Neural Networks 146 (2022) 36–68

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

ARCNN framework for multimodal infodemic detection

Chahat Raj, Priyanka Meel *

Department of Information Technology, Delhi Technological University, India

ARTICLE INFO

Article history:
Received 17 April 2021
Received in revised form 17 October 2021
Accepted 4 November 2021
Available online 13 November 2021

Keywords:
COVID-19 fake news
Infodemic
Deep learning
Multimodal fusion
Neural networks

ABSTRACT

Fake news and misinformation have adopted various propagation media over time, nowadays spreading predominantly through online social networks. During the ongoing COVID-19 pandemic, false information is affecting human life in many spheres The world needs automated detection technology and efforts are being made to meet this requirement with the use of artificial intelligence. Neural network detection mechanisms are robust and durable and hence are used extensively in fake news detection. Deep learning algorithms demonstrate efficiency when they are provided with a large amount of training data. Given the scarcity of relevant fake news datasets, we built the Coronavirus Infodemic Dataset (CovID), which contains fake news posts and articles related to coronavirus. This paper presents a novel framework, the Allied Recurrent and Convolutional Neural Network (ARCNN), to detect fake news based on two different modalities: text and image. Our approach uses recurrent neural networks (RNNs) and convolutional neural networks (CNNs) and combines both streams to generate a final prediction. We present extensive research on various popular RNN and CNN models and their performance on six coronavirus-specific fake news datasets. To exhaustively analyze performance, we present experimentation performed and results obtained by combining both modalities using early fusion and four types of late fusion techniques. The proposed framework is validated by comparisons with state-of-the-art fake news detection mechanisms, and our models outperform each of them.

1. Introduction

COVID-19 is a potentially fatal pandemic and has raised a breathtaking infodemic associated with it. 'Infodemic' is a term coined by the World Health Organization (WHO) to describe the spread of false news in enormous amounts globally at the time of the coronavirus pandemic. After the 2016 US presidential election, the pandemic has appeared as one of the most significant events of misinformation propagation where each individual on the Internet has been a source or consumer of misinformation (Allcott & Gentzkow, 2017). Fake news is not a technical issue in the media but rather a deliberate human activity (Adiba et al., 2020). News is manipulated several times when it travels through word of mouth (Burkhardt, 2017). Rumors and false information used to spread verbally (unofficially) or through official news media in the older times. Nowadays, the ease of access to technology is allowing misinformation to spread rapidly to a larger mass. Social media networks being a largescale communication platform are subjected to widespread misinformation. Multimedia information propagation on the Internet has progressively increased with the intent of reaching a larger audience. Visuals like images and videos bypass human minds more promptly than long and often dull texts, leave a lasting impact (Meel & Vishwakarma, 2021). Users on social media have varied ideologies and each user perceives information differently depending on several factors including education, personal background, political stand, religious inclination, and demographics (Singh et al., 2020). The information thus can be manipulated several times in the course of reaching people (Ferrara et al., 2020). Social media users with malicious intent are using multimedia as a tool to proliferate false information. Although technological advancement aims to nurture human lives, it also provides challenging regressions, leading to challenges in the field of fake news detection (Figueira & Oliveira, 2017). Information credibility analysis is becoming more complex because authenticating visual information is more complicated than verifying plain text. Such detection tasks are performed using machine learning and deep learning techniques (Shu et al., 2017).

False news can be broadly divided into misinformation (news that people spread unaware of its credibility) and disinformation (false news mainly spread with a defined motive) (Meel & Vishwakarma, 2020). People tend to gain news from social media due to the huge popularity, feasibility of access and low cost of the distribution media, often succumbing to misinformation that causes severe impacts (Narwal, 2018). The COVID-19 pandemic is a topic of concern, and many people are taking up to impart

* Corresponding author.
E-mail addresses: chahatraj58@gmail.com (C. Raj), priyankameel86@gmail.com (P. Meel).

https://doi.org/10.1016/j.neunet.2021.11.006
0893-6080/

information on online channels, either being informed or ill-informed. Credibility analysis and fact-checking of every single piece of information on the Internet is not feasible, given the volume and velocity of the incoming data. While it is of utmost importance to deliver authentic information to the masses, the Internet is flooded with false news that links coronavirus to a wide range of entities (Orso et al., 2020).

The infodemic is transmitted as rapidly as COVID-19 itself, in some cases faster, owing to advanced Internet technology and online social media platforms (Allahverdipour, 2020). This has evolved conspiracy theories, political agendas, fake advisories, and more. Several false claims stating multiple remedies as a cure for the disease have appeared, misleading people into self-medication and unproven treatment procedures (Naeem et al., 2021). A representation of a few examples of fake news related to coronavirus is shown in Fig. 1. These screengrabs were collected from social media platforms, and the information supplied has been verified and declared false or misleading by official fact-checking sites. These examples show fake remedies suggested by people to cure coronavirus. Fig. 1(i) shows a fake claim attributed to WHO that advises people to stop eating bakery items. Various fake claims circulated on the Internet state that an advisory or prevention mechanism has been issued by WHO or various other reputed official organizations. Internet users have claimed cures for the disease, transmitting the information through multimedia usage, spreading the infodemic, both textually and visually. Visuals catch one's attention more promptly and are easier to comprehend, unlike text which requires conceptual understanding. This makes the study of multimodal content critical. We, therefore, design a framework that exploits both textual and visual matter to perform the classification of fake and real news.

1.1. Research objectives

Overwhelmed by the enormous amount of fake news during the pandemic, we were encouraged to design an architecture that discerns misleading information based on its inherent features. The main goal of this work is to establish a unified framework that alleviates fake news detection tasks to help mitigate the infodemic. We propose the ARCNN (Allied Recurrent and Convolutional Neural Network) model to distinguish COVID-19-related fake and real news. Studies relying on multimodal information for online news verification are limited. Existing research is primarily focused on text-based fake news detection utilizing traditional machine learning algorithms. The primary drawback with machine learning algorithms is that they require a manual feature extraction process. The proposed ARCNN overcomes this limitation by incorporating deep learning architectures that automatically learn feature extraction using neurons during the model training phase. Another gap in machine learning algorithms is their inability to mine inherent features within the information. In contrast, our framework has the advantage of recognizing patterns in the data provided, such as identifying the writing style in text or recognizing tampering in images. Also, to handle large volumes of data, deep learning-based ARCNN is more effective where machine learning-based frameworks fail.

Visual data is an essential contributing factor in the spread and detection of fake news. Unlike the detection works used up to now in the infodemic detection domain, our framework uses multimodal features from COVID-19-related discussions on the web and fuses them to generate classification predictions. Inspired by previous multimodal approaches (Ajao et al., 2018; Khattar et al., 2019; Singhal et al., 2019), we utilize RNN and CNN architectures and fine-tune them to precisely fit coronavirus-related texts and images. Our architecture relies upon inherent textual and visual features that the ARCNN model efficiently exploits. This focuses on knowledge-based detection in the text domain, which analyses the writing style differences of fake and real news, and secondly, self-extraction of visual features by CNNs for efficient image classification. The choice of using improved RNN models, namely LSTM and Bi-LSTM, is to mine the advantage of their high data while retaining the capacity for sequential inputs. The proposed RNN sequences in this work can extract useful long-term dependencies in textual data. The proposed LSTM and Bi-LSTM networks can learn textual patterns in fake and real news at a sentence level. The lags between the occurrences of similar patterns are remarkably handled and used for classification by LSTMs. Also, the proposed networks overcome the vanishing gradient problem commonly encountered in traditional RNNs. The RNN stream of the proposed ARCNN serves to detect valuable patterns in fake news by identifying the writing style. The usage of CNNs for image classification is supported by the fact that they can identify inherent features within an image. In addition to being computationally efficient, CNNs can recognize distinctive features in images of different classes. The CNN architectures and optimization in the proposed ARCNN offer high adaptability to any input data. In order to build a multimodal approach that uses both textual and visual information present in an online post, the RNN and CNN models need to be combined. Researchers perform such a combination using two primary techniques: early fusion and late fusion (Atrey et al., 2010). As suggested by their names, the combination is performed at an early stage prior to training the deep learning model in the case of early fusion, whereas, in late fusion, the features extracted are combined after training each of the models separately. Early fusion is performed by concatenating the features obtained from each model. Late fusion employs four techniques: sum, max, average, and weighted average. Their respective mathematical operations support the fusion of features from different models. Early fusion is a complex operation, whereas late fusion is easier to perform (Atrey et al., 2010). However, early fusion requires less computation time because training is performed only once. Late fusion, being relatively more straightforward, has longer training durations. To explore the effects of both fusion mechanisms, we developed two variants of the ARCNN framework. Thus, combining the proposed RNN and CNN pipelines, we present one of the earliest multimodal frameworks for infodemic detection. A summary of the contributions of this work is as follows:

1. We introduced the Coronavirus Infodemic Dataset (CovID), a multimodal dataset consisting of over 3500 real and fake news with text and images.
2. We proposed novel ARCNN architecture that incorporates RNN and CNN models. The RNN stream is experimented with by using LSTM and Bi-LSTM. To experiment with various CNN architectures, we proposed a novel CNN model. We also used four pre-trained CNN models – Visual Geometry Group (VGG)-16, InceptionV3, Xception, and MobileNetV2 – fine-tuning them to achieve high performance for fake news classification.
3. We experimented with five methods to fuse text and image modalities using early and late fusion. The early fusion mechanism in our approach performs simple concatenation of features. The late fusion variant uses average fusion, weighted-average fusion, sum fusion, and max fusion techniques for combining the RNN and CNN models.
4. The performance of the proposed ARCNN model was evaluated by experimenting with multiple combinations of RNN and CNN models on six multimodal COVID-19 fake news datasets including ReCOVery, CoAID, MediaEval 2020, and our proposed CovID.
5. Our work analyzed the percentage contribution of textual and visual features in misinformation detection.

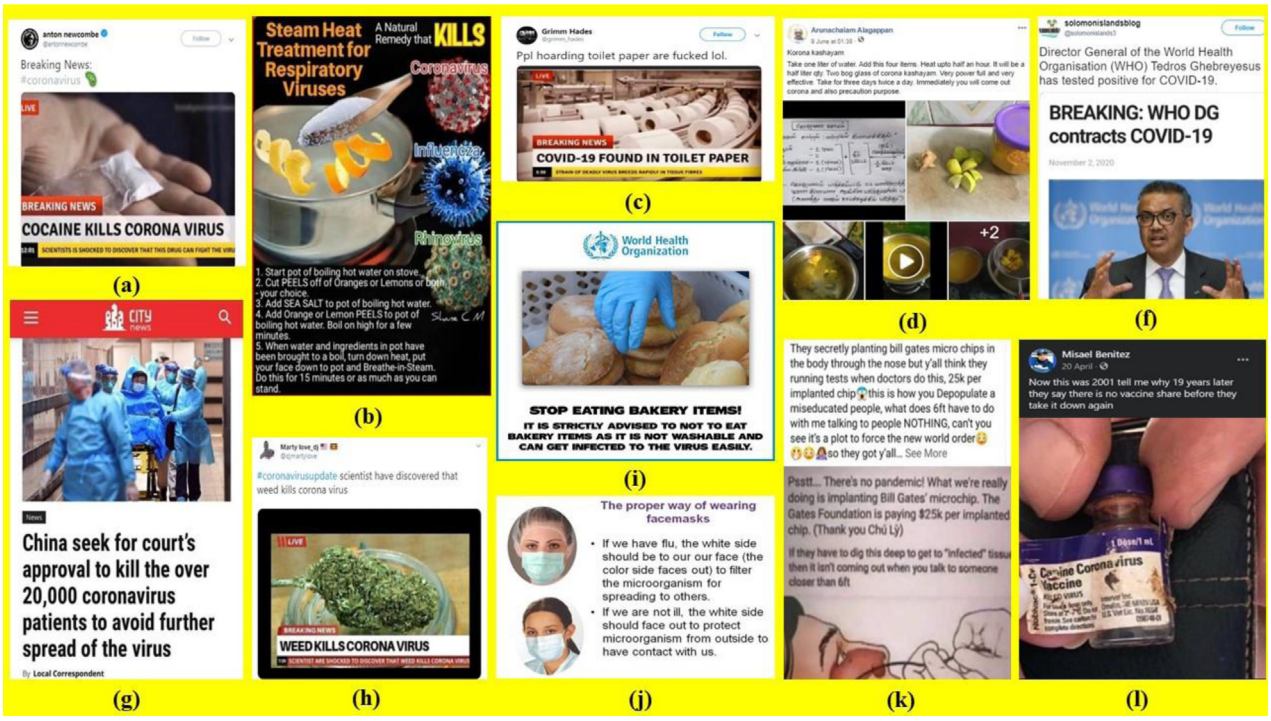

Fig. 1. Examples of fake news related to COVID-19.

6. Evaluation results were presented in terms of a wide range of metrics to present an exhaustive analysis based on accuracy, precision, recall, F1-score, roc-auc score, FPR, specificity, and MCC.

The outcomes of this research follow:

1. Bi-LSTM was a better RNN choice over LSTM for textual fake news detection as it performed marginally better. In visual classification, XceptionNet was the leader with the highest maximum, average, and minimum accuracy, followed by MobileNetV2, VGG-16, InceptionV3, and the proposed CNN model.
2. Weighted-average fusion had the highest accuracy, followed by early fusion, average fusion, sum fusion, and max fusion. Thus, the framework worked best when text and images were assigned a suitable weight while combining the modalities. Although complex, early fusion was the second-best combinatorial method with a reduced runtime.
3. Visuals played a critical role in fake news identification. The weighted-average fusion demonstrated a 30%–50% contribution of images toward infodemic detection.
4. Experimentation on distinctive datasets that contained posts from news articles and social media showed that the corpora constitution played a vital role in infodemic detection, signifying the influence of writing style on detection mechanisms. The proposed framework performed better classification on social media posts than on complex and fairly long written news articles.

The rest of the paper is organized as follows: Section 2 discusses the previous work performed in multimodal fake news detection and ongoing research in the infodemic domain. In Section 3, we describe our proposed dataset, CovID, its features, and the collection procedure. Section 4 presents the proposed methodology and novel ARCNN architectural details. Experimental details and evaluation results are demonstrated in Section 5. Section 6 summarizes our work with the conclusion and prospects.

2. Related work

We discuss the existing literature in three sub-sections. Section 2.1 explores the recent trends in infodemic detection. In Section 2.2, we describe the existing work that performs multimodal fake news detection. Section 2.3 covers various fusion mechanisms used in past literature for such tasks.

2.1. Infodemic detection

The infodemic is amplifying at an alarming rate, and so are the detection methodologies. Researchers have begun experimenting with artificial intelligence algorithms and are readily providing solutions. The aim is to establish a baseline that detects and mitigates fake news as quickly as possible. Recent advances in infodemic detection have been observed in experiments involving text-based analysis and feature extraction. Majumder and Das (2020) developed an LSTM framework with an attention mechanism to detect Twitter users who spread false information. They check suspected users' Twitter accounts for misinformation tweets. Recent work is moving toward multilingual infodemic detection. In Thai texts, NLP-based transfer learning techniques have been used to detect multilingual fake news (Mookdarsanit & Mookdarsanit, 2021). A bilingual (Arabic/English) COVID-19 tweet dataset has been introduced for infodemic detection (Elhadad et al., 2021b). Another work introduces an ensemble-based deep learning model to detect COVID-19 fake news using textual data (Elhadad et al., 2021a). The approach uses feature engineering with various DL algorithms – sequential model, CNN, RNN-LSTM, RNN-GRU, BiRNN-GRU, and RCNN – combining the results using the max voting technique. Another approach using ensemble learning for tweet classification is a two-level approach (Al-Rakhami & Al-Amri, 2020). The classification is based on tweet- and user-level features using classifiers such as Naïve Bayes, Decision Trees, Support Vector Machines, Random Forest, and K-Nearest Neighbors. Al-Ahmad et al. (2021) used bio-inspired algorithms for feature selection using genetic algorithms.

Recent work uses web search results to detect textual web infodemic using a link2vec approach (Shim et al., 2021). Kaliyar et al. (2021) developed an infodemic dataset, FN-COV, and designed a hybrid model combining CNN and RNN for detection. These works demonstrate architectures that detect fake news based only on textual features.

