ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 13 Issue 10 October 2024, Page No. 26576-26588
ISSN: 2319-7242 DOI: 10.18535/ijecs/v13i10.4921

Developing A Web Browser Extension to Prevent the Spread of Fake News

Research Assistant Merve Esra TAŞCI [1*], Assistant Proffesor Yonca BAYRAKDAR YILMAZ [2]

[1] Sabahattin Zaim University, Faculty of Engineering and Natural Sciences, Software Engineering Department, İstanbul, Türkiye

[2] Çanakkale Onsekiz Mart University, Faculty of Engineering, Computer Engineering Department, Çanakkale, Türkiye

Abstract

Fake news causes huge social problems such as misdirecting and provoking the masses, creating an atmosphere of chaos by spreading fear. Detecting and stopping the dissemination of fake news has been an important and priority issue due to its rapid spread, difficulty to detect, and negative effects. In our study, a new Chrome extension that detects fake news has been developed in order to detect and prevent the spread of fake news. In the study; natural language processing, data mining methods, and various machine learning techniques for instance Passive Aggressive, Random Forest, Support Vector Machine, AdaBoost, XGBoost, and Long-Short-Term Memory (LSTM) algorithms are used. The accuracy rates of the algorithms, confusion matrices, AUC rates, and ROC curves were compared. The created machine learning model has been implemented in the online internet environment by using Flask and Rest API in the Python program. Finally, the Chrome extension interface was built using Javascript, HTML, and CSS. The LSTM algorithm gave the highest prediction result with a rate of 90.72% compared to other algorithms.

Keywords: Fake news, Machine learning algorithms, Natural language processing, Data mining, Python programming

1. Introduction

The widespread usage of the internet has led to social media platforms replacing traditional mass communication tools. Particularly in the last decade, the digital world and social media have become crucial and popular for people to interact and socialize with each other. Globally, individuals would rather follow breaking news on social networks. Accessing the internet from mobile devices has become a habit in modern technological societies (Klyuev, 2018: 1). Social media is widely used by large users due to the opportunities it provides such as being free, easily accessible information and the ability to individuals to express themselves (Shu, et al., 2018). Accession to the internet easily has resulted in the sharing and dissemination of unverified news, rumors, and misinformation. The accuracy and reliability of the accessed news have become a significant issue affecting individuals and societies. At the same time, it is effortless to access any news content on online platforms (Vo, et al., 2018). The difficulty of controlling digital data has led to the dissemination of unstructured data containing text, images, and videos that are unreal and have no accuracy. These unreal data spread through likes, shares, and recommendation systems on various web pages and social media platforms. The great majority of these unreal data constitutes fake news. Fake news is the intentional transformation of a topic that captures the public's interest into fake news and is presented to the target users. Fake news is generally created and spread for reasons such as damaging the reputation of individuals or institutions, diminishing credibility, creating distrust and chaos in societies, instilling fear, misleading the masses, or making money through advertising. What sets fake news apart from other types of news is its intentional manipulation of the emotions, thoughts, and actions of society to influence them in the desired direction (Tandoc, 2019). Preventing the dissemination of fake news is challenging due to more interest in fake news rather than true news. Twitter, Facebook, and Instagram widely used because of their popularity by individuals or groups that create and spread fake news content. Fake news is mostly created and disseminate on Twitter (Vo, et al., 2018). During the 2016 American presidential elections, several tweets that are classified as fake about Trump were disseminated on Twitter. (Bovet, et al., 2019). Following the US Presidential election, lots of fake news spreading on social media. Spotting fake news has become a hot topic because of the detrimental effects of fake news. Fake news has noteworthy negative consequences for individuals and communities. So, Chrome extension have been developed to detect and prevent the dissemination of fake news.

2. Related Works

Numerous research have focused on detecting and preventing the spread of fake news. Patankar et al. created a novel browser extension that identifies fake news and suggests novel news to users, using the Latent Dirichlet Allocation (LDA) approach. (Patankar, et al., 2019: 232-234). Shu et al. created a browser extension named "FakeNewsTracker" to assist people analyze and forecast fake news on Twitter. Their technique includes merging news content with user behavior. LSTM algorithm was used to detect fake news. (Shu, et al., 2018). Botnevik et al. created the "BRENDA" browser extension, which detects fake news by extracting page URLs. They used the Newspaper3k Python program to collect URL content, which was then evaluated using the SADHAN model designed particularly for detecting fake news. (Botnevik, et al., 2020). Paka et al. created a browser extension called "Cross-SEAN" to analyze the validity of tweets sent on Twitter. This extension not only identifies fake news but also informs users with the percentage possibility that the news detected as fake. (Paka, et al., 2021: 10). Cui et al. created a new deep learning framework called "dEFEND". This framework uses both text content and user comments to increase the accuracy of fake news detection (Cui, et al., 2019). Kesarwani et al. detected fake news with 79% accuracy using the K-Nearest Neighbor (KNN) algorithm. They conducted their study on a Buzz Feed dataset (Kesarwani, et al., 2020). Reis et al. created a novel fake news detector that uses various machine learning methods, including K-Nearest Neighbor (KNN), Random Forest, Naive Bayes, Support Vector Machine (SVM), and Extreme Gradient Boosting (XGB). They trained their dataset using various algorithms and compared their performance in terms of accuracy, AUC (Area Under the Curve), and F1 score rates. Their findings revealed that the XGB and Random Forest algorithms achieved the greatest AUC rates among the algorithms examined. (Reis, et al., 2019).

