Java is used implement ES-ECA, which uses the Liar dataset to assess its performance. The assessed primary, F1-score, precision. The following the definition of proportion (correct forecasts) to all predictions made is known as accuracy.

The formula for accuracy is (true positives + true negatives) / (true positives + true negatives + false positives + false negatives). (3.3)

Precision defined as ratio to all positive predictions generated. It can be represented as follows:

Precision can be calculated as true positives / (true positives + false positives). (3.4)

The following formula, known as recall, measures the proportion to actual

= recall. (3.5)

which provides a fair evaluation of both metrics and is calculated as follows,

Equal to 2 * recall * precision / (recall + precision). (3.6)

The assessment measures provide an objective way to rate how well the algorithm detects false information. To facilitate comparison, each classifier's performance is also assessed separately using the same measures based on 3.1.

Certainly 3.1 visually represents distinct (Naive Bayes), (K-Nearest Neighbors), (Random Forest), (Enhanced Stacking Ensemble Classification Algorithm)—applied to a dataset.

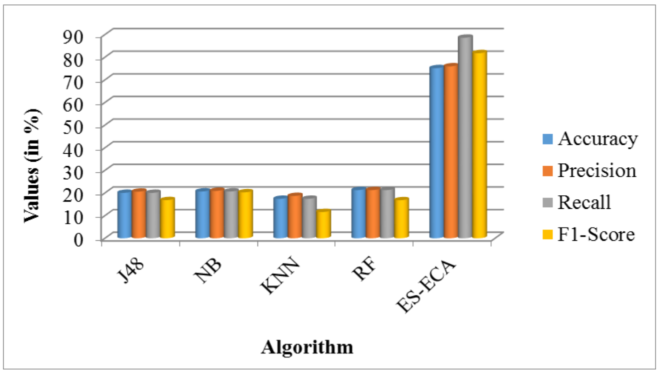

Figure 3.1: Comparing the Performance of Classifiers

superior performance of the ES-ECA classifier over the other classifiers in all evaluation metrics is evident from Figure 3.1.

It obtained a 75.18% accuracy, a 75.98% precision, an 88.62% recall, and an 81.81% F1-score. KNN received the lowest scores for all measures, whereas the other classifiers produced lower values for accuracy, precision, recall, and F1scores. These results definitely indicate that the ES-ECA best choice, as it has the maximum efficacy in detecting fake news.

10. CONCLUSION

All things considered, the spread of false information in our culture emphasizes how urgent it is to put efficient detection methods in place., which was presented in this chapter, has proven to be more effective than previous methods among other metrics. suggested methodology uses the metric for feature extraction, segments individual statements using preprocessing techniques, and generates n-grams as feature representations. Improved versions of the J48 decision tree method, Naive Bayes algorithm, and k-Nearest Neighbors algorithm are among the basis classifiers that are combined in the improved stacking ensemble approach. The testing results unequivocally confirm the recommended algorithm's efficacy in the field of fake news identification. This algorithm is a useful instrument for addressing and lessening the negative effects that bogus news has on society. As such, it presents a huge opportunity for researchers, journalists, and politicians to detect and prevent fake news from spreading in the future.

REFERENCES

1. Ahmad, I., Yousaf, M., Yousaf, S., & Ahmad, M. O. “Fake news detection using machine learning ensemble methods”, Complexity, Vol-2020, Page 1-11, 2020.

2. https://datasets.activeloop.ai/docs/ml/datasets/liar-dataset/

3. Yoav Freund Robert E. Schapire. “Experiments with a New Boosting Algorithm Machine Learning”, Proceedings of the Thirteenth International Conference, Page 1-9,1996

4. Jiang, T. A. O., Li, J. P., Haq, A. U., Saboor, A., & Ali, A. “A novel stacking approach for accurate detection of fake news”, IEEE Access, 9, Page 22626-22639, 2021.

5. Ye Tian, Yang Feng. “RaSE: Random Subspace Ensemble Classification”, Journal of Machine Learning Research .Vol-22, Issue-7, Page 1-93,2021.

6. Mahabub, A. “A robust technique of fake news detection using Ensemble Voting Classifier and comparison with other classifiers”, SN Applied Sciences, Vol-2, Issue-4, Page 525-536, 2020.

7. Nasir, J. A., Khan, O. S., & Varlamis, I. “Fake news detection: A hybrid CNN-RNN based deep learning approach”, International Journal of Information Management Data Insights. Vol.1, Issue-1, Page 100007, 2021.

8. Umer, M., Imtiaz, Z., Ullah, S., Mehmood, A., Choi, G. S., & On, B. W. “ Fake news stance detection using deep learning architecture (CNN-LSTM)”, IEEE Access, 8, Page 156695-156706, 2020.

9. Saleh, H., Alharbi, A., & Alsamhi, S. H. “OPCNN-FAKE: Optimized convolutional neural network for fake news detection”, IEEE Access 9, Page 129471-129489, 2021.