2.2. Multimodal frameworks

Initial infodemic detection tasks have focused only on the linguistic features of COVID-19-related news. Currently, text on the Internet is usually accompanied by graphics and, in the past decade, this has presented a challenging task to which researchers have responded with their respective innovations. Vishwakarma and Jain (2020) presented a survey of state-of-the-art fake news mechanisms. Boididou et al. (2018) neatly summarized various fake news classification methodologies for multimedia on Twitter. A deep study of different modalities used for fake news detection explains the text, image, temporal, and network modalities and their advantages (Anoop et al., 2019). For ease of understanding multimodal systems that utilize semi-supervised and unsupervised machine learning algorithms, (Saini et al., 2020) contributed a descriptive study of state-of-the-art mechanisms. Singh et al. (2021) recently developed an algorithm for multimodal fake news detection that exploits textual and visual features. Khattar et al. (2019) proposed MVAE constituting an encoder, decoder, and fake news detector; with their deep learning framework based on Bi-LSTM and VGG-19 models. Ajao et al. (2018) performed this task using LSTM and hybrid CNNs. Singhal et al. (2019) introduced SpotFake architecture that used BERT and VGG-19 for textual and visual detection, combining these features using concatenation. (Yang et al., 2018) combined textual and visual features with explicit statistical features and trained the data using Bi-LSTM and CNN, combining the features with early fusion. Event Adversarial Neural Networks were introduced by (Wang et al., 2018) to extract event-invariant features, and was proposed for fake news arriving from variant events. Cui et al. (2019) developed the SAME framework that takes input sentiments and user comments and passes them to LSTM and VGG-16 models that use an attention mechanism to generate predictions. A new method performs two types of classification – topic and tweet levels – suggesting that tweets of the same category will fall under the same topic having similar credibility scores and would aid in efficiently classifying similar types of news (Jin et al., 2015) . Shu et al. (2019) also used CNN and VGG-16 and carried out the detection process utilizing user profile features. The record shows that previously designed frameworks have broadly used LSTM and Bi-LSTM for text classification, and VGG has been utilized generally for image classification. Motivated by these frameworks and to expand this research, we experimented with various existing deep CNN architectures.

2.3. Fusion mechanisms

Different fusion mechanisms are used to combine features from different data modalities to design such multimodal detection frameworks. This can be performed in various ways, and each combination has a different effect on the classification results. Two techniques are widely used for multimodal combination: early and late fusion (Atrey et al., 2010). Early fusion is often referred to as feature-level fusion, with features from different data modalities combined at an early stage using an operation. This type of combination is performed prior to training the model. In contrast, late fusion, also referred to as decision-level fusion, depends on the results obtained by each data modality individually. These individual decisions are then combined using a suitable mathematical operation. This entire process is carried out after the training of each deep learning model for different data modalities and is therefore known as late fusion.

To analyze how the proposed method would perform the best, we experimented with early and late fusion techniques to fuse textual and visual features. Our motivation comes from the existing literature. SpotFake trains text and image models separately and combines them using a simple concatenation technique (Singhal et al., 2019). Researchers have experimented with AdaBoost to apply late fusion to their classification model (Maigrot et al., 2016). The MVAE and TI-CNN methods have also been implemented with early fusion that uses a simple concatenation method (Khattar et al., 2019; Yang et al., 2018). Jin et al. (2017) proposed using an attention mechanism for fusing visual features and concatenation for final classification. Lago et al. (2019) applied a weighted late fusion approach that assigns weights w1 and w2 to text and image classification probabilities by calculating their product and later adding them. The details and mathematical background of these fusion mechanisms are explained in detail in Section 4.3. A summary of related works and the classification models they used with their fusion mechanisms is provided in Table 1.

3. Proposed dataset

The infodemic rose at an alarming rate as the pandemic spread over the globe, and given the extreme worries in curbing the COVID-19, fighting an infodemic during global chaos has become quite challenging. There is a scarcity of multimodal infodemic datasets, which is crucial for developing fake news detection systems. Researchers worldwide have responded to understand the complexities and acted promptly to introduce various infodemic datasets and detection methodologies. Kishore Shahi and Nandini (2020) introduced one such repository, FakeCovid, a multilingual collection of fact-checked news across 105 countries. Their dataset motivated us to create CovID, our multimodal dataset for textual and visual fake news detection. Rather than using FakeCovid to extract visual features, we decided upon extracting news articles from scratch. This aided us in building a dataset of a more extensive date range from 4 January to 30 October 2020. Another limitation we encountered in FakeCovid was the biased nature of the dataset with very few numbers of authentic or reliable articles. To overcome the limitations, we proposed building CovID from scratch, extracting real and fake news items along with their visual content.

3.1. Data collection and pre-processing

This section explains the step-by-step approach of dataset preparation and cleaning, and is pictorially represented in Fig. 2. We extracted data from various news sources like news websites, fact-checking websites, and Twitter. To create a balanced dataset, we used the following sources for each label.

Poynter: The Poynter Institute maintains the International Fact-Checking Network with the view of debunking false news across the world. During the infodemic, they have maintained a database of coronavirus-related fact-checked news articles in more than 40 languages from websites of several countries. We started by scraping page URLs of fact-checked articles listed on the Poynter website. Beautiful Soup aided the extraction, a Python library used to crawl elements from web pages. After getting the URLs of all fact-checked articles, we used them to crawl various details in the dataset, the most important being news title, news text, and image URL. We merged various strong and weak categories of false news under the fake label. These categories are false, false context, conspiracy, false headline, inaccurate, incorrect,

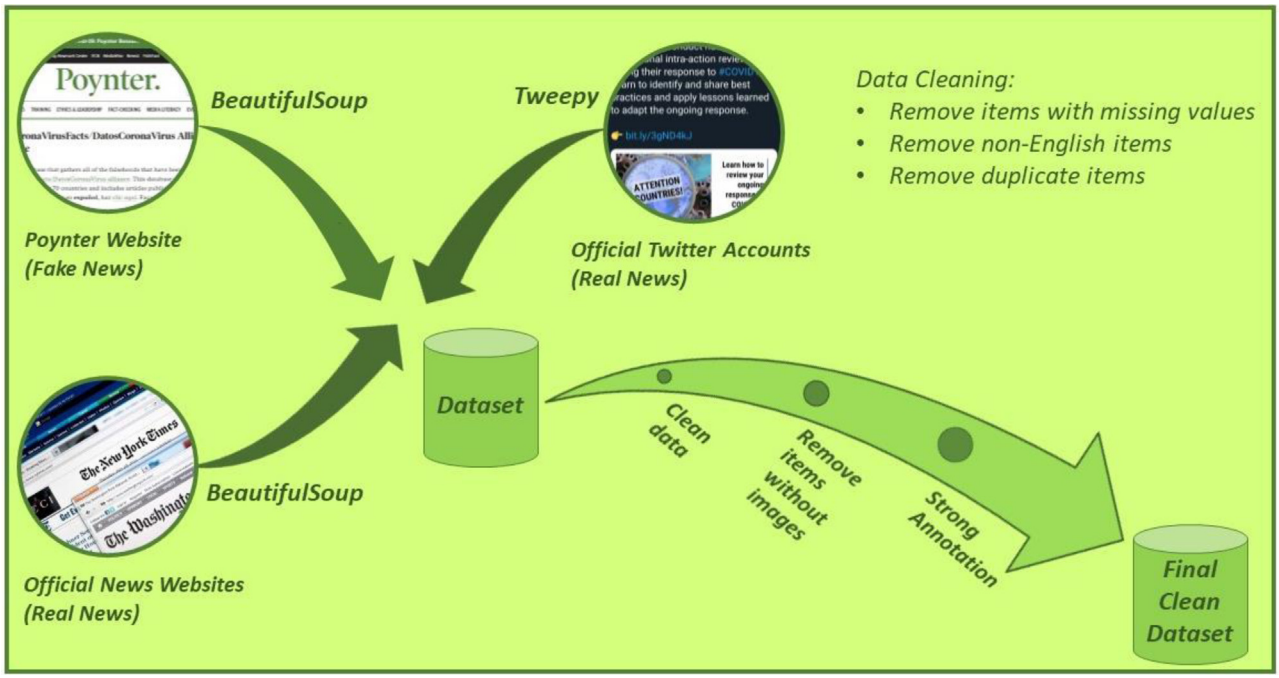

Fig. 2. Data collection and pre-processing workflow.

Table 1
Summary of related multimodal fake news detection tasks listing models and fusion methods used.

Reference	Modality	Method	Model	Fusion method
Khattar et al. (2019)	Text, Image	MVAE	Bi-LSTM, VGG-19	Early fusion
Singhal et al. (2019)	Text, Image	SpotFake	BERT, VGG-19	Concatenation
Yang et al. (2018)	Text, Image, Explicit features	TI-CNN	Bi-LSTM, CNN	Early fusion
Wang et al. (2018)	Text, Image	EANN	Text-CNN, VGG-19	Concatenation
Cui et al. (2019)	Text, Image, Sentiments	SAME	LSTM, VGG-16	Early fusion
Krishnamurthy et al. (2018)	Text, Video, Audio	MLP	Text-CNN, 3D-CNN	Concatenation, Hadamard + Concatenation
Lago et al. (2019)	Text, Image	NLP + Forensics	Sentiment, Similarity, Frequency, CNN	Late fusion (weighted average)
Jin et al. (2017)	Text, Image	Att-RNN	LSTM, VGG-19	Late fusion

mainly false, misleading, primarily false, pants on fire, partially false, and partly false. The false information debunked in the fact-checking articles is a mix of social media posts, contributed mainly by Facebook and Twitter users, and malicious websites posting false claims. This set builds up our data under the "False" label category with news titles, text, and image links.

Official news websites: Deep learning algorithms learn on training data to be able to distinguish between classes. This generates the need for well-classified data under different labels. We received a meager count of true articles from the Poynter website, and accordingly we shifted to collecting true articles from official sources of news. We created a list of official news websites that are providing trustworthy news during the times of COVID-19. The extraction process was the same as for extracting false news articles. We used each website and collected news articles linked with coronavirus. The keywords used were "COVID-19", "COVID", and "coronavirus". We obtained a collection of true news titles, text, and image URLs.

By examining the collection thus obtained, we determined that data under the false label was a mix of false news articles from websites and social media posts. In contrast, the data under the true category contained only official news articles. Knowledge-based detection, which we proposed to apply in our framework, works upon the writing style of the text. It focuses on how sentences are structured, and words are linked together. Paying heed to this minute detail of the technique, we perceived that official news was structured formally and in a well-defined way compared to social media posts with inconsistent writing styles. To provide bias-free detection, we decided to contrast our collected false news with an unbiased mix of news articles and social media posts. To proceed in this direction, we extracted social media's true news from Twitter. These contrasting datasets assisted us in inspecting the effect of corpora in fake news detection tasks. We call these two versions of our dataset CovID I and CovID II.

Twitter: The extraction process was supported by Twitter REST API using Tweepy to extract historical tweets. We shortlisted

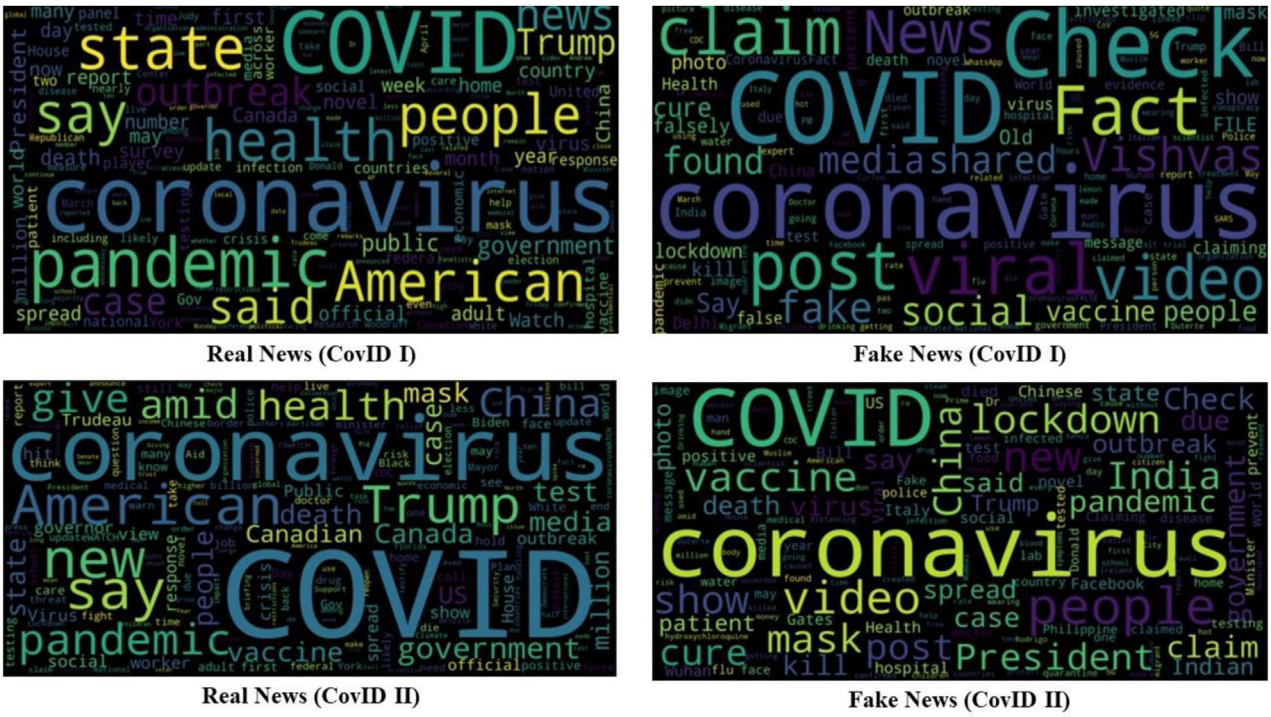

Fig. 3. Word clouds of real and fake news from CovID I and CovID II.

official Twitter users who provided authentic information during the times of COVID and fetched all of their tweets dated from to 1 January to 30 October 2020. The extraction process provided us with multiple information, of which we used tweet texts and image URLs.

Pre-processing: The pre-processing steps involve many data cleaning steps to filter out unwanted data. First, we removed all multilingual data, making our repository solely English news. We then removed all news items that did not contain visual information. For multimodal detection, we kept only news articles with accompanying images. We then removed duplicate items and any rows containing missing values. After performing a content analysis of the remaining data, we strongly labeled our dataset with manual annotation. The items were weakly labeled as true or false depending on their extraction sources. For final confirmation of their classification, two annotators went through each item in the dataset and provided a strong label based on a mutual decision concerning the items. The annotation was supported by inter-coder reliability where the annotators verified the labels by authenticating the headlines, text, and images. To stay updated with the continuously evolving scientific knowledge about COVID-19, the datasets were regularly fact-checked and verified. Word clouds for real and fake classes of both the proposed datasets are shown in Fig. 3.

4. Proposed methodology

We envisioned the fake news detection task as a combination of text and image classification. Deep learning is widely used and proving effective in such tasks. We proposed the ARCNN, using an RNN model for text classification and CNN for image classification. We introduced two variants of the ARCNN model, which differ in how the text and image modalities are combined for compelling predictions. The workflow is illustrated in Fig. 4. The architecture diagram for ARCNN is presented in Fig. 5, depicting early and late fusion variants of the proposed ARCNN architecture.

4.1. RNN component

The selection of RNNs for text classification is based on their advantage of having a memory base. Unlike simple neural networks, the input of the current layer intakes the output of the previous layer, forming a connection that remembers previous sequences and helps predict the next step. All the hidden layers in an RNN can be merged as one recurrent layer. The RNNs have proved to be extremely important as they can process information of arbitrary length and remember this information throughout the recurrent states. Despite such capability of traditional RNNs, their use involves the vanishing gradient problem. This issue arises because the RNN allocates a deeper memory for recent input signals than for the previous ones (Xiao et al., 2018). The problem is resolved using backpropagation through a particular type of RNN known as Long-Short-Term Memory Networks (LSTMs), as shown in Fig. 6. The top horizontal line in Fig. 6 is known as the "cell state" responsible for storing and removing information. The LSTM networks incorporate a gate mechanism by using an input gate (i_t), output gate (o_t), and a forget gate (f_t). The gated structure of this new RNN overcomes the problem of traditional RNNs. These gates perform pointwise multiplication to process the input information.

For inputs given as $x_{t-1}, x_t, x_{t+1}, \ldots, x_n$, the current state in an LSTM is calculated as $h_t = f(h_{t-1}, x_t)$, where h_{t-1} is the previous state and h_t is the current state. The forget gate (f_t) selectively chooses which information must be transferred to the following cell states. It is mathematically represented as follows:

$$\boldsymbol{f_t} = \sigma(\boldsymbol{W_f} \cdot [\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_f}) \tag{1}$$

where σ denotes the activation function and W_f and b_f represent the weight and bias, respectively, at a given time t at the forget gate layer. Next, the input gate is responsible for deciding which information is to be stored in the cell state given by Eq. (2):

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

where W_i and b_i are the weight and bias for the input gate, respectively.