Dua et al. created I-FLASH, a fake news detection system that not only identifies fake news but also explains how it was classified. This unique detector provides users with information into why particular material is considered fake. They conducted their research using two new datasets, FactCheck and FactCheck2, which were gathered from numerous Twitter accounts and websites covering a wide range of socioeconomic topics such as education, crime, and technology. They tested and compared the performance of numerous models, including logistic regression with TF-IDF for machine learning, bidirectional LSTM with GloVe word embeddings for deep learning, and pre-trained Bidirectional Encoder Representations from Transformers models. These models were assessed individually on both datasets. (Dua, et al., 2023). Various studies have employed diverse methodologies for detecting fake news. Davoudi et al., for instance, introduced a novel approach using propagation tree and stance network analysis in their automatic fake news detection system, termed DSS. The DSS model comprises three key components: Dynamic Analysis, Static Analysis, and Structural Analysis. In Dynamic Analysis, a recurrent neural network was employed to capture the evolution patterns of propagation trees and stance networks over time. Static Analysis utilized a fully connected network to capture comprehensive characteristics of propagation trees and stance networks at the detection endpoint. The Structural Analysis component utilized the node2vec algorithm for graph embedding to encode the structural attributes of propagation trees and stance networks. These components collectively determined the accuracy of text analysis outputs. The developed DSS model demonstrated exceptional accuracy in detecting fake news, particularly in the early stages of dissemination (Davoudi, et al., 2022). Palani et al. argued that traditional methods for detecting fake news are often ineffective. In response, they developed CB-Fake, a novel multimodal tool based on BERT and CapsNet. Unlike conventional approaches, CB-Fake integrates textual and visual information extracted from news articles to determine their authenticity. The model enhances classification accuracy by leveraging both textual content and visual features. A significant innovation of their study lies in the application of CapsNet, a capsule neural network model that employs the routing-by-agreement algorithm to extract informative visual features from news article images (Palani, et al., 2022). Despite numerous studies on preventing fake news,achieving high success in semantic modeling of texts has often been elusive. Addressing this challenge, Zhang et al. have introduced a novel deep learning tool for rapid fake news detection in cyber-physical social services. Their study utilized a dataset comprising short texts sourced from Chinese social media platforms. To accommodate the characteristics of these texts, which are brief and in Chinese, they employed a convolution-based neural computing framework for robust semantic analysis. This approach ensures both efficient processing speed and effective detection capabilities. The developed tool has shown remarkable performance in their experiments (Zhang, et al., 2023).

Paka et al. developed a browser extension named 'Cross-SEAN' designed to identify fake news shared on Twitter. This extension not only detects fake news but also provides users with the percentage likelihood of a detected news item being fake (Paka, et al., 2021). Warman et al. have introduced a novel browser extension for detecting fake news, leveraging the BERT model, which they regard as highly effective for this purpose. This extension not only performs real-time detection of fake news but also offers users clear and comprehensible explanations for its classifications (Warman, et al., 2023). Valesco et al. have created a browser extension called FactIt, employing the logistic regression algorithm for detecting fake news. The extension's performance was assessed using the ISO 9126 software quality model, where it received high ratings across various criteria including functionality, reliability, usability, efficiency, maintainability, and portability (Velasco, et al., 2023). Borges developed an extension named Gossip Checker, focusing on fake news detection. In this model, Borges utilized TF-IDF vectorization coupled with Passive Aggressive Classifiers to enhance accuracy and efficiency in identifying misinformation (Borges, 2022).

3. Materials and Methods

3.1. Material

Political news, social events, cultural and art news, and new dataset built from gathering from three datasets were used in the study. Three datasets -political, social events, cultural and art news datasets- were obtained from the Kaggle site. Here's a detailed description:

The political dataset includes 23,481 fake and 21,417 real news. Too much data might cause issues such as increased memory use and longer work hours. 5,000 fake news and 4,800 real news data were selected to improve machine learning efficiency and memory usage. The social events dataset includes 72,134 data. For the study, 10,000 data were selected. The culture and art news dataset include 23,196 data. For the study, 10,000 data were selected.

The final dataset was created choosing 16,285 data from the political, social events, and culture and art news datasets. This selection aimed to maintain data homogeneity and ensure consistent accuracy rates across different subsets, while optimizing computational resources.

3.2. Method

The Chrome extension for detecting fake news was developed in two parts. In the first part, data mining techniques were performed on the datasets and finding best-performed model. The second part entailed implementing several programs including as CSS, HTML, JavaScript, and Python, to develop the extension itself.

The development of the Chrome extension progressed through stages including data mining applied to the datasets used and subsequent implementation of various programs and languages such as CSS, HTML, Javascript, and Python, as detailed in Figure 1.

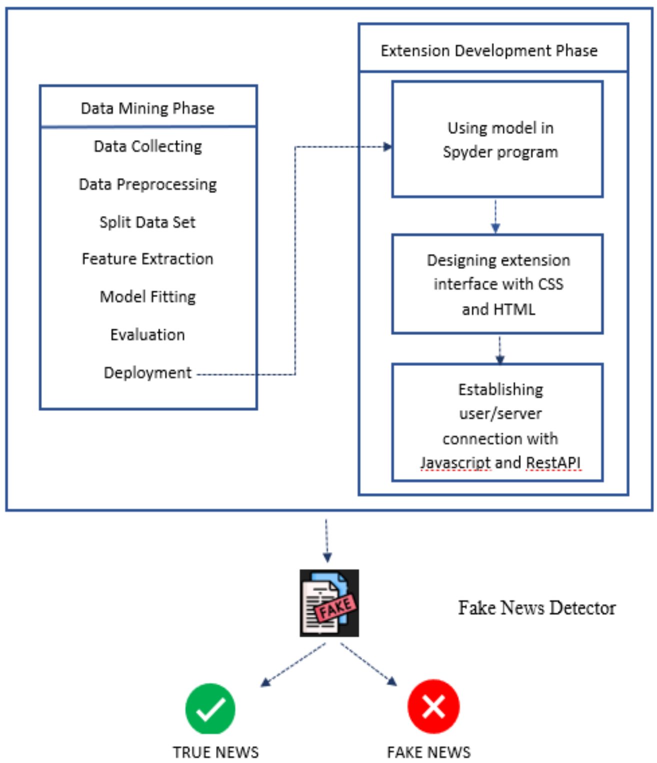


Figure 1. Applied stages during the development of the Chrome extension.

The algorithm for the data mining stage has been selected. A Chrome extension has been developed using Python along with CSS, HTML, JavaScript, and RestAPI. Upon execution of the extension, users are presented with "FAKE NEWS" if the article is determined to be fake, or "TRUE NEWS" if it is deemed authentic.