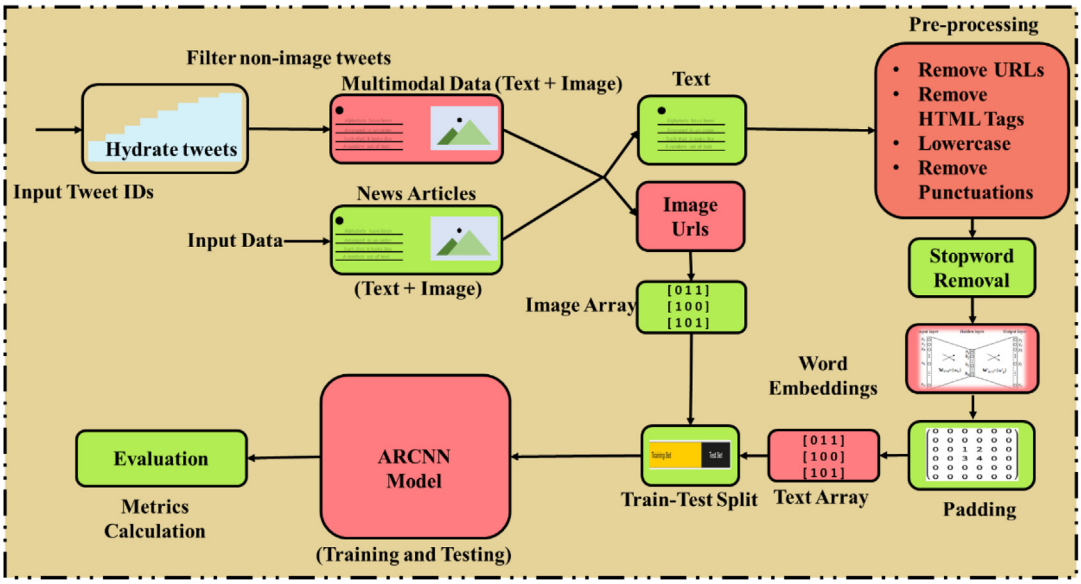

Fig. 4. Workflow of the proposed methodology.

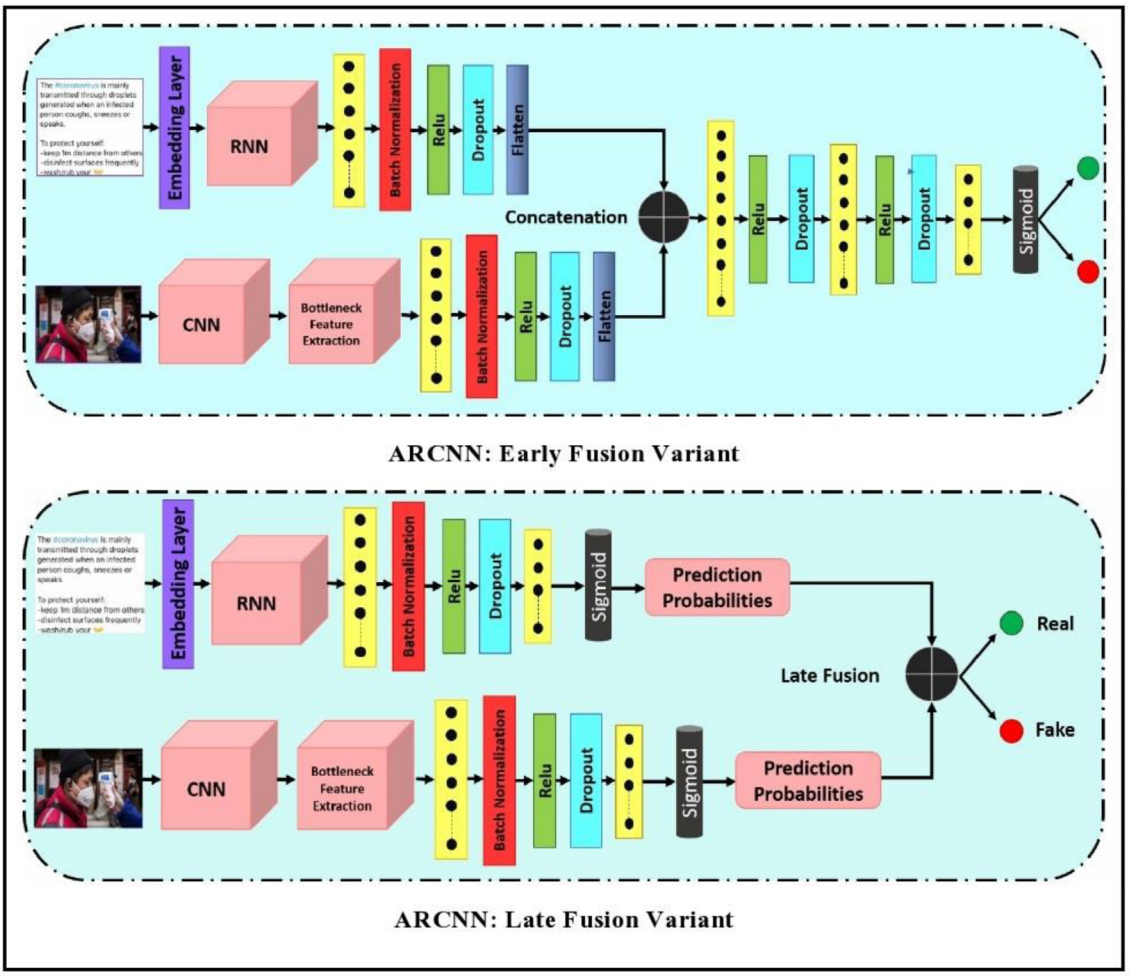

Fig. 5. ARCNN architecture diagram.

The activation function, the sigmoid function produces a vector $\hat{C}_t$, defined by Eq. (3):

$$\hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

The previous cell C_{t-1} is updated to C_t using Eq. (4):

$$C_t = f_t * C_{t-1} + C_{t-1} + i_t * \hat{C}_t \quad (4)$$

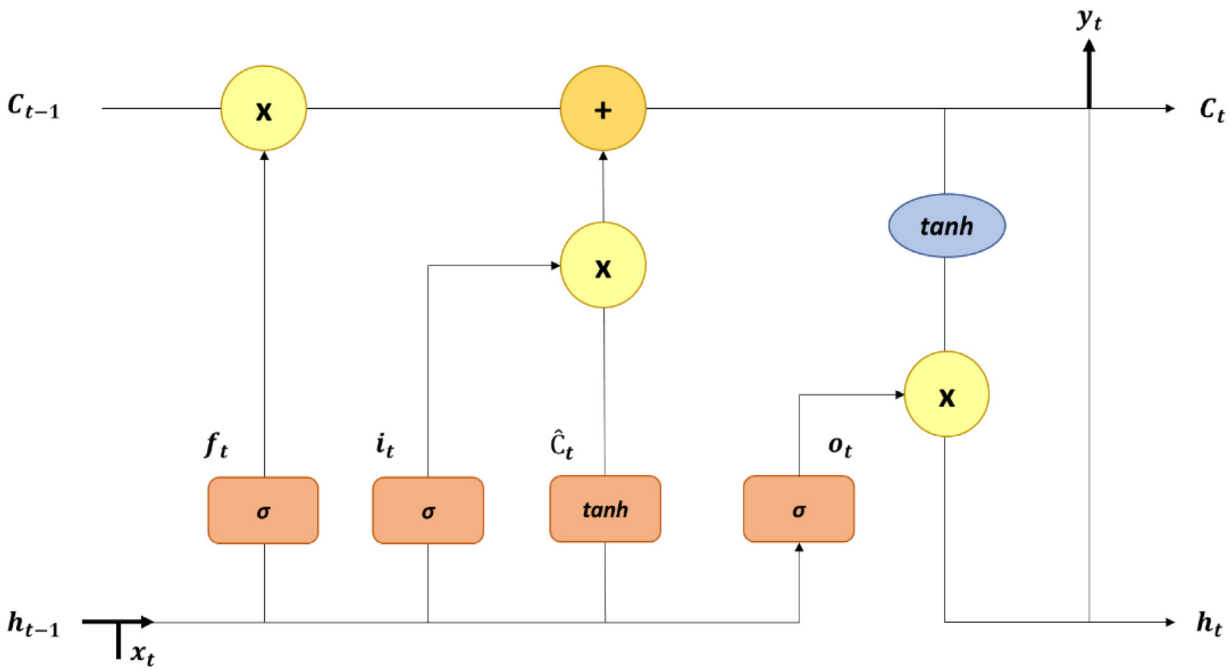

Fig. 6. Architectural representation of LSTM.

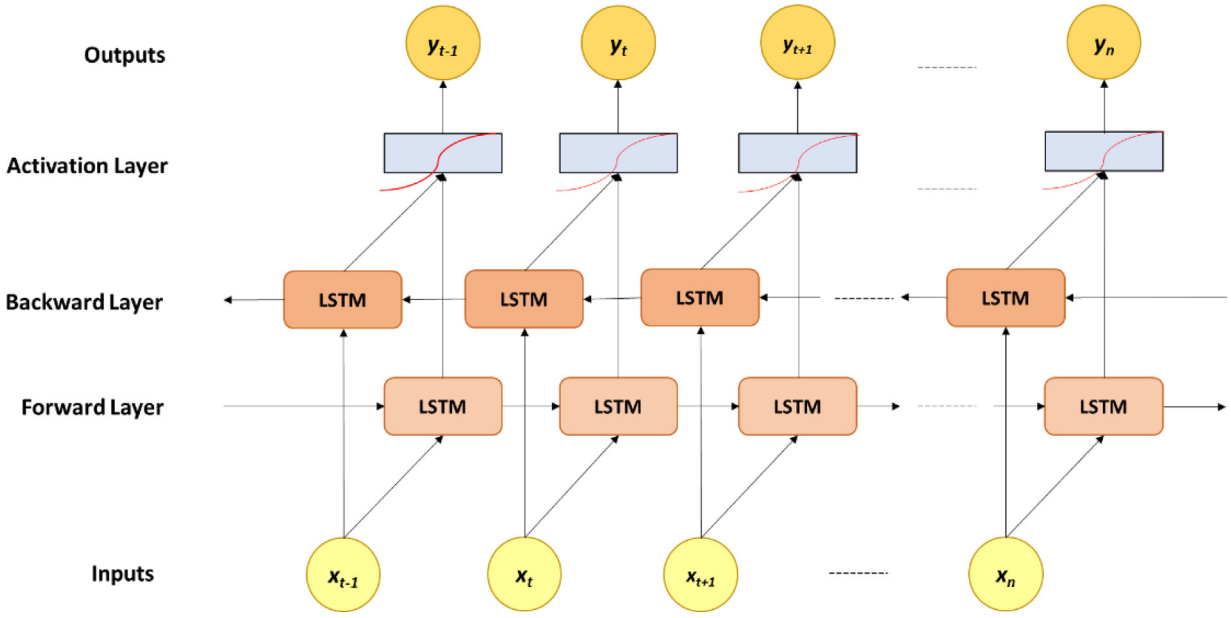

Fig. 7. Architectural representation of bidirectional LSTM.

The final cell state is responsible for providing output o_t of the network, defined as follows:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

where W_o and b_o are the weight and bias at the output layer, respectively.

A bidirectional LSTM (Bi-LSTM) network adds to the advantage of a simple LSTM network. While LSTM is unidirectional and can store only past information in its cell states, a Bi-LSTM directs information forward and backward. Its architecture is illustrated in Fig. 7. For a given sequence of inputs $x_{t-1}, x_t, x_{t+1}, \ldots, x_n$, the output from the forward layer $\vec{h}$ s calculated, whereas for a reverse sequence, $x_n, x_{n-1}, x_{n-2}, \ldots, x_{t-1}$, the output $\overleftarrow{h}$ is calculated through the backward layer where $h = o_t * tanh(C_t)$. The output of the Bi-LSTM network is denoted as:

$$Y_T = y_{t-1}, y_t, \ldots, y_{t+n} \tag{6}$$

where $y_t = \sigma(\vec{h}, \overleftarrow{h})$ and σ is a concatenation operation.

Text input provided to the proposed ARCNN goes to an embedding layer, after which it is fed to an RNN model, an LSTM or a Bi-LSTM model, followed by a series of fully connected layers as illustrated in Fig. 5. The first in the series is a dense layer, after which a batch normalization layer stabilizes the input. We use the ReLU activation function, given by Eq. (7):

$$ReLU = \begin{cases} 0, if x < 0, \\ x, if x \geq 0. \end{cases} \tag{7}$$

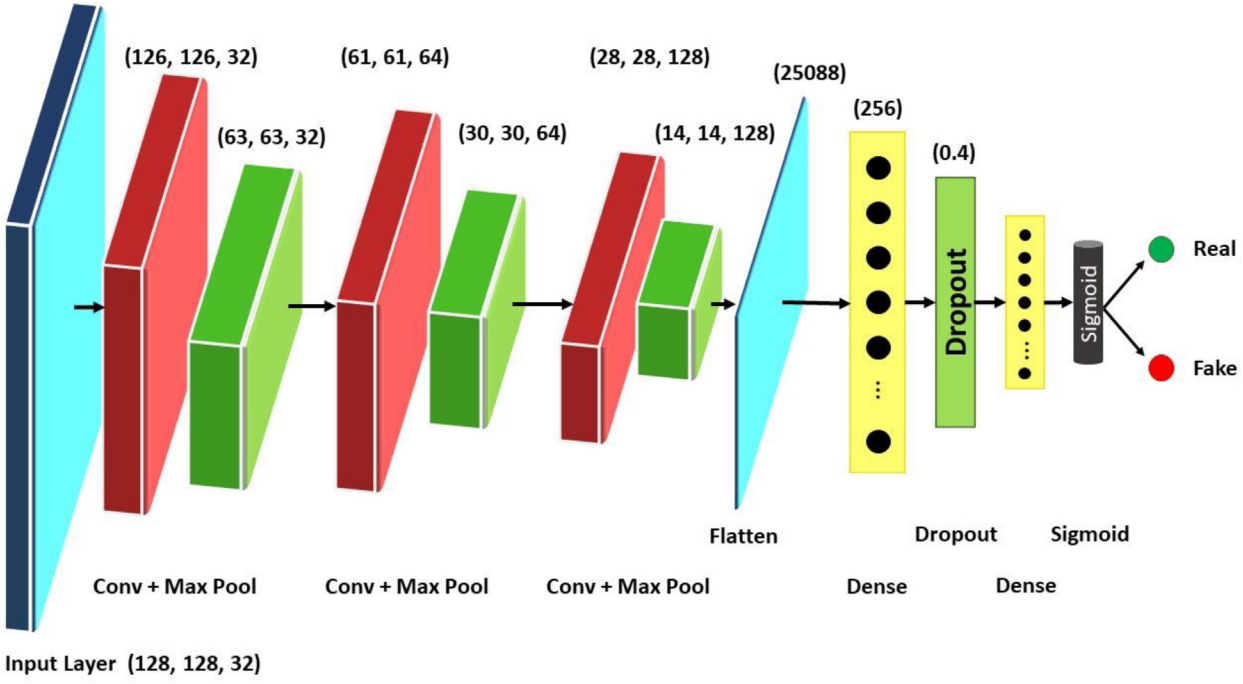

Fig. 8. Proposed CNN architecture.

Table 2
Information of each layer in the proposed RNN architecture.

Layer	Input	Output	Parameters
Embedding	(None, 300)	(None, 300, 50)	50 000
LSTM/Bi-LSTM	(None, 300, 50)	(None, 128)	58 880
Dense	(None, 128)	(None, 256)	33 024
ReLU	(None, 256)	(None, 256)	0
Dropout	(None, 256)	(None, 256)	0
Dense	(None, 256)	(None, 1)	257
ReLU	(None, 1)	(None, 1)	0

A dropout layer is used to prevent overfitting. The output thus received is flattened for dimensionality reduction. Table 2 demonstrates the input, output, and parametric information of the proposed RNN component.