3.2.1. Data Mining Phase

The primary and pivotal stage involves collecting appropriate data for the dataset. Figure 2 provides a detailed overview of how the data mining process is implemented. In our study, we utilized three pre-existing datasets from previous research and constructed a fourth dataset by combining data from these sources. Following dataset creation, the subsequent step in the data mining process is data preprocessing. When the data sets taken from the Kaggle site are examined, the news adding columns such as news title, news text content, news topic, publication date, and news label. To streamline our analysis, unnecessary columns were removed, leaving only the news title, news content, and labels for use.

A new column was then generated by combining the news title and news text. Subsequently, a thorough check was conducted to ensure that there were no missing or corrupt data in either the rows or columns. With data integrity confirmed, the data preprocessing stage commenced, applying natural language processing operations to the dataset. Specifically, all uppercase letters were converted to lowercase, and non-word characters such as symbols and numbers were eliminated. Punctuation marks and unnecessary spaces were removed. Prepositions and conjunctions (Stopword) that do not have literal meaning but enable establishing a semantic relationship between words and phrases were removed as they would not be used in the data mining stage.

During the dataset splitting stage, 20% of the data was reserved for testing purposes, while the remaining 80% was allocated for training. In data mining, information is typically incorporated into databases in the form of text or numerical values. These expressions, encompassing text, numbers, and punctuation marks, are transformed into machine-readable language through a process known as vectorization. Various algorithms have been developed for this purpose, including the Term Frequency-Inverse Document Frequency (TF-IDF) algorithm utilized in our study.

TF-IDF is one of the frequently used statistical methods in determining the importance of one or more words used in a file, document, or dataset. Its basic idea is that a term's relevance shouldn't be dismissed based just on its low frequency if it occurs often in one area of a text or dataset but infrequently elsewhere. The algorithm posits that terms appearing frequently in certain sections may possess strong discriminatory capabilities and therefore should be considered during classification tasks (Liu, et al., 2018: 1).

During the model application stage, the training of the model with the training dataset involved employing various machine learning methods. These included the online learning algorithm Passive Aggressive Classification, supervised learning algorithms Support Vector Machine and Random Forest, boosting algorithms AdaBoost and XGBoost, as well as the deep learning algorithm LSTM across the four datasets. In the evaluation stage, the performance of these algorithms was compared based on metrics such as accuracy rates, confusion matrices, ROC curves, and AUC values. Subsequently, in the deployment stage, it was determined to integrate the LSTM algorithm into the Chrome extension for detecting fake news due to its superior performance. The trained LSTM model was serialized and saved to disk for use within the Spyder IDE program.

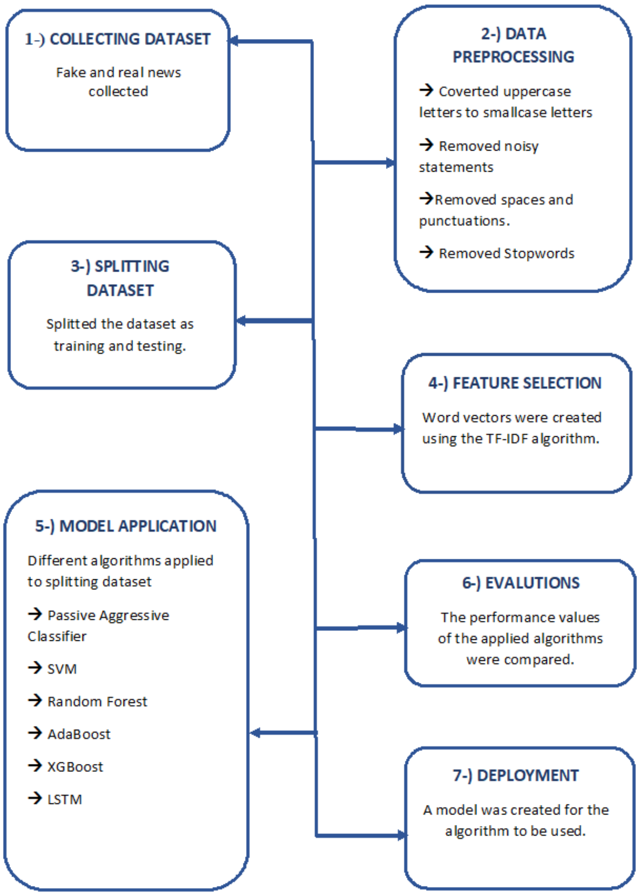


Figure 2. Data mining phase methods that applied at each stage.

3.2.2. Chrome Extension Development Phase

The Chrome extension was developed using the Spyder IDE program. Its popup interface was designed using HTML and CSS, while JavaScript was utilized to enable functionality for the buttons within the extension popup. The extension operates as a web application within a web environment, facilitated by the Flask framework to support its functionality.

Web applications are structured with a client-side interface (the user interface) and a server-side component responsible for processing and managing user interactions. The communication between the client and server is facilitated through a Rest API (Representational State Transfer Application Programming Interface). Rest API utilizes the HTTP protocol to transmit requests (using GET and POST methods) from the client to the server, receive responses, and facilitate communication between web applications and the server infrastructure. In our project, server requests are handled using Asynchronous JavaScript and XML (AJAX) along with the XML Http Request object. When an AJAX call is initiated by the browser, the XML Http Request is activated, prompting the web browser to send an HTTP request to the server. Upon receiving the request, the server processes the data using commands like GET, PUT, or POST, and subsequently returns the processed data to the web browser. The browser then displays this data on the user's page.

Initially, the model selected for use was saved to disk. To integrate this model into the extension, a Python file was created. This Python file incorporates the saved model file and manages the processing of news text entered into the text box within the user interface, applying the prediction function to this text. To support the functionality in the web environment, Flask framework was utilized. Flask is a lightweight micro-framework developed in Python, specifically designed for building web applications. A web framework like Flask provides essential tools, libraries, modules, and technologies required for developing various types of web-based applications, such as web pages, blogs, and wikis.

3.2.3. The User Interface of the Fake News Detector

The popup interface includes a text area where users can input news text upon clicking the extension. Figure 3 shows the interface of the Chrome extension.

Figure 3. The user interface of Chrome extension.

The user interface features a text box where users can manually input news titles or text for verification without navigating away from their current webpage. Figure 3 provides a visual representation of the Chrome extension's interface. Below the text box, two buttons are available: CHECK and RESET. Users can enter news content into the text box and click the CHECK button to verify its authenticity. Clicking the RESET button clears the text box, useful for resetting after checking news. If the CHECK button is clicked without entering any text, a message "This area cannot be empty. Please enter any text to check." appears in the popup. Upon verification:

- If the entered news is confirmed as true, the popup displays "TRUE NEWS" accompanied by a green checkmark.
- If the entered news is identified as fake, the popup shows "FAKE NEWS" with an exclamation mark.

This feature enables users to easily check news content directly from the Chrome extension.

4. Experimental Results

The study conducted datasets that contain political, social events, and culture-arts news, and new dataset combined from these three datasets. Varied machine learning - Support Vector Machine, Random Forest, Passive Aggressive Classification, AdaBoost, XG Boost, and LSTM and deep learning algorithm -LSTM- were used on these datasets. Accuracy rates, cross-validation scores, confusion matrices, and Area Under the Curve (AUC) performance metrics were utilized to compare the algorithms' performance in the study.Using performance matrices, we aim to determine the best effective algorithm for detecting fake news.

4.1. Comparison of Accuracy Rates and Cross-Validation Rates

Table 1 shows the accuracy and cross-validation rates for datasets including political, social, cultural and arts news, as well as a novel dataset created by combining these three datasets.

Table 1. Accuracy and Cross-Validation Rates.

	Dataset consisting of Political News		Dataset consisting of Social Events		Dataset consisting of Culture and Art News		Dataset consisting of news from three datasets	
	Accuracy Rates	Cross-validation Rates	Accuracy Rates	Cross-validation Rates	Accuracy Rates	Cross-validation Rates	Accuracy Rates	Cross-validation Rates
Passive Aggressive	99.44%	99.60%	93.20%	91.90%	77.90%	78.34%	83.33%	84.24%
SVM	99.54%	99.60%	93.70%	91.76%	82.75%	82.84%	87.50%	88.03%
Random Forest	99.74%	99.80%	90.00%	88.79%	81.30%	81.99%	85.20%	86.11%
AdaBoost	99.90%	99.95%	91.80%	91.60%	80.05%	80.32%	86.12%	86.67%
XGBoost	99.90%	99.95%	93.80%	93.56%	81.75%	81.16%	88.33%	88.48%
LSTM	99.95%	99.93%	95.65%	95.15%	89.30%	89.45%	90.72%	90.84%

The Passive Aggressive classification algorithm achieved the lowest accuracy rate at 99.44% on political news dataset. The LSTM algorithm achieved the highest accuracy rate of 99.95%. Following LSTM, AdaBoost and XGBoost algorithms also demonstrated high accuracy rates. Given the high accuracy rates observed, cross-validation was employed to assess potential overfitting.

Cross-validation is a method used to estimate how well a model trained on training data will generalize to unseen validation data. It evaluates the model's performance on validation data while training it on the remaining training data. Table 1 presents the rates obtained after applying cross-validation to the models. It is observed that the prediction rates on the test and validation data are close to each other after applying cross-validation. According to the results, it is confirmed that there is no overfitting. It is essential to note that the assumption of overfitting should not be based solely on very high accuracy rates. The use of a high-quality dataset significantly contributed to achieving these high accuracy rates.

When examining the dataset comprising social events, the algorithms achieved the following accuracy rates for detecting fake news: Passive Aggressive classification at 93.2%, Support Vector Machine at 93.7%, Random Forest at 90.0%, AdaBoost at 91.8%, XGBoost at 93.8%, and LSTM at 95.65%. Similar to the political dataset, the LSTM algorithm exhibited the highest accuracy rate. XGBoost and Support Vector Machine algorithms achieved the next highest accuracy rates following LSTM. According to the other algorithms Random Forest achieved the lowest accuracy rate at 90%.

High accuracy rates were achieved. Cross-validation technique was used to check whether there was overfitting. It was observed that prediction rates on both test and validation data were very close. We could conclude that there is no overfitting. The social events dataset performed well and demonstrated effective performance in predicting fake news despite not achieving as high accuracy rates as the political dataset For the dataset of culture and arts news, the algorithms achieved the accuracy rates for predicting fake news respectively: Passive Aggressive classification at 77.90%, Support Vector Machine at 82.75%, Random Forest at 81.30%, AdaBoost at 80.05%, XGBoost at 81.75%, and LSTM at 89.30%. As observed in previous datasets, the LSTM algorithm achieved the highest accuracy rate. The Support Vector Machine and XGBoost algorithms showed the next highest accuracy rates following LSTM. Passive Aggressive achieved the lowest accuracy rate at 77.90%.

Algorithms applied to the dataset of culture and arts news achieved lower accuracy rates compared with datasets of political and social events. Cross-validation was employed to ensure whether there was overfitting. The accuracy rates for test and validation data were very similar. This consistency shows that there was no overfitting. The performance of the dataset of culture and arts was remarkable despite having lower accuracy rates compared to other datasets.

The last dataset combining news from the three datasets -political, social events, and culture-arts- the algorithms achieved the accuracy rates for detecting fake news respectively: Passive Aggressive 83.33%, Support Vector Machine 87.5%, Random Forest 85.2%, AdaBoost 86.12%, XGBoost 88.33%, and LSTM 90.72%. The LSTM algorithm demonstrated the highest accuracy rate as in previous values. The XGBoost and Support Vector Machine algorithms achieved the next highest accuracy rates after LSTM. The passive-aggressive achieved the lowest accuracy rate. These findings highlight the effectiveness of LSTM on detecting fake news. Cross-validation was applied to ensure whether there was overfitting because of high accuracy rate. It was observed that the prediction rates on the test and validation data were close to each other after cross-validation. We could conclude that there was no overfitting.