4.2. CNN component

For image classification, CNN architectures have shown outstanding performance in multiple domains. They are among the most popular deep architectures due to their advantage of extracting and learning implicit visual features without much preprocessing. The CNNs are capable of understanding the spatial and temporal dependencies in an image, which aids in better classification. The CNNs are a type of neural network that performs a "convolution" operation on the input data. A convolutional operation $*$ on functions f and g is given by the following formula:

$$(f * g)(t) \triangleq \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau \tag{8}$$

where the product of functions f and g is calculated by reversing and shifting one of these functions. The network consists of an input layer, an output layer, and multiple hidden layers. Each convolutional layer takes as its parameters the kernel size, stride, and zero padding. Convolutions work with a series of pooling and fully connected layers. Feature extraction is performed using convolutional and pooling layers, where pooling layers are responsible for input dimensionality reduction. The proposed architecture uses a max-pooling operation. Fully connected layers perform the classification using a sigmoid function as the appropriate activation function. We used four pre-trained CNN architectures – VGG-16, InceptionV3, XceptionNet, and MobileNetV2 – to fine-tune them to achieve the best performance. We also proposed a simple self-designed CNN model to compare its performance with pre-existing pre-trained models. The image input is fed to a CNN model, after which bottleneck feature extraction is performed. Parameter tuning in a CNN is performed similarly to a RNN by adding dense, batch normalization, ReLU, and dropout layers, as illustrated in Fig. 5. Output from the image sequence is also sent to a flatten layer. From both text and image sequences, we receive flattened outputs of the same dimension. These flattened outputs are then used to fuse the features as per the desired fusion mechanism.

The proposed CNN model's architecture is represented in Fig. 8. This additional model was designed to study the effect of a new convolutional model for the task, comparing the results using pre-trained models. We eliminated the separate bottleneck feature extraction stage used with other pre-trained models (Fig. 5) and let the CNN do this itself. In early fusion, the layers from the CNN to dropout in the image pipeline were replaced by this proposed CNN model, and the next flatten layer in ARCNN stayed in place. To avoid redundancy, the flatten layer in the proposed CNN model was removed, and layers only up to the dropout layer were added. For late fusion, this CNN architecture replaced layers starting from the CNN block to the sigmoid layer. The construct of this model contained three convolutional layers, each followed by a max pool operational layer, further flattening the feature vectors followed by fully connected layers. A dense layer was appended with which a dropout with a probability of 0.4 was used. Table 3 shows the input, output, and parametric information of the model. Proposed CNN architecture is scalable and easily reproducible.

4.3. Fusion mechanisms

An algorithmic explanation of early and late fusion variants of the ARCNN model is described by Algorithms 1 and 2. In the early

Table 3
Information of each layer in the proposed CNN architecture.

Layer	Input	Output	Parameters
Conv2D	(128, 128, 3)	(126, 126, 32)	896
MaxPooling2D	(126, 126, 32)	(63, 63, 32)	0
Conv2D	(63, 63, 32)	(61, 61, 64)	18 496
MaxPooling2D	(61, 61, 64)	(30, 30, 64)	0
Conv2D	(30, 30, 64)	(28, 28, 128)	73 856
MaxPooling2D	(28, 28, 128)	(14, 14, 128)	0
Flatten	(14, 14, 128)	25 088	0
Dense	25 088	256	6 422 784
Dropout	256	256	0
Dense	256	1	257

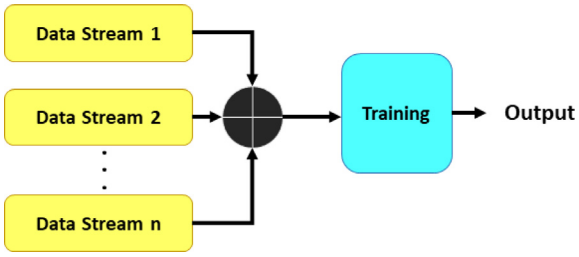

Fig. 9. Early fusion systematic flow.

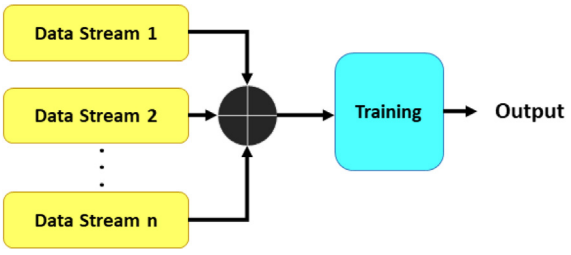

Fig. 10. Late fusion systematic flow.

fusion variant, the outputs from flattened layers are joined using simple concatenation. The next phase involves the addition of a series of dense layers with dropout and ReLU activation functions. Classification is supported by the sigmoid activation function for binary classification. In the late fusion variant, the initial phase is similar to early fusion, where text and image data are passed through RNN and CNN layers, and a sequence of fully connected layers is added, including dense, batch normalization, ReLU, and dropout layers. Instead of flattening the outputs herein, they are led to a sigmoid layer for individual training and generation of prediction probabilities. The classification results are obtained in the form of probabilistic values for each modality, and then combined using late fusion techniques. The fusion methodologies used in the ARCNN architecture are discussed below.

Early fusion: In multimodal frameworks, fusing multimedia modalities is a challenging task. Early fusion, also known as data-level fusion or fusion in feature space, combines features extracted from different data streams before training the model. Data from different streams are of different dimensions. These are to be scaled or normalized at a fixed dimension for all types of data. We performed this using a flatten layer that brings features to the same scale. Feature vectors V_t and V_i from different data streams were integrated into a single large vector V_c. This combined vector handled all multimodal features and performed a one-time training. The combination of vectors was carried out by an operation between V_t and V_i, which was a simple concatenation operation in our case. The following operation represents this:

$$V_c = V_t \oplus V_i \tag{9}$$

where $\oplus$ is the operator between the two. Early fusion is an advantageous approach as it learns features in a collaborative environment as a unified representation of data streams. No separate training phases were required for each data stream. Features from all data streams were combined, and then a single training phase was carried out. This makes the process faster and efficient. Fig. 9 shows the flow of the early fusion process.

Late fusion: Also known as decision-level fusion, late fusion is performed later based on the classification decisions from all data streams. Late fusion is easier to perform and provides a simple and scalable architecture. Learning of features is performed before integration, whereas, in early fusion, features are combined first and then passed for training. Each data stream of different modalities is fed to a training model, and decisions are extracted in terms of prediction probabilities. These prediction vectors are then combined using a suitable combinatorial operation. Fig. 10 represents the flow of the late fusion process. We used decision-level scores from text and image streams in the proposed work and fused them accordingly.

The fusion function f that fuses decisions of text and image streams is denoted by $f: P_t, P_i \rightarrow P_c$, where P_t and P_i are two different feature maps that denote the decisions of each stream in probabilistic values. The combined probabilities denoted by P_c give the output decisions after late fusion. The late fusion scores thus obtained are denoted as P_{av} (average), P_m (maximum), P_s (sum), and P_w (weighted average).

Average fusion: This combines modalities by taking a simple average of prediction vectors. Mathematically, it is represented as follows:

$$P_{av} = (P_t + P_i)/2 \tag{10}$$

where combined prediction P_{av} is calculated by averaging, i.e. summing up values from all streams and then dividing by the number of data streams. Combining only text and image features, the number of data streams is 2, which is used to divide the sum of P_t and P_i.

Max fusion: This technique uses the maximum value of probability, i.e. prioritizing the decision with a larger weight or value than the other to select the higher contributing score between the feature maps. This is performed by a simple maximum function denoted as follows:

$$P_m = max(P_t, P_i) \tag{11}$$

Sum fusion: This sums up the values of feature maps obtained from both data streams simply by adding up the values. It is expressed as follows:

$$P_s = P_t + P_i. \tag{12}$$

Weighted-average fusion: In this fusion mechanism, random weights w_t and w_t are assigned to feature maps from both

Algorithm 1: Early fusion with ARCNN

Input: $A = \{a_1, a_2, \ldots a_n\}$ is set of text vectors, $B = \{b_1, b_2, \ldots b_n\}$ is set of images of size 128*128, $Y = \{y_1, y_2, \ldots y_n\}$ is a set of labels for A and B.

1. Split A, B, and Y into three subsets as $\{(A_1, B_1, Y_1), (A_2, B_2, Y_2), (A_3, B_3, Y_3)\}$ for 60% training, 20% validation, and 20% testing.
2. **For** $i = 1$ to 3, **do**
3. Add respective RNN and CNN models $M_1$and M_2.
4. Extract bottleneck features from M_2 for image input B.
5. Append series of fully connected layers (dense, batch normalization, relu) to $M_1$and M_2.
6. Apply dropouts with 0.5 probability to $M_1$and M_2.
7. Flatten both text and image feature vectors thus obtained, V_t and V_i, to make them unidimensional.
8. Combine V_t and V_i using concatenation and obtain a combined feature vector, $V_c = V_t \oplus V_i$.
9. Add fully connected layers (dense, batch normalization, relu) to V_c setting dropout values as 0.4.
10. Apply binary sigmoid classifier and calculate final prediction probabilities P_f.
11. Calculate the performance of the testing set.
12. **end for**
13. **Return** performance on the testing set.

Algorithm 2: Late fusion with ARCNN

Input: $A = \{a_1, a_2, \ldots a_n\}$ is a set of text vectors, $B = \{b_1, b_2, \ldots b_n\}$ is a set of images of size 128*128, $Y = \{y_1, y_2, \ldots y_n\}$ is a set of labels for A and B.

1. Split A, B, and Y into three subsets as $\{(A_1, B_1, Y_1), (A_2, B_2, Y_2), (A_3, B_3, Y_3)\}$ for 60% training, 20% validation, and 20% testing.
2. **For** $i = 1$ to 3, **do**
3. Add respective RNN and CNN models $M_1$and M_2.
4. Extract bottleneck features from M_2 for image input B.
5. Append series of fully connected layers (dense, batch normalization, relu) to $M_1$and M_2.
6. Apply dropouts with 0.5 probability to $M_1$and M_2.
7. Add binary sigmoid classifier to both M_1 and M_2 individually and obtain independent prediction probabilities P_t and P_i on testing set.
8. Combine P_t and P_i using late fusion operations and obtain combined prediction probabilities, $P_c = P_t \odot P_i$.

   Late fusion operations: $P_c = P_{av} = (P_t + P_i)/2$

   $P_c = P_s = P_t + P_i$

   $P_c = P_m = max(P_t, P_i)$

   $P_c = P_w = P_t * w_t + P_i * w_i$
9. Calculate the performances on the testing set.
10. **end for**
11. **Return** performances on the testing set.

streams. This has an advantage over the other methods as it helps decide which data type contributes to better detection. Playing with the values of assigned weights provides a route to experimentation to decide which weights would make the model best performing. Mathematically, the arbitrary weights, ranging from 0.0 to 1.0, each weight complementing the other, are multiplied by their respective prediction probability arrays and then summed up. It is defined as follows:

$$P_w = (P_t * w_t + P_i * w_i) \tag{13}$$

where w_t and w_i are weights assigned for text and image streams, respectively.

5. Experimental result analysis

This section is divided into five sub-sections. Beginning with detailing the fake news datasets used in this work in Section 5.1, we walk through the implementation and evaluation settings, parametric details, experimental process on various combinations of models, and fusion methods to the evaluation metrics used for task evaluation in Section 5.2. Section 5.3 presents the results in a tabular form. Detailed analysis and comparison of results, elaborating multiple insights, are provided in Section 5.4. In Section 5.5, we report the results of the ablation study. Toward the end of Section 5.6, we compare our method with existing unimodal and

Table 4
Details of datasets used.

Dataset	Referred to as	Type	Real news count	Fake news count	Total
CovID I	D1	Articles, posts	1059	1310	2369
CovID II	D2	Tweets, posts	1171	1303	2474
ReCOVery (Zhou et al., 2020)	D3	Articles	1345	651	1996
ReCOVery (Zhou et al., 2020)	D4	Tweets	3968	924	4892
CoAID (Cui & Lee, 2020)	D5	Tweets	565	517	1082
MediaEval (Pogorelov et al., 2020)	D6	Tweets	791	289	1080

multimodal baselines in terms of accuracy score and highlight the validation of the proposed work.

5.1. Datasets

For the performance evaluation of the proposed ARCNN framework, we used six multimodal datasets that included news articles and tweets containing real and fake information. Among these, we created two datasets as described in Section 3, two subsets of the ReCOVery dataset containing a collection of news articles and associated tweets, a CoAID dataset with health-related tweets, and a MediaEval 2020 benchmark dataset. We evaluated the two subsets of the ReCOVery dataset separately to analyze the effect of corpora on the performance of our model. Detailed information on these datasets is provided in Table 4.

5.1.1. CovID I

We introduced this dataset[1] in lieu of the urgent need for infodemic datasets to meet the requirement of deep learning algorithms. This is a multimodal dataset consisting of textual and visual information of fake news related to coronavirus. CovID I consists of fake and real news articles from websites and social media posts. This dataset is referred to as D1 in the results section. The sources of fake news are various fact-checking websites registered with Poynter IFCN. True news has been extracted from official news website articles.

5.1.2. CovID II

This dataset[2] is proposed to assist the study of the effect of corpora on the proposed ARCNN model. Fake news originates in both social media posts and fake news articles. As our detection is primarily knowledge-based, the writing styles of text have an impact on the detection. Distinguishing text based on how a sentence is written, its formation, vocabulary, and grammar play a significant role in our task. There are differences in the ways social media posts are written from the way official news is structured. It is essential to differentiate between true and real social media posts since both follow a writing style different from news articles. We use a mix of true articles for this detection, primarily extracted from Twitter posts and a few from news articles, to create an unbiased set with a mix of fake posts and articles. This dataset is referred to as D2.

5.1.3. ReCOVery

Zhou et al. (2020) introduced this multimodal repository[3] consisting of 2029 news articles on COVID-19 collected during January–May 2020 containing textual, visual, temporal, and network information. There are also 140 820 tweets related to these news articles added to the dataset. We utilized these news articles and tweet IDs as separate datasets, hereafter referred to as D3 and D4, respectively, for textual–visual detection. Only items containing both textual and visual information were used, and the rest were discarded.

5.1.4. CoAID

Cui and Lee (2020) proposed a COVID-19 Healthcare Misinformation Dataset[4], a repository of health-related fake news spread via news websites and on Twitter. The dataset contains news article titles, user tweets, and associated user interactions, i.e. tweet replies. Since the image URLs were not available for news articles, we utilized only the tweet IDs available in the dataset to extract multimodal information. We were finally left with 565 real and 517 fake tweets containing both textual and visual content.

5.1.5. MediaEval 2020

The onset of the COVID-19 pandemic coincided with the release of 5G technology, which gave rise to a distinct conspiracy that claimed that the arrival of COVID-19 was due to the masts of 5G networks. This led to a violent situation of destroying 5G poles in the UK. MediaEval 2020 issued a benchmark dataset[5] for fake news detection, which is a collection of misinformation related to 5G-linked coronavirus conspiracies, other COVID conspiracies, and non-conspiracy tweets (Pogorelov et al., 2020). We categorized all the conspiracy tweets within a single label of fake news and used non-conspiracy tweets as real tweets.

5.2. Implementation settings

All experiments were performed on Google Colab which provides up to 13.53 free RAM and 12 GB NVIDIA Tesla K80 GPU. The proposed framework was built and implemented in Python 3 using Keras deep learning framework. Input data were split into 60% training, 20% validation, and 20% testing. All models were trained with binary cross-entropy for 15 epochs with a batch size of 64. In the late fusion variant of ARCNN, where image and text models were trained separately, we used the Adam and RMSprop optimizers for image and text classifiers, respectively. In the early fusion variant, we used the Adam optimizer.

Training was performed on 10 different combinations of RNN and CNN models. Early fusion models were trained and evaluated separately as they followed a different training route to late fusion. Late fusion models on all datasets were run separately, evaluating all late fusion methods on a single training and testing run for each dataset. Thus, for one dataset evaluating one model, we extracted five sets of results (one set belonging to one fusion method), corresponding to each of the 10 combinations of RNN and CNN models. We obtained a total of 50 sets of results for each dataset. A description of model settings is provided in Table 5.