4.2. Comparison of Confusion Matrices Values

The confusion matrix is used for the pupose of visually represents the performance of algorithms (Shahbaz et al., 2019). The confusion matrices values of each dataset are shown in Table 2.

Table 2. Comparison of Confusion Matrices Values.

	Dataset consisting of Political News				Dataset consisting of Social Events				Dataset consisting of Culture and Art News				Dataset consisting of news from three datasets			
	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP
Passive Aggressive	1048	5	6	901	903	67	69	961	279	216	226	1279	1110	317	226	1604
SVM	1049	4	5	902	909	61	65	965	234	261	84	1421	1157	270	137	1693
Random Forest	1052	1	4	903	878	92	108	922	234	261	113	1392	1149	278	204	1626
AdaBoost	1053	0	2	905	874	96	68	962	193	302	97	1408	1104	323	129	1701
XGBoost	1053	0	2	905	891	79	45	985	201	294	71	1434	1152	275	105	1725
LSTM	1053	0	1	906	895	60	27	1018	292	190	24	1494	1151	252	50	1802

The LSTM, XGBoost, and AdaBoost algorithms performed better on political news dataset. Both AdaBoost and XGBoost algorithms, 1053 fake news were correctly classified as fake. Only 2 out of 907 real newscwere misclassified as fake. The LSTM algorithm correctly classified all 1053 fake news as fake. Only one out of 907 real news classified as fake. The passive-aggressive algorithm performed very poorly. It classified 5 out of 1053 fake news as real and 6 out of 907 real news as fake. These results highlight that the LSTM algorithm achieved the greatest accuracy and reliability when discriminating between fake and real news.

The Support Vector Machine, XGBoost, and LSTM algorithms outperformed dataset of social events. The Support Vector Machine classified 61 out of 970 fake news as real and 65 out of 1030 real news as fake. The XGBoost algorithm classified 79 out of 970 fake news as real and 45 out of 1030 real news as fake. The LSTM algorithm classified 60 out of 955 fake news as real and 27 out of 1045 real news as fake.

For the dataset of culture and arts news, the Support Vector Machine, XGBoost, and LSTM algorithms demonstrated the highest performance. The Passive Aggressive classification algorithm demonstrated the lowest performance. The Support Vector Machine misclassified 261 out of 495 fake news as real and 84 out of 1505 real news as fake. Similarly, the XGBoost algorithm misclassified 294 out of 495 fake news as real and 71 out of 1505 real news as fake. On the other hand, the LSTM algorithm misclassified 190 out of 482 fake news as real and only 24 out of 1518 real news as fake. When examining the confusion matrix, it was evident that, except for the LSTM algorithm, more than half of the fake news were incorrectly identified as real by all algorithms. Specifically, in the AdaBoost algorithm, nearly 75% of fake news were predicted as real.

Upon reviewing the outcomes, the LSTM algorithm exhibited higher accuracy compared to other algorithms. However, its performance was hindered by a lower rate of detecting fake news compared to other datasets. The models applied to the culture and arts dataset showed strong proficiency in identifying real news but struggled to detect fake news at a comparable rate. Table 1 illustrates an average detection rate of around 80% for fake news across the models. While these figures suggest effective performance in identifying fake news, the confusion matrix reveals that the models primarily excelled in correctly classifying real news as real, reflecting an imbalance in the dataset with more of real news than fake news. This dataset bias skewed the models towards better prediction of real news due to its larger representation in the training data.

Therefore, comprehensive evaluation criteria are essential for assessing algorithm performance.

Turning to the dataset comprising news from the three sources, the Support Vector Machine, XG Boost, and LSTM algorithms achieved the highest performance. The Support Vector Machine misclassified 270 out of 1427 fake news as real and 137 out of 1830 real news as fake. XG Boost algorithm misclassified 275 out of 1427 fake news as real and 105 out of 1830 real news as fake. Finally, the LSTM algorithm misclassified 252 out of 1403 fake news as real and 50 out of 1852 real news as fake.

4.3. Comparison of Roc Curves

The ROC curve and AUC are important metrics for evaluating classification methods. The ROC curve is a probability curve that shows the model's performance at various categorization criteria. It clearly illustrates how effectively the model distinguishes between classes. AUC, or Area Under the Curve, measures the model's ability to discriminate between classes. A higher AUC suggests better model performance since it represents more separability of classes. Fundamentally, the AUC evaluates the model's expected separability of positive and negative classes. This graphical depiction is important for evaluating the model's performance in classification tasks, with a higher AUC suggesting more accurate predictions and greater class distinction.

Figures 5, 6, 7, and 8 depict ROC curves for each model applied to the dataset. In Figure 5, all models exhibit AUC values above 99%, resulting in nearly identical curves. Such high values signify exceptional ability to distinguish between real and fake news. Moving to Figure 6, LSTM, XGBoost, and Support Vector Machine algorithms demonstrate superior performance in distinguishing true and fake news, with LSTM achieving an AUC of 95.57%, XG Boost 93.74%, and Support Vector Machine 93.70%. Figure 7 highlights LSTM, Support Vector Machine, and Passive Aggressive classification as the top performers in terms of AUC rates. Notably, Passive Aggressive achieves high AUC despite lower accuracy. Conversely, Random Forest, AdaBoost, and XG Boost, with average accuracy rates around 80%, show lower AUC values. Analysis across different datasets reveals significantly lower AUC rates for models applied to culture and arts datasets, indicating less effective discrimination between fake and real news. In Figure 8, LSTM, Support Vector Machine, and XG Boost also show the highest AUC rates, with Passive Aggressive recording the lowest at 82.72%.

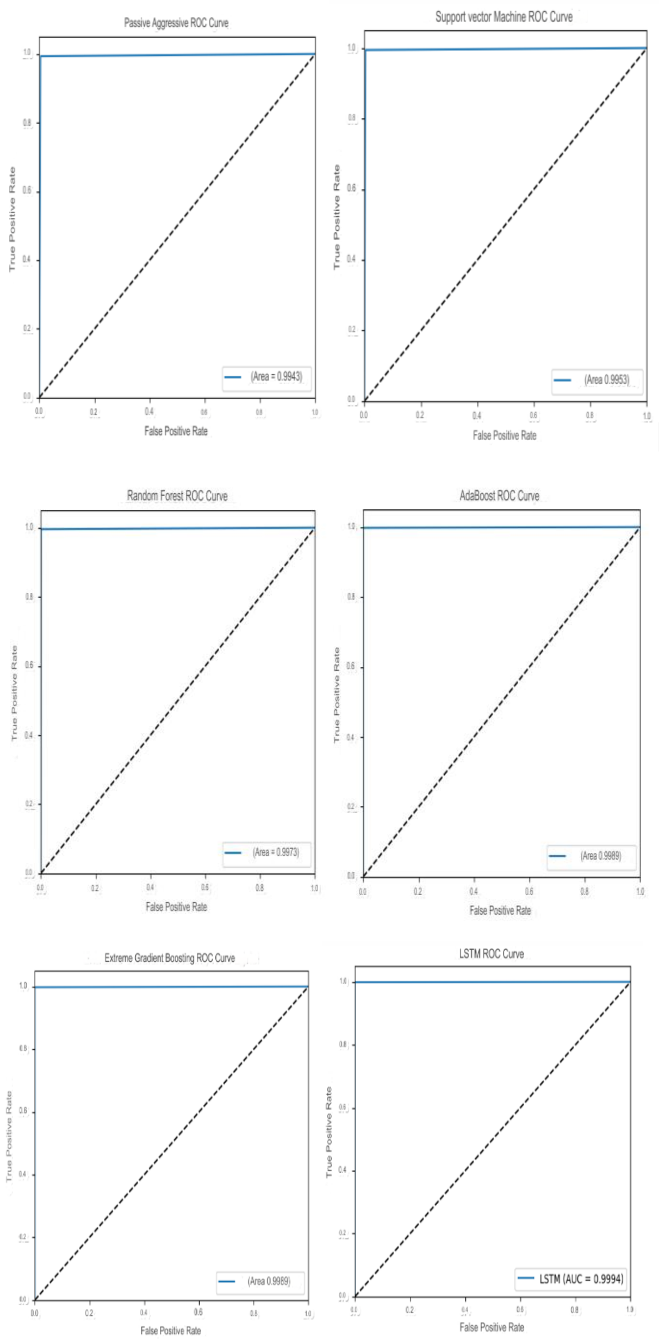


Figure 5. ROC Curves of Models for Political Dataset.

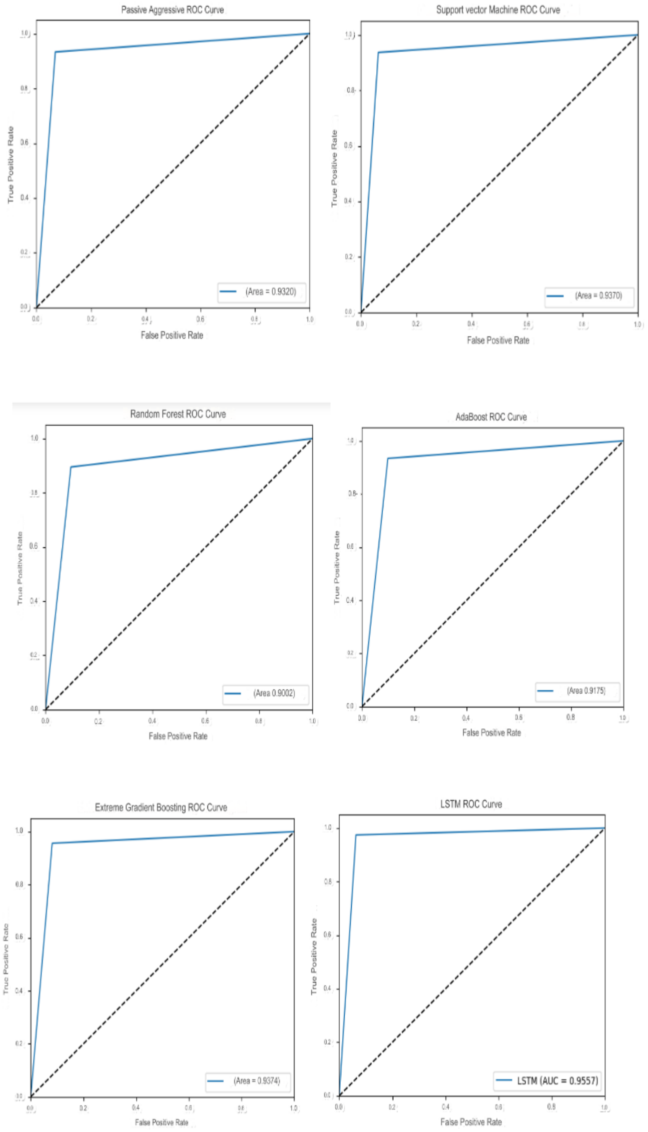
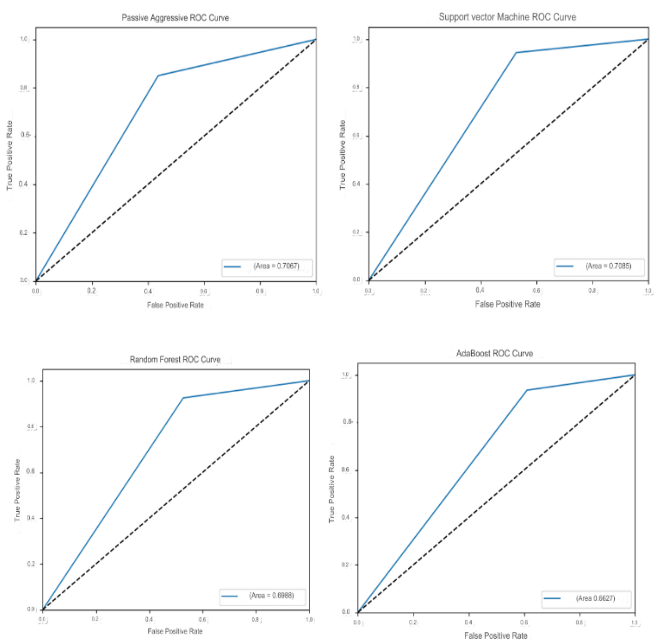


Figure 6. ROC Curves of Models for Social Events Dataset.