We employed a wide range of evaluation metrics for performance comparison of the proposed framework. The performance scores were listed in F1-measure, accuracy, precision, recall (TPR), FPR, ROC, specificity, and MCC score. These values can be calculated using confusion matrix values as described by the mathematical equations that follow:

$$\boldsymbol{F1\ score} = \frac{\boldsymbol{2TP}}{\boldsymbol{2TP + FP + FN}} \tag{14}$$

[1] https://drive.google.com/file/d/1bjMrvPIgwAXt_nvtmP0vFqEqEtYq_YmS/view?usp=sharing

[2] https://drive.google.com/file/d/1ivBi9T0GoY3vkQiabWEQg6CnPSvkpAh7/view?usp=sharing

[3] https://github.com/apurvamulay/ReCOVery

[4] https://github.com/cuilimeng/CoAID

[5] https://github.com/multimediaeval/2020-Fake-News-Detection-Task

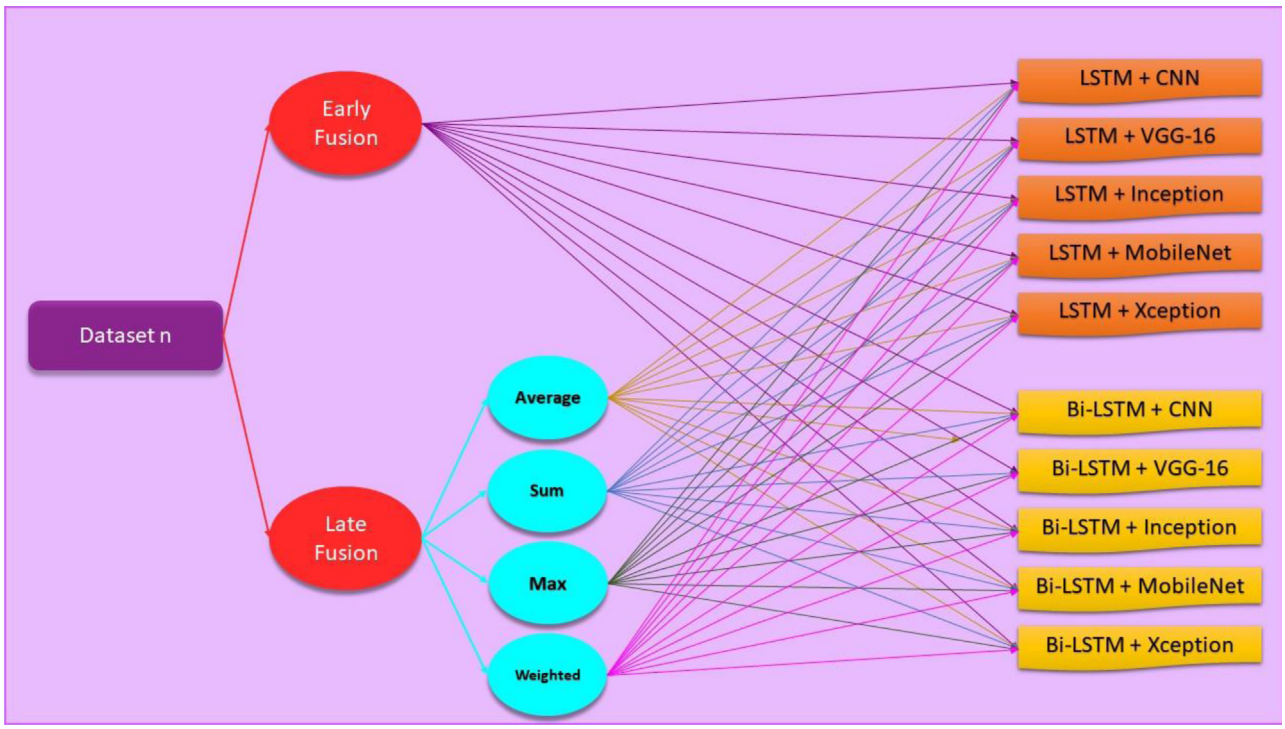

Fig. 11. Various combinations of classification models and fusion methods used for experimentation.

Table 5
RNN and CNN models used for text and image classification and their combinations.

Model	RNN (text)	CNN (image)	Combination
M1		CNN	LSTM + CNN
M2		VGG-16	LSTM + VGG-16
M3	LSTM	InceptionV3	LSTM + InceptionV3
M4		MobileNetV2	LSTM + MobileNetV2
M5		XceptionNet	LSTM + XceptionNet
M6		CNN	Bi-LSTM + CNN
M7		VGG-16	Bi-LSTM + VGG-16
M8	Bi-LSTM	InceptionV3	Bi-LSTM + InceptionV3
M9		MobileNetV2	Bi-LSTM + MobileNetV2
M1		XceptionNet	Bi-LSTM + XceptionNet

$$\textbf{\textit{Accuracy}} = \frac{TP + TN}{P + N} \tag{15}$$

$$\textbf{\textit{Precision}} = \frac{TP}{TP + FP} \tag{16}$$

$$\textbf{\textit{Recall}}\ (\textbf{\textit{True Positive Rate}}) = \frac{TP}{TP + FN} \tag{17}$$

$$\textbf{\textit{False Positive Rate}}\ (FPR) = \frac{FP}{FP + TN} \tag{18}$$

$$\textbf{\textit{Specificity}} = \frac{TN}{FP + TN} \tag{19}$$

$$\textbf{\textit{Mathew's Correlation Coefficient}}\ (MCC) = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}} \tag{20}$$

5.3. Results

This section presents the results obtained by performing all the experiments on six datasets. The consolidated results are presented in Table 6 for each dataset. For each dataset, we used 10 model combinations, M1–M10, and the fusion of each of them was performed in five different ways, also represented in Fig. 11, thus summing up to 50 experiments on each dataset. The results were calculated for eight evaluation metrics: accuracy, F1-score, precision, recall, false-positive rate, ROC, specificity, and MCC score – provided in Appendix. The result analysis is presented in the next section for a clear understanding and the insights gained through these results.

5.4. Result analysis

In this sub-section, we extensively discuss the insights gained from the wide range of experiments performed. All the necessary information is represented graphically and also explained alongside. Graphical representation of accuracy results for 10 classification models used on six datasets is displayed in Figs. 14–19.

From the wide set of results, the highest accuracies obtained in each dataset are represented in Fig. 12. Datasets D2 (CovID II) and D5 (CoAID) demonstrated their top accuracy values as 100%. This shows that all the news items in the testing sets were classified correctly by models M1 (LSTM + CNN) and M10 (Bi-LSTM + Xception), respectively. Both of these datasets included a majority of social media posts, D2 with a collection of posts from various online social networks, and D5 with a collection of tweets. The remaining datasets also achieved good classification accuracies with different models. Comparing the F1 scores for each of the six datasets (Fig. 13) showed that D2 and D5 obtained

Table 6
Accuracy percentage of proposed ARCNN on six datasets.

	Fusion mechanisms	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
D1	Early fusion	83.43	89.43	86.29	90.29	86.57	82.86	83.43	80.29	92.86	86.57
	Avg fusion	82.96	88.86	83.14	92.00	88.57	87.48	88.29	82.29	87.14	86.29
	Max fusion	75.32	82.29	76.86	81.43	80.29	78.18	84.57	80.86	83.43	83.43
	Sum fusion	75.09	82.29	76.86	81.43	80.29	77.22	84.57	80.86	83.43	83.43
	Weighted avg	88.85	89.43	84.86	92.00	88.86	90.05	89.43	87.71	89.43	88.29
D2	Early fusion	100.0	93.55	99.46	96.51	98.66	62.54	96.51	77.69	88.98	98.92
	Avg fusion	99.51	86.29	87.63	90.59	91.94	99.51	85.22	83.06	88.98	87.63
	Max fusion	83.83	84.95	86.83	89.52	90.86	83.09	86.56	87.10	90.86	89.78
	Sum fusion	83.83	84.95	86.83	89.52	90.86	83.09	86.56	87.10	90.86	89.78
	Weighted avg	99.76	98.39	98.12	98.39	98.39	100.0	100.0	100.0	100.0	100.0
D3	Early fusion	80.12	76.66	79.25	77.81	78.67	81.66	75.50	80.98	77.10	77.52
	Avg fusion	80.12	79.48	74.28	73.70	79.48	81.66	75.79	74.64	78.96	77.81
	Max fusion	75.68	78.90	73.99	73.99	73.99	77.03	78.39	75.22	79.54	78.96
	Sum fusion	75.87	78.61	73.99	73.99	73.99	76.83	78.39	75.22	79.54	78.96
	Weighted avg	80.12	83.82	73.7	76.88	79.77	81.66	78.10	74.64	80.98	79.54
D4	Early fusion	82.31	91.72	92.43	92.22	92.02	85.17	92.73	91.5	91.91	91.91
	Avg fusion	81.47	91.40	90.07	92.43	90.07	84.64	91.72	86.90	91.61	91.20
	Max fusion	76.35	90.99	90.89	90.89	90.48	81.36	92.23	91.91	92.12	91.81
	Sum fusion	76.39	90.99	90.89	90.89	90.48	80.99	92.23	91.91	92.12	91.81
	Weighted avg	82.31	92.73	92.22	92.43	92.02	85.17	91.72	91.50	91.91	91.91
D5	Early fusion	87.35	80.09	81.02	63.43	85.19	87.35	89.81	88.43	84.72	91.20
	Avg fusion	97.85	80.65	76.96	84.33	78.80	95.38	78.34	78.80	84.79	79.23
	Max fusion	88.00	89.86	88.94	92.63	92.52	83.08	91.71	89.86	92.63	89.86
	Sum fusion	87.69	89.86	88.94	92.63	92.52	82.46	91.71	89.86	92.63	89.86
	Weighted avg	98.46	97.70	97.70	98.62	97.24	97.23	94.93	96.77	95.39	94.93
D6	Early fusion	84.81	80.57	84.36	84.36	84.36	85.44	85.78	85.78	86.73	86.26
	Avg fusion	85.76	67.30	84.36	81.99	84.83	86.71	67.30	83.41	83.89	84.83
	Max fusion	86.08	84.83	84.36	84.36	84.36	86.08	84.36	84.36	84.36	84.83
	Sum fusion	86.08	84.83	84.36	84.36	84.36	86.08	84.36	84.36	84.36	84.83
	Weighted avg	86.71	84.83	85.31	84.83	84.83	86.39	84.83	86.26	84.83	85.78

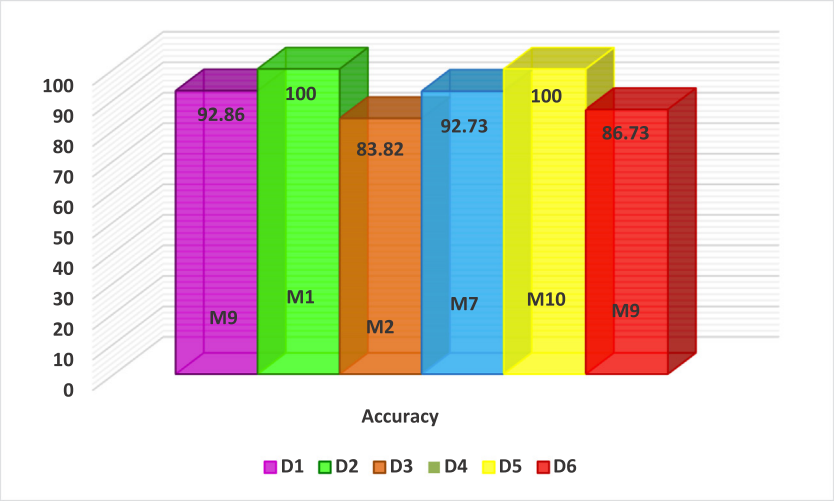

Fig. 12. Highest accuracies obtained in each dataset.

100% and 98.54% scores, representing a balanced dataset. The F1 scores were good when the dataset had balanced items for each category. Slightly lower accuracies or F1 scores resulted with the imbalanced nature of datasets used and model selection. Overall, the proposed architecture provided good results for fake news classification.

5.4.1. Performance comparison on each dataset based on classification model and fusion method

This section analyzes the experimentation results using 10 classification models, each with five fusion methods, and the accuracy trends for these are shown in Figs. 14–19. These trends showed high accuracies when using weighted-average fusion and early fusion. The highest accuracies on all datasets were in the range of 80%–100%. This indicates that the proposed ARCNN model is an effective multimodal classifier. The figures allow selection of the best performing classification models and fusion methods for such tasks. Despite good performance from all models, it is worth noting that models using Bi-LSTM displayed marginally better results. However, both LSTM and Bi-LSTM performed equally well. In terms of image classifiers, VGG-16, MobileNetV2, and proposed CNN models provided the highest results for all datasets. According to experimental observation, VGG-16 took a longer training time than other models. This is a disadvantage for VGG-16, despite providing good results. Other classifiers were comparatively faster with acceptably good classification accuracies.

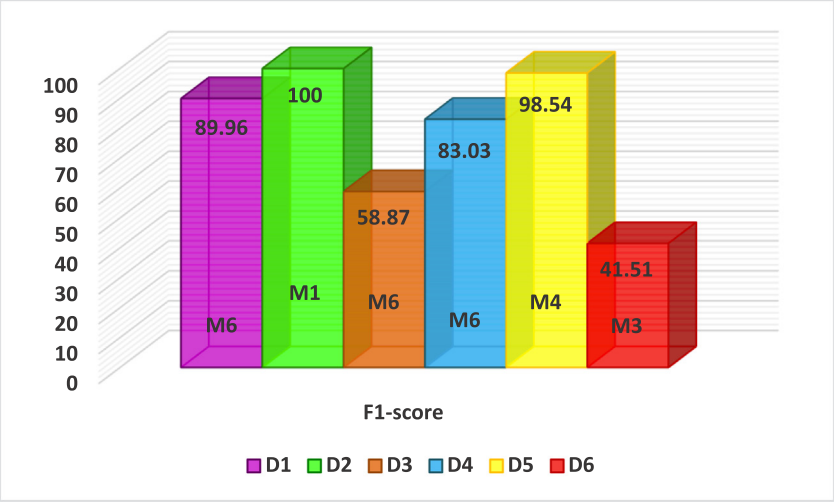

Fig. 13. Highest F1-scores obtained in each dataset.

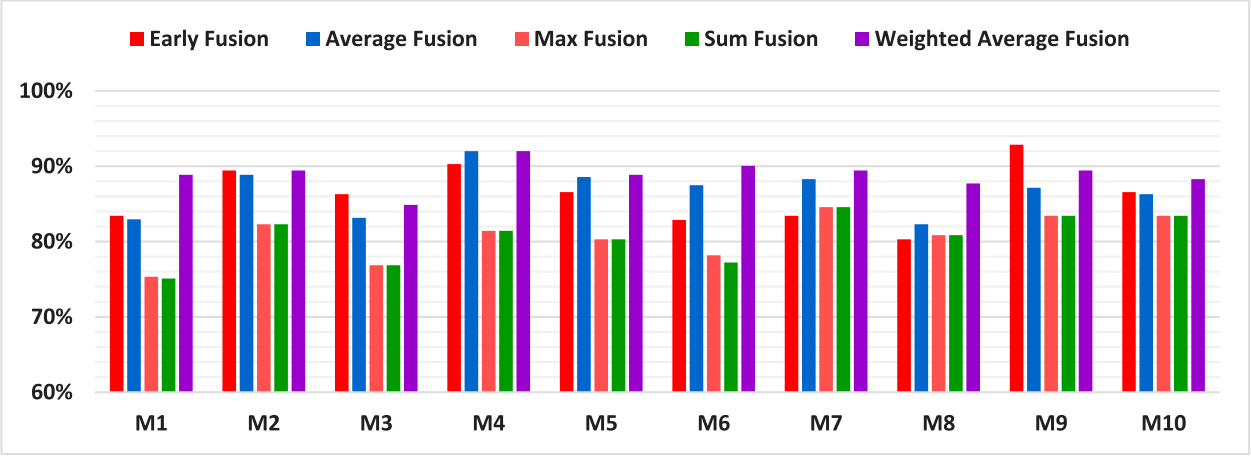

Fig. 14. Performance comparison on D1.

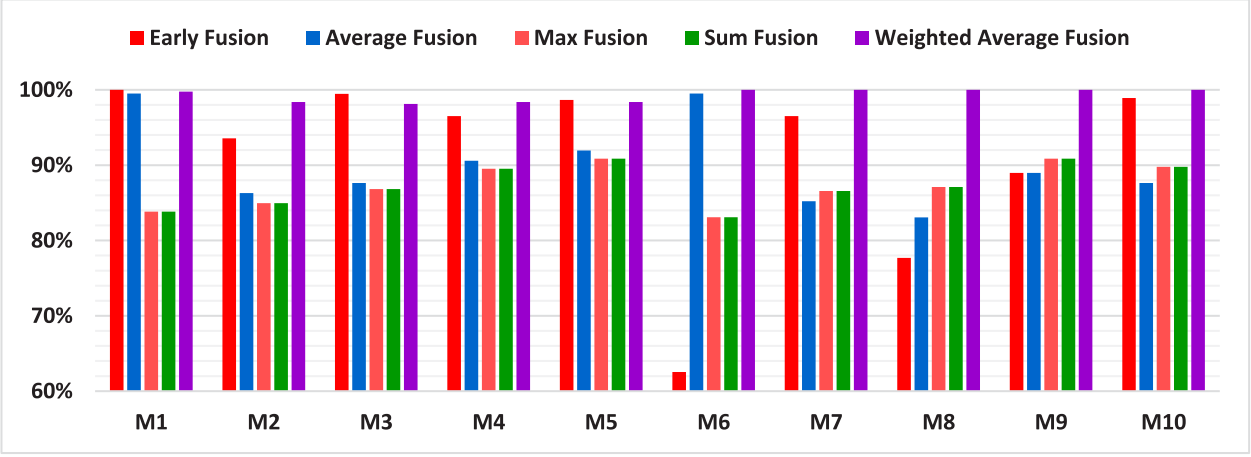

Fig. 15. Performance comparison on D2.

5.4.2. Comparative analysis of classification models on six datasets

Further, we narrowed down our analysis to interpret consistency in the performance of the 10 classification model combinations used in our experiments. The performance graphs are presented in Figs. 20–25. Although model M9 (Bi-LSTM + MobileNetV2) achieved the highest accuracy on dataset D1, model M4 performed more consistently because average accuracy was much closer to the maximum. Model M2, despite delivering more consistent results, fell below the average and the maximum accuracy values of M4. In the D2 dataset, M1 seemed an obvious winner, followed by M5 and M4 models. Dataset D3, flooded with news articles of long and complex texts, posed a credible

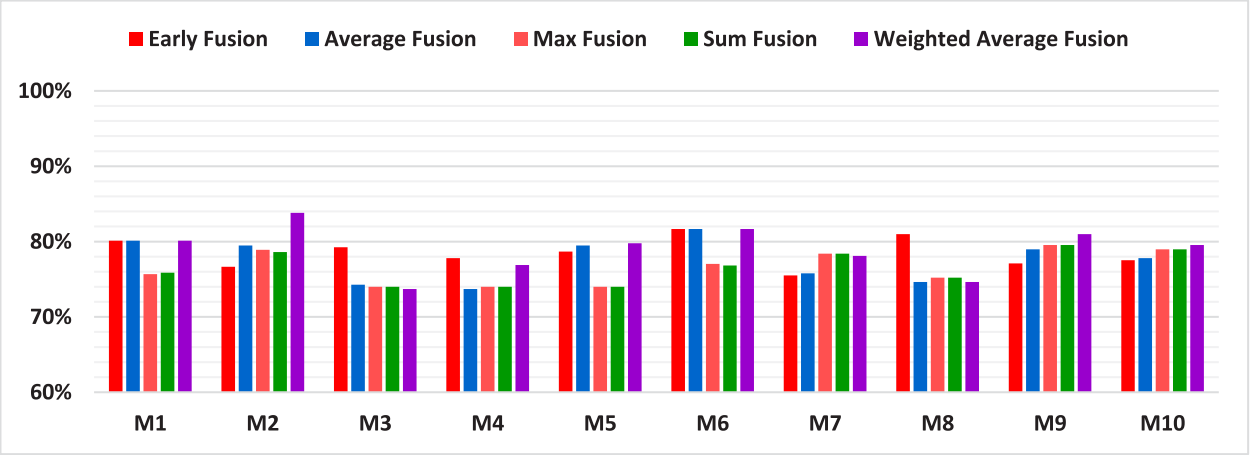

Fig. 16. Performance comparison on D3.

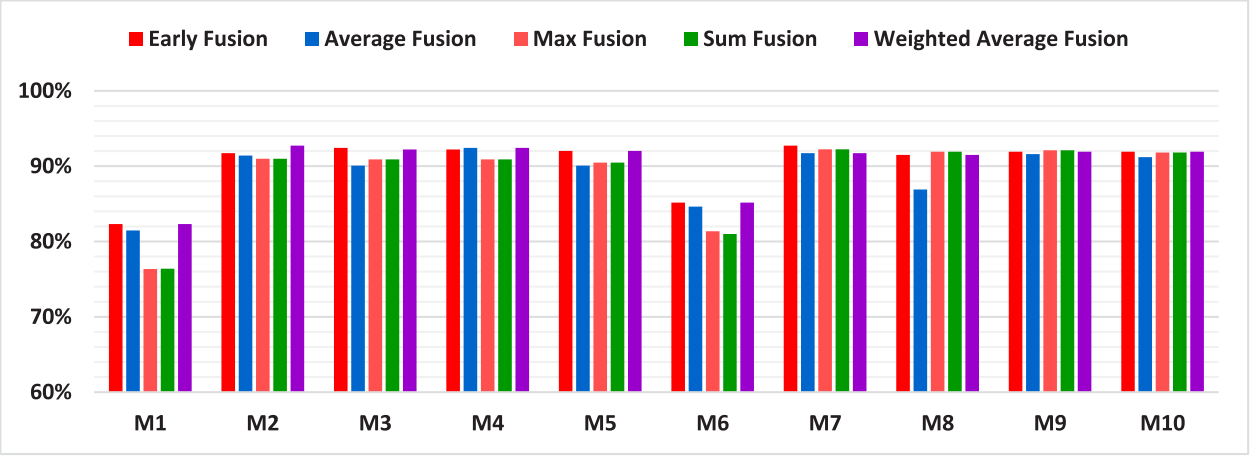

Fig. 17. Performance comparison on D4.

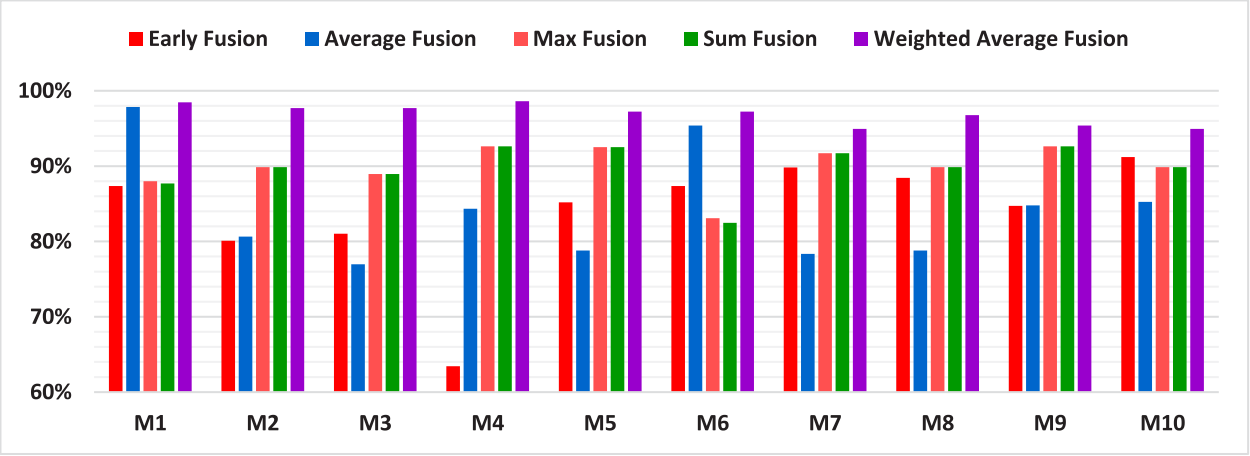

Fig. 18. Performance comparison on D5.

challenge to the performance of all 10 fusion techniques. The maximum and average accuracy were lower than those obtained in all other datasets for most of the fusion methods. Most fusion methods performed equally well in dataset D4, with high repeatability and consistency in accurate results. In dataset D5, the model continuously learned to differentiate between fake and real pieces of information, and hence most of the methods provided highly accurate detection, similar to dataset D2. This can be attributed to the fact that there was a balance in real and fake news count proportions in these datasets. Dataset D6 was highly biased regarding the number of real news counts, and had the most recurring troughs (M2 and M7) in the plot. None of the methods provided significant results (above 90%) as they did with other datasets. The outcomes were indifferent, irrespective

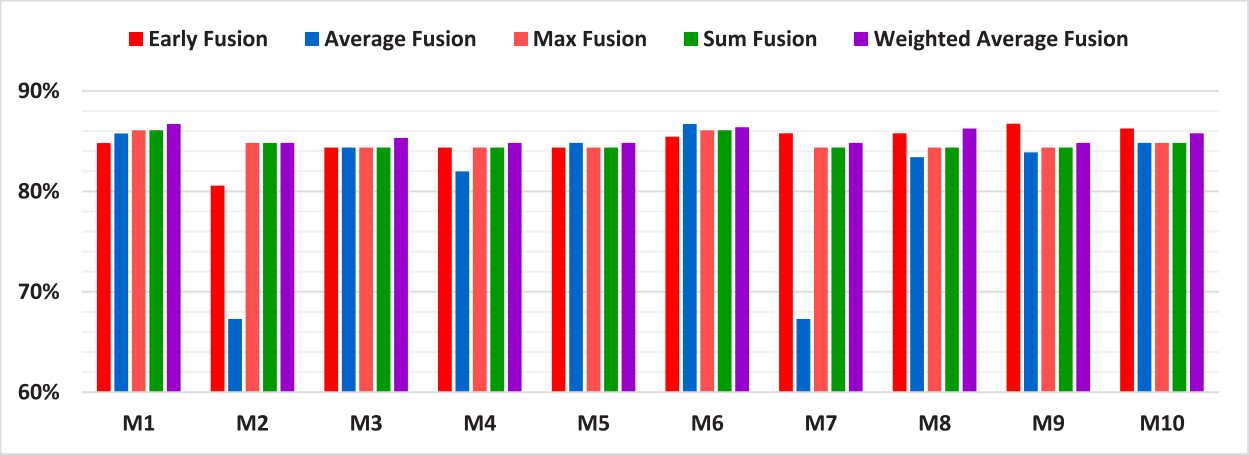

Fig. 19. Performance comparison on D6.

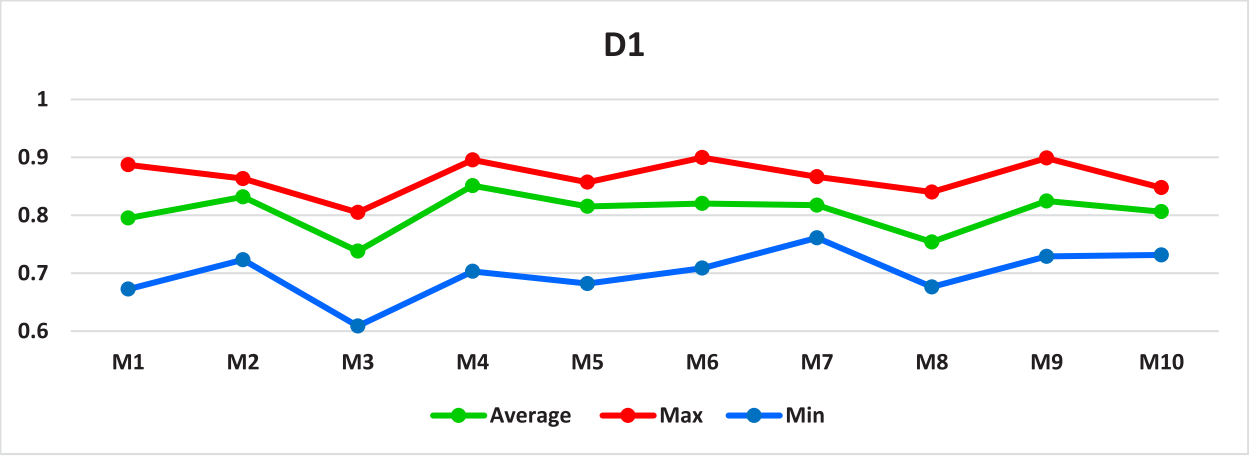

Fig. 20. Comparative analysis of classification models on D1.

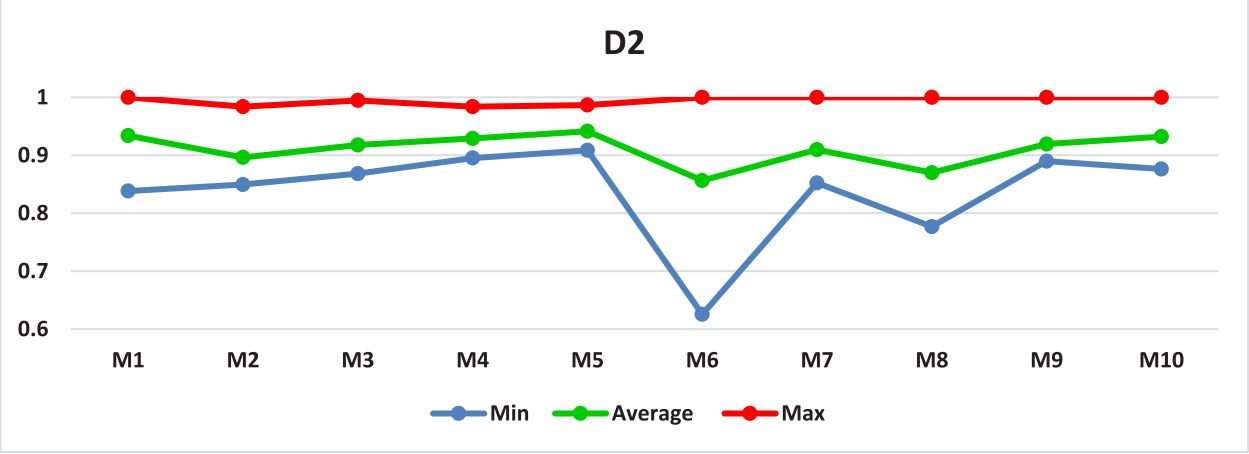

Fig. 21. Comparative analysis of classification models on D2.

of the method chosen. The classification models achieved good maximum and average accuracies for most datasets. Datasets with slightly lower results (∼80%) are attributed to the type of information. Our image classification models performed well overall. Lower than 90% accuracy scores in D3 were due to complex and lengthy texts of articles. Being highly biased, dataset D6 produced accuracy scores close to 85%.

5.4.3. Comparative analysis of classification models on all fusion methods

Another important criterion that determines the accuracy of fake news detection is the fusion technique applied to each model. Comparison graphs are shown in Figs. 26–30. The first of the five techniques applied was early fusion. Models M1, M3, M5, and M10 performed significantly better than others in early

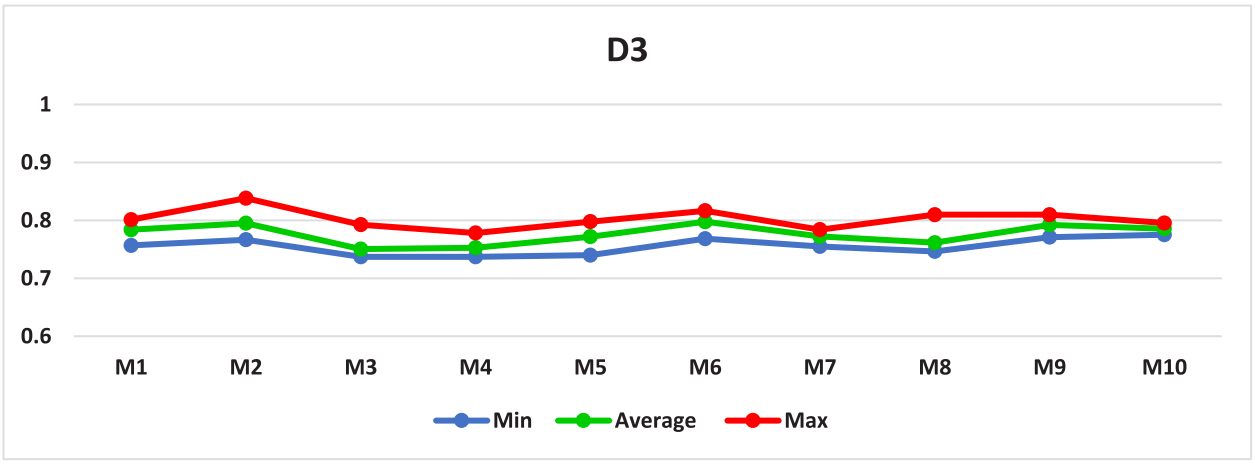

Fig. 22. Comparative analysis of classification models on D3.

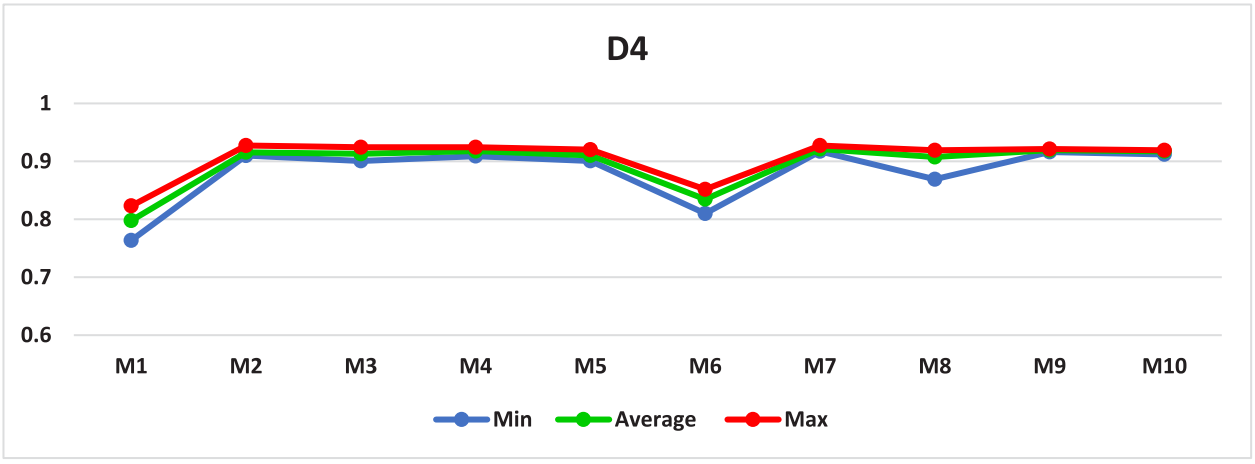

Fig. 23. Comparative analysis of classification models on D4.