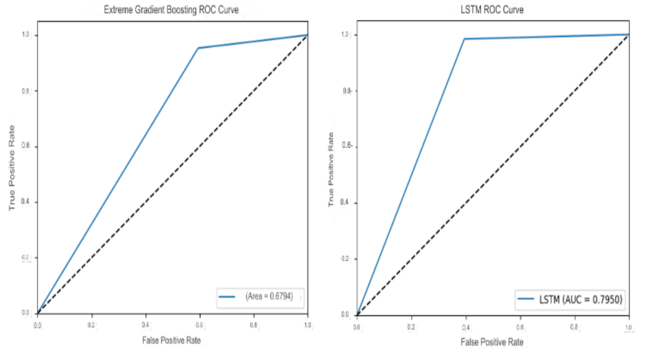


Figure 7. ROC Curves of Models for Culture and Arts Dataset.

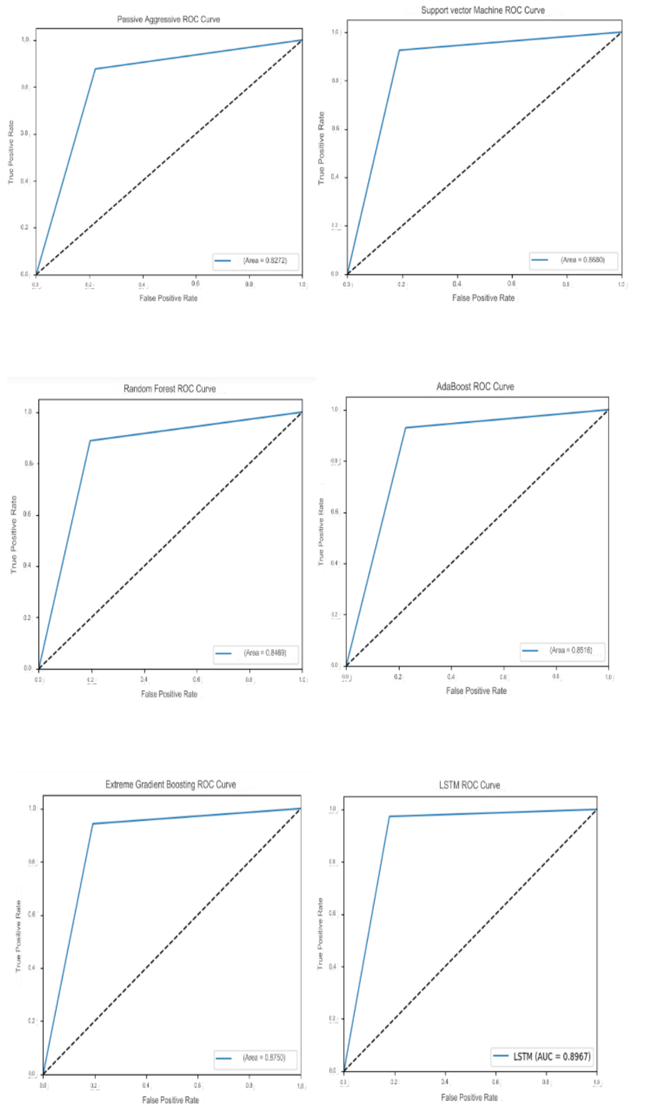


Figure 8. ROC Curves of Models for Datasets Consisting of Political, Social Events, and Culture-Arts News.

4.4. Comparison of AUC Rates

Table 3 presents the AUC rates of models applied to each dataset. In the political news dataset, all models achieve AUC values exceeding 99%, indicating highly effective discrimination between real and fake news. For the social events dataset, LSTM, XGBoost, and Support Vector Machine algorithms demonstrate superior performance, with LSTM achieving an AUC of 95.57%, XGBoost 93.74%, and Support Vector Machine 93.70%. In the culture and arts news dataset, LSTM, Support Vector Machine, and Passive Aggressive classification yield the highest AUC rates. Notably, Passive Aggressive achieves a high AUC despite lower accuracy rates. Random Forest, AdaBoost, and XGBoost algorithms, with average accuracy rates of roughly 80%, had lower AUC values in this dataset. In comparison, the culture and arts dataset has significantly lower AUC rates than other datasets. We could say that LSTM, Support Vector Machine, and XGBoost algorithms displayed high performance when looking at AUC rates. On contrast, Passive Aggressive has the lowest performance when contrasted with other algorithms.

Table 3. Comparison of AUC Rates.

	Dataset consisting of Political News	Dataset consisting of Social Events	Dataset consisting of Culture and Art News	Dataset consisting of news from three datasets
	AUC Rates	AUC Rates	AUC Rates	AUC Rates
Passive Aggressive	99.43%	93.20%	70.67%	82.72%
SVM	99.53%	93.70%	70.85%	86.80%
Random Forest	99.73%	90.02%	69.88%	84.69%
AdaBoost	99.89%	91.75%	66.27%	85.16%
XGBoost	99.89%	93.74%	67.94%	87.50%
LSTM	99.94%	95.57%	79.50%	89.67%