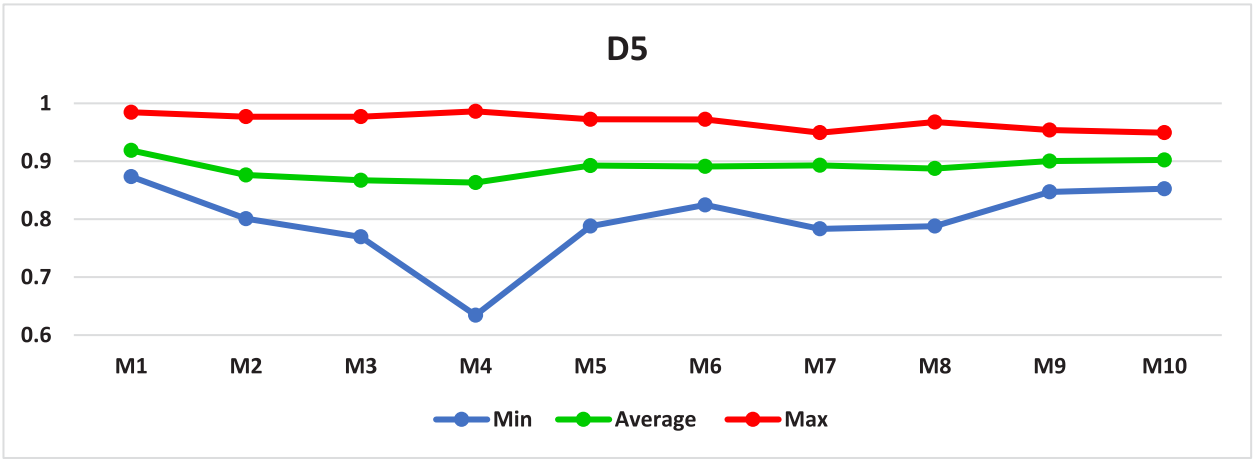

Fig. 24. Comparative analysis of classification models on D5.

fusion. In the remaining four late fusion techniques, M1 and M5 performed consistently better or at par compared to the other eight models. Model M1, where LSTM is used with proposed CNN architecture, produced consistent results over all 10 models and all fusion techniques. Hence, the proposed CNN architecture offered excellent stability in classification on all datasets. Model M5 with LSTM and XceptionNet also produced favorable results under all circumstances.

5.4.4. Comparative analysis of fusion methods on six datasets

This analysis selected the best fusion methods by comparing the performance trends of all fusion methods for a dataset. Fusion-based comparative graphs for each dataset are shown in

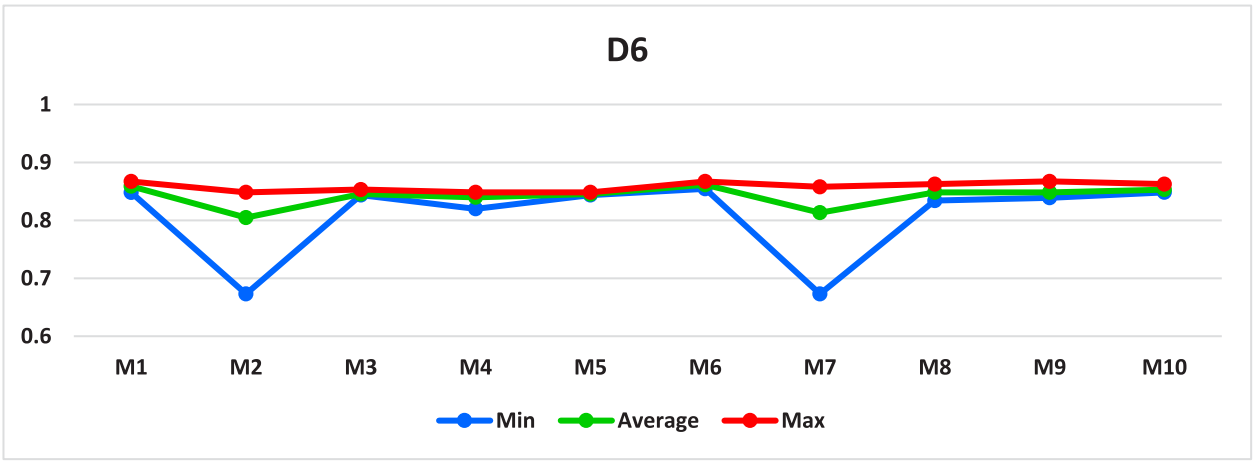

Fig. 25. Comparative analysis of classification models on D6.

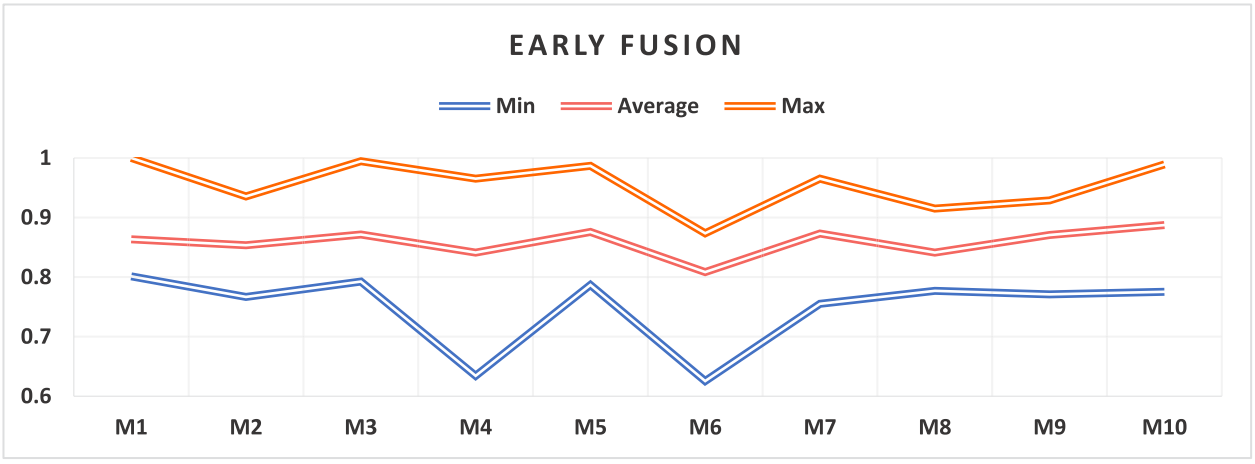

Fig. 26. Comparative analysis of early fusion with all classification methods.

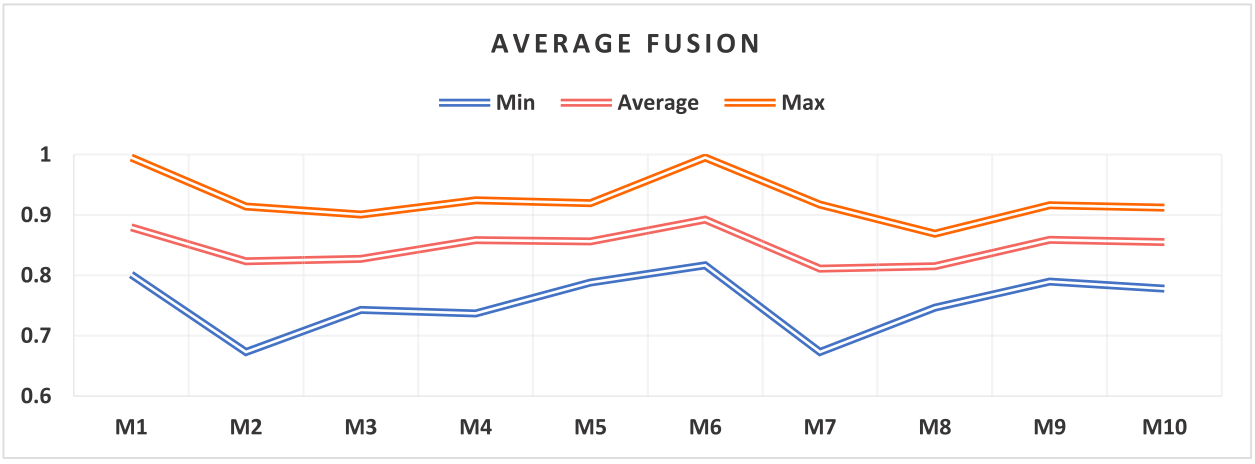

Fig. 27. Comparative analysis of average fusion with all classification methods.

Figs. 31–36. Dataset D1 provided the best maximum and average performance with weighted-average fusion. The next best performers were early and average fusions. Max and sum fusion methods performed worse than all other methods, with their maximum fusion results often lower than average results from other fusion methods. Dataset D2 had the maximum accuracy of 100% for early fusion and weighted-average fusion. Average fusion was third in terms of maximum and average accuracies. Comparing overall performance, weighted-average fusion performed stably with fewer deviations among minimum, average, and maximum results; whereas early fusion performance in D2 showed a considerable gap or deviation among the three. All fusion methods performed similarly for dataset D3. Maximum accuracy values were 80% or greater in each case. Dataset D3

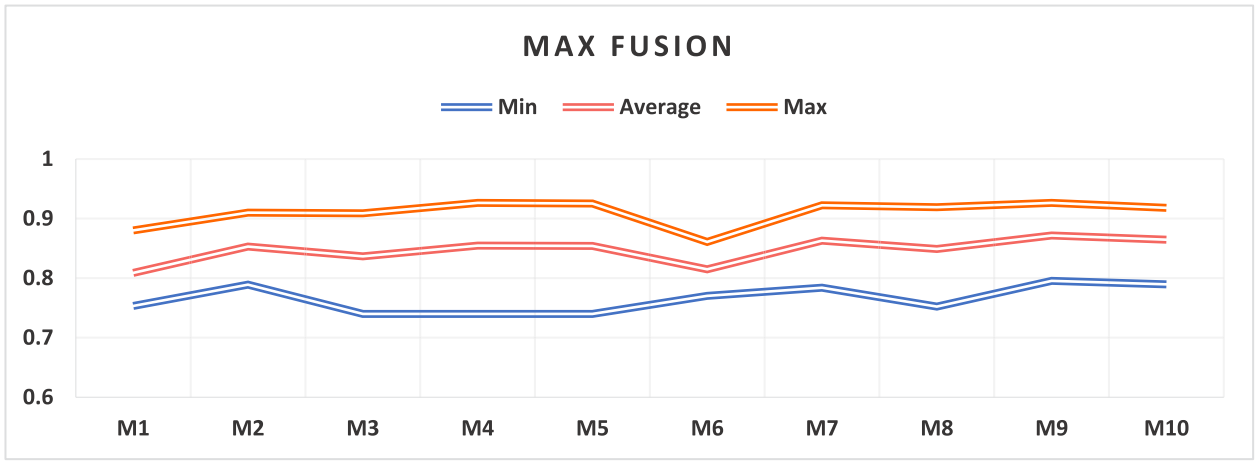

Fig. 28. Comparative analysis of max fusion with all classification methods.

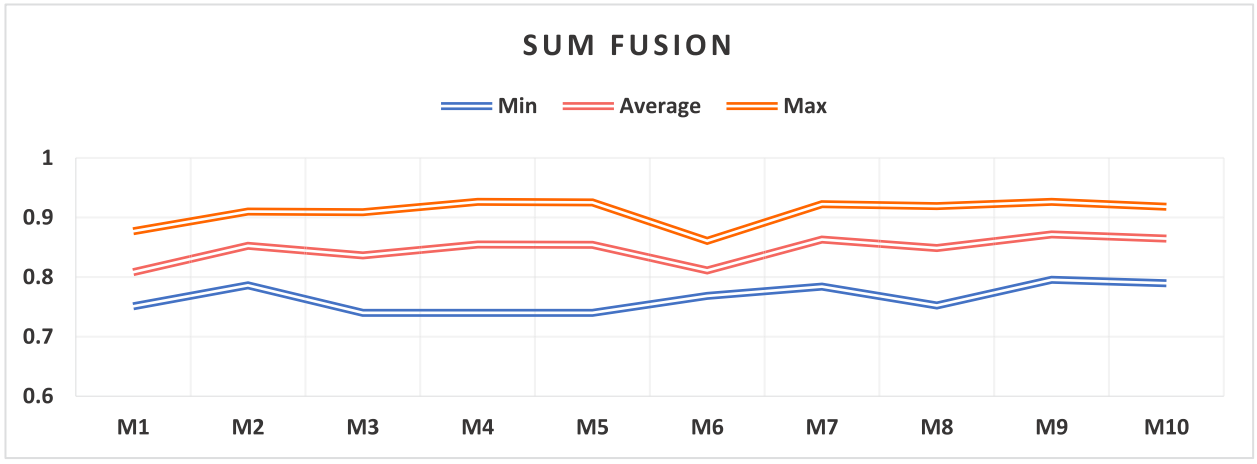

Fig. 29. Comparative analysis of sum fusion with all classification methods.

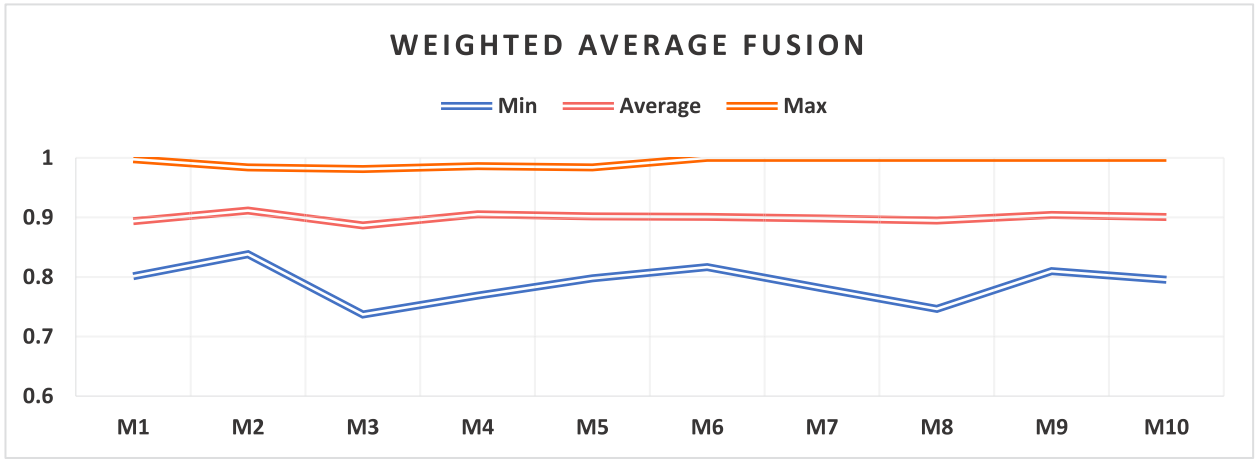

Fig. 30. Comparative analysis of weighted-average fusion with all classification methods.

also had the highest max and average scores and weighted-average fusion was the best. In D4, maximum accuracies for all fusions exceeded 90%, and average values were also close to 90%. Early fusion and weighted-average fusion gave the best results, followed by average fusion. Max and sum fusion methods also showed promising results on this dataset. For dataset D5, weighted-average fusion was best followed by average fusion and early fusion in terms of maximum results, but with more deviation in the latter two. All fusion methods achieved quite similar average performance with dataset D6, but with average fusion fluctuating more. Overall analysis of fusion methods considering all datasets indicates that weighted-average fusion had the highest performance with the least variance. Early fusion was next in terms of performance and lower variance. Average fusion was

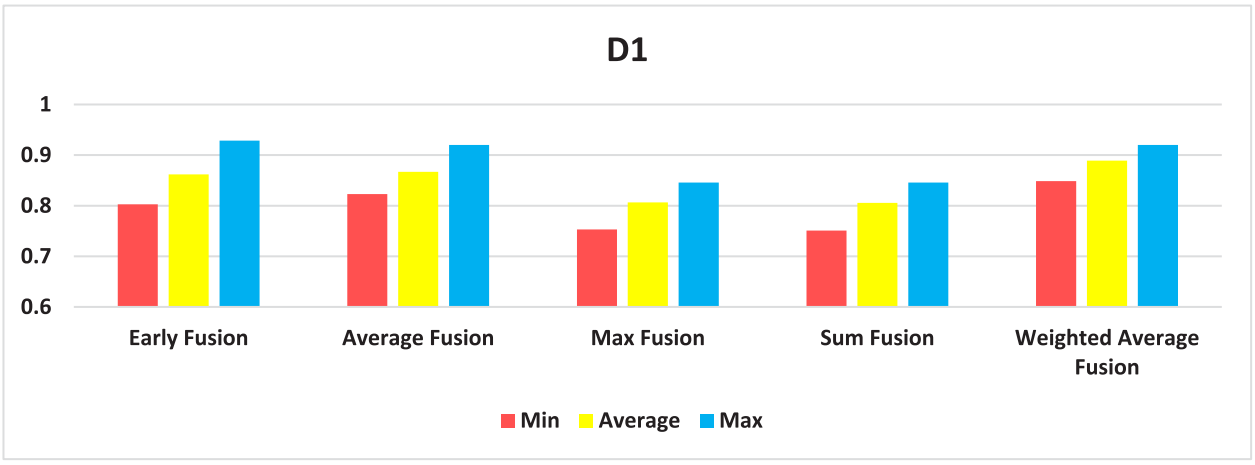

Fig. 31. Comparative analysis of fusion methods on D1.