5. Conclusion

News is disseminated rapidly before its correctness can be confirmed with the advancement of technology and the effect of the human component. This rapid dissemination enables fake news to reach large audiences without discovery. Individuals and society are experiencing panic, dread, anxiety, turmoil, and disorder, highlighting the need for measures to combat the spread of fake news. This study provides an automated Chrome plugin that detects and prevents the spread of fake news. The study assessed and compared model performance using a variety of datasets and machine learning techniques, as well as several performance assessment criteria. Overall, the LSTM method outperformed the other three algorithms across four datasets. The next most successful algorithms were AdaBoost, XGBoost, and Support Vector Machine. A model was created using a dataset including news from three distinct sources, allowing the Chrome extension to successfully predict different forms of fake news. The final dataset's accuracy rates for fake news identification were 83.33% for Passive-Aggressive Classification, 87.5% for SVM, 85.2% for Random Forest, 86.12% for AdaBoost, 88.33% for XGBoost, and 90.72% for LSTM. Users can input news into the Chrome extension's text box to verify its authenticity via the extension popup. This study is distinct in allowing users the autonomy to check news of their choosing. By combining machine learning algorithms with natural language processing techniques, the study achieved a high fake news detection rate. The goal is to mitigate the negative effects of fake news. The developed Chrome extension enables swift identification and prevention of fake news spread, reducing its detrimental impact. Future enhancements could improve the extension's performance. Currently, the user interface provides a binary classification of news as fake or true. Adding an explanatory note for the classification could enhance user trust in the results. Additionally, displaying the percentage likelihood of news being true or fake could be beneficial. While the extension currently supports only English, incorporating other languages could broaden its user base. These and other features could be added to the Chrome extension in future updates.

References

1. Borges, T. L. (2022). “Chrome Extension for Misinformation Detection”. Aian Journal of Convergence in Technology. VOL 8 NO 3 (2022). https://doi.org/10.33130/AJCT.2022v08i03.002

2. Botnevik, B., Sakariassen, E. ve Setty, V. (2020). “BRENDA: Browser Extension for Fake News Detection”. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. doi:10.1145/3397271.3401396
3. Bovet, A. ve Makse, H. A. (2019). “Influence of fake news in Twitter during the 2016 US presidential election”. Nature Communications, 10(1). doi:10.1038/s41467-018-07761-2
4. Cui, L., Shu, K., Wang, S., Lee, D. ve Liu, H. (2019). “dEFEND". Proceedings of the 28th ACM International Conference on Information and Knowledge Management - CIKM ’19. doi:10.1145/3357384.3357862
5. Davoudi, M., Moosavi, M. R., Sadreddini, M. H. (2022). “DSS: A hybrid deep model for fake news detection using propagation tree and stance network”. Expert Systems with Applications, Volume 198. https://doi.org/10.1016/j.eswa.2022.116635
6. Dua, V., Rajpal, A., Rajpal, S. et al. (2023) I-FLASH: Interpretable Fake News Detector Using LIME and SHAP. Wireless Pers Commun 131, 2841–2874. https://doi.org/10.1007/s11277-023-10582-2
7. Kesarwani, A., Chauhan, S. S. ve Nair, A. R. (2020). “Fake News Detection on Social Media using K-Nearest Neighbor Classifier”. 2020 International Conference on Advances in Computing and Communication Engineering (ICACCE). doi:10.1109/icacce49060.2020.9154
8. Klyuev, V. (2018). “Fake News Filtering: Semantic Approaches”. 2018 7th International Conference on Reliability”, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). doi:10.1109/icrito.2018.8748506
9. Liu, C., Sheng, Y., Wei, Z. ve Yang, Y.-Q. (2018). “Research of Text Classification Based on Improved TF-IDF Algorithm”. 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE). doi:10.1109/irce.2018.8492945
10. Paka, W. S., Bansal, R., Kaushik, A., Sengupta, S. ve Chakraborty, T. (2021). “Cross-SEAN: A cross-stitch semi-supervised neural attention model for COVID-19 fake news detection”. Applied Soft Computing, 107, 107393. doi: 10.1016/j.asoc.2021.107393
11. Palani, B., Elango, S. & Viswanathan K, V. CB-Fake: A multimodal deep learning framework for automatic fake news detection using capsule neural network and BERT. Multimed Tools Appl 81, 5587–5620 (2022). https://doi.org/10.1007/s11042-021-11782-3
12. Patankar, A., Bose, J. ve Khanna, H. (2019). “A Bias Aware News Recommendation System”. 2019 IEEE 13th International Conference on Semantic Computing (ICSC). doi:10.1109/icosc.2019.8665610
13. Reis, J. C. S., Correia, A., Murai, F., Veloso, A., Benevenuto, F. ve Cambria, E. (2019). “Supervised Learning for Fake News Detection”. IEEE Intelligent Systems, 34(2), 76–81. doi:10.1109/mis.2019.2899143
14. Shahbaz, M., Ali, S., Guergachi, A., Niazi, A., & Umer, A. (2019, July). Classification of Alzheimer's Disease using Machine Learning Techniques. In Data (pp. 296-303)
15. Shu, K., Mahudeswaran, D. ve Liu, H. (2018). “FakeNewsTracker: A tool for fake news collection, detection, and

visualization”. Computational and Mathematical Organization Theory. doi:10.1007/s10588-018-09280-3
16. Tandoc, E. C. (2019). The facts of fake news: A research review. Sociology Compass, e12724. doi:10.1111/soc4
17. Velasco, A. T., Roi, A. Cortez, C., Camay, J. M. B., Michael, I., Giba, C., Diloy, M. A., "Factit: A Fact-Checking Browser Extension," 2023 IEEE 12th International Conference on Educational and Information Technology (ICEIT), Chongqing, China, 2023, pp. 342-347, doi: 10.1109/ICEIT57125.2023.10107833.
18. Vo, N. and Lee, K. (2018). “The Rise of Guardians”. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval-SIGIR ’18. doi:10.1145/3209978.3210037
19. Warman, D., Kabir, M. A. (2023) "COVIDFakeExplainer: An Explainable Machine Learning based Web Application for Detecting COVID-19 Fake News," 2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Nadi, Fiji, 2023, pp. 01-06, doi: 10.1109/CSDE59766.2023.10487649.
20. Zhang, Q., Guo, Z., Zhu, Y., Vijayakumar, P., Castiglione, A., Gupta, B. B. (2023). “A Deep Learning-based Fast Fake News Detection Model for Cyber-Physical Social Services”. Pattern Recognition Letters, Volume 168, Pages 31-38. https://doi.org/10.1016/j.patrec.2023.02.026.