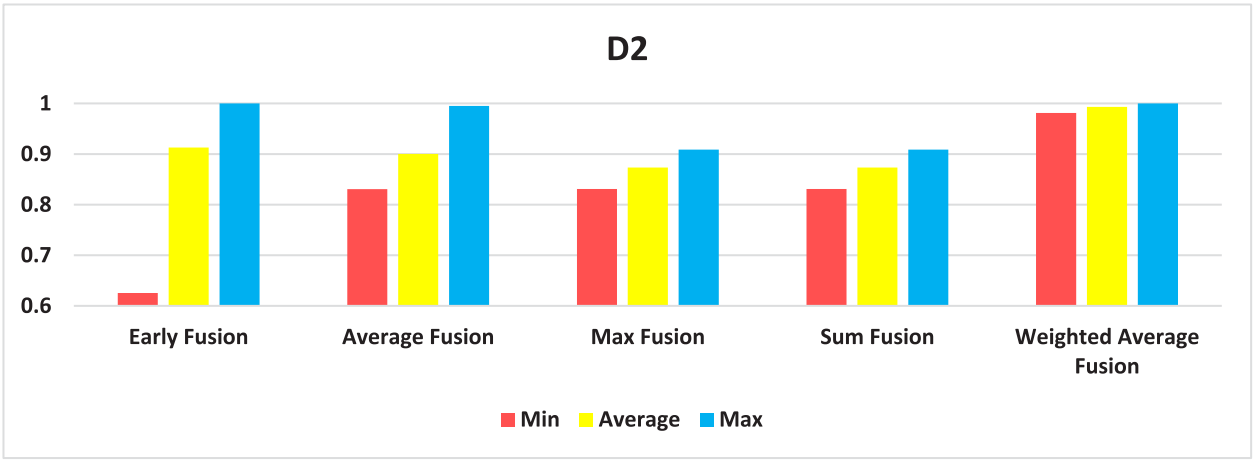

Fig. 32. Comparative analysis of fusion methods on D2.

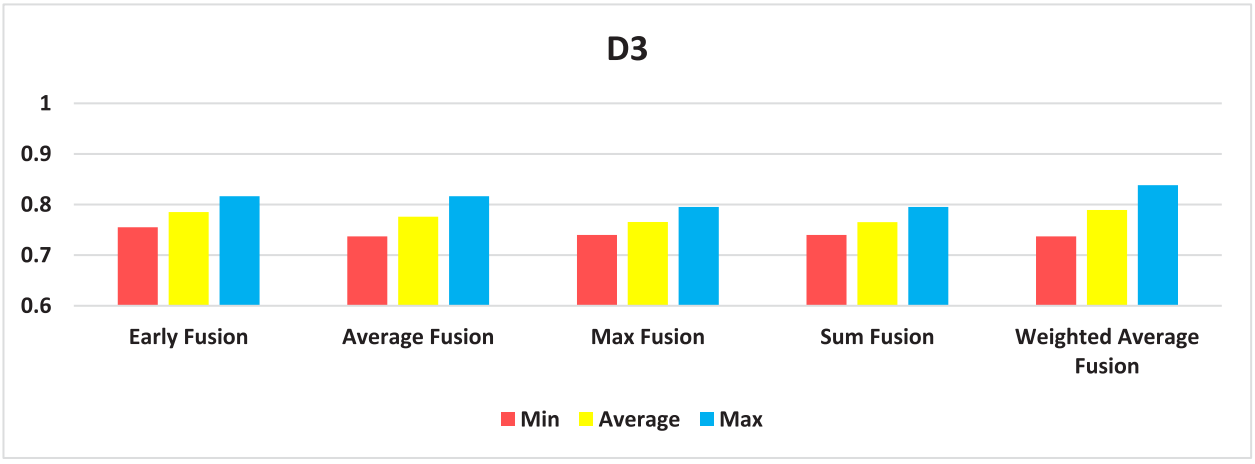

Fig. 33. Comparative analysis of fusion methods on D3.

a good performer but with instability among maximum, average, and minimum results. Sum and max fusion often provided similar results, both showing lower metrics in all cases.

5.4.5. Text and image contribution in weighted-average fusion

In weighted-average fusion, weights are assigned to each data modality, where weights can be described as their contribution to the final prediction results. Each modality, text, and image is assigned a value within 0–1 to represent their share in the final results. We experimented with values to generate optimum results with weighted-average fusion, alternating among the weights. The results in this paper are the best results obtained under such fusion. The weights assigned to the modalities in each classification model for datasets D1–D6 are shown in Fig. 37.

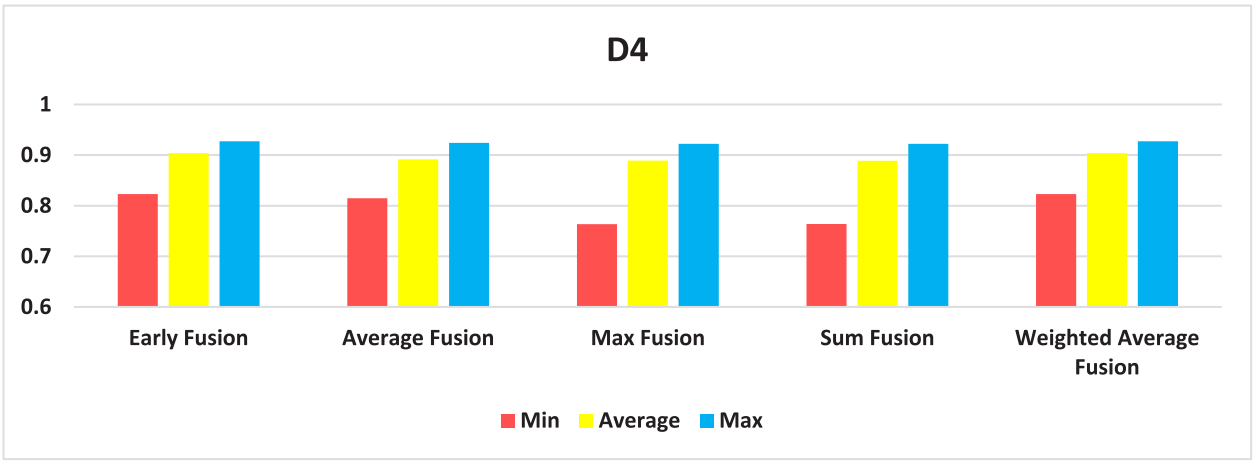

Fig. 34. Comparative analysis of fusion methods on D4.

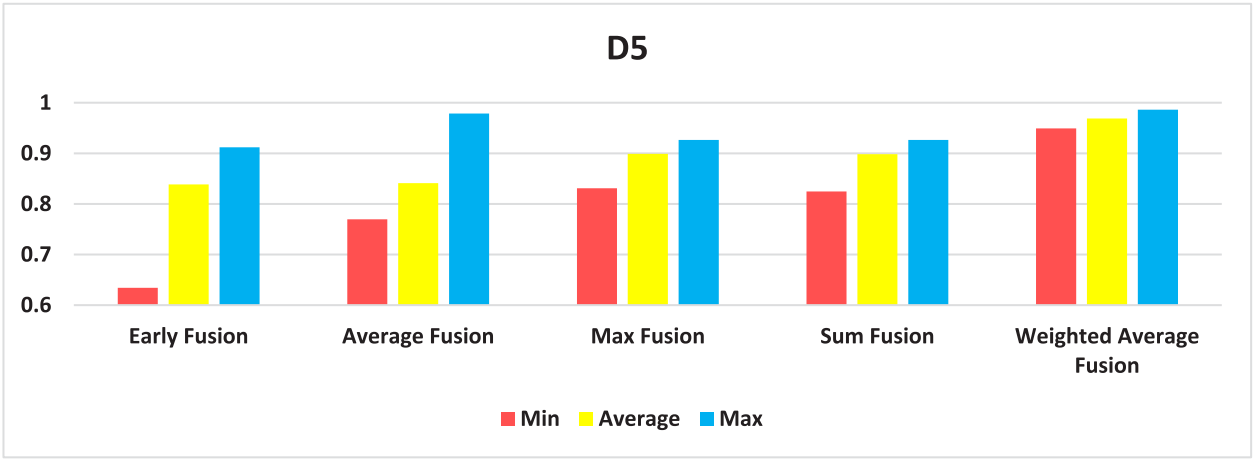

Fig. 35. Comparative analysis of fusion methods on D5.

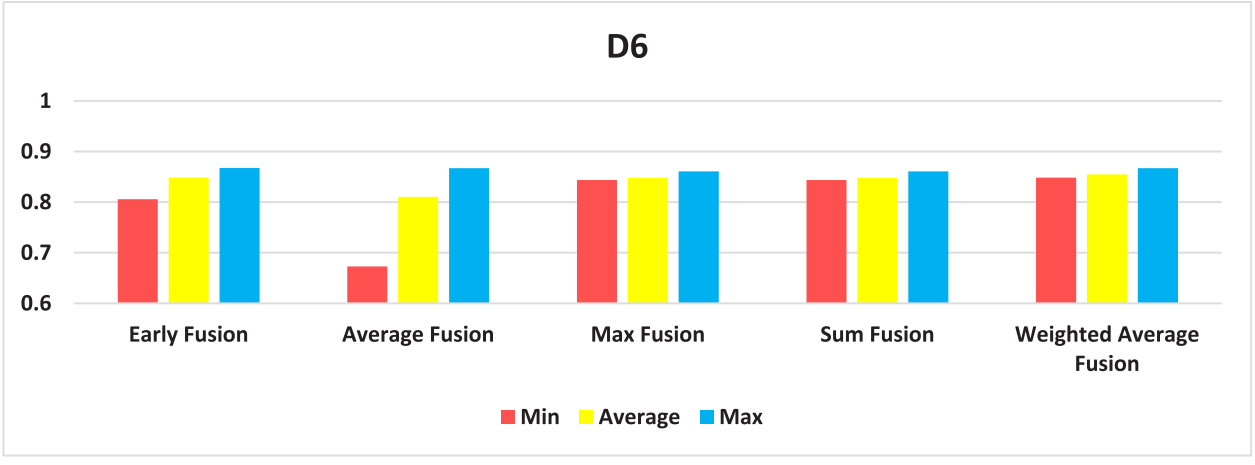

Fig. 36. Comparative analysis of fusion methods on D6.

Analyzing the text and image contribution individually for each dataset showed multiple trends. This variation is attributed to the quality of the datasets. In D1, most classification models worked well by assigning 65% weightage to text and the remaining 35% to images. Observations in D2 were quite varied with four models used with 85% text and 15% image weightage. The rest of the models used 30%–40% image weightage with the remaining to text. Dataset D3 worked well by assigning 40%–50% weightage to visual data. In D4, values fluctuated, displaying most stability with 35%–45% for the image. In D5, most models worked best by

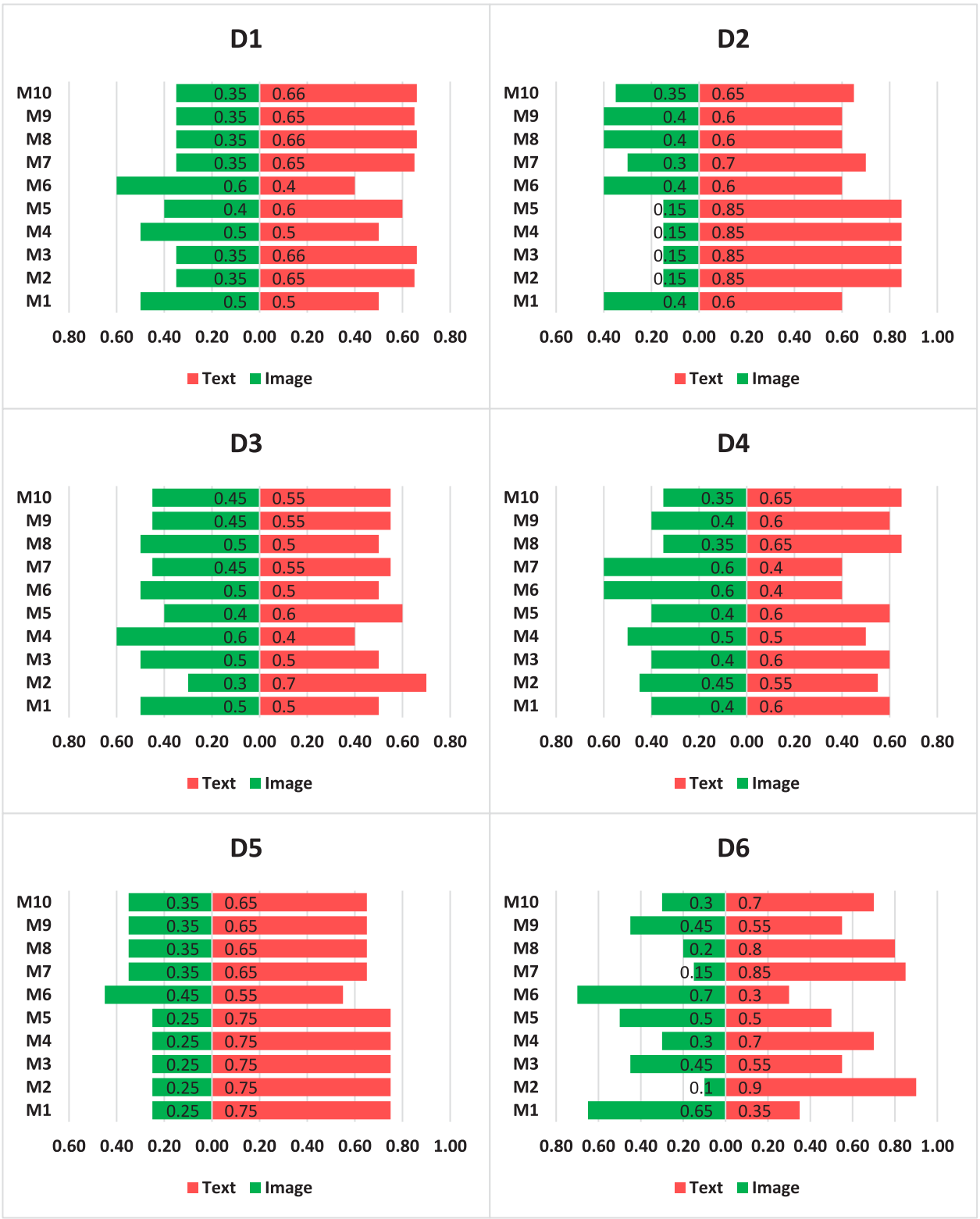

Fig. 37. Text and image contributions in all datasets for weighted-average fusion.

assigning 35% to the image, while others used 25% weightage for the image. In D6, weight assigning for achieving the highest possible accuracies showed a random trend for all models. Overall, visual data was a compelling factor in fake news detection, with average contribution of 30%–50% when combined with textual modality.

5.4.6. Overall performance analysis

To conclude, we provide comparisons based on overall performances by deciding optimum classification models and fusion methods. Weighted-average fusion was the best fusion method with the highest maximum, average, and minimum results (Fig. 38). The next best performance was for early fusion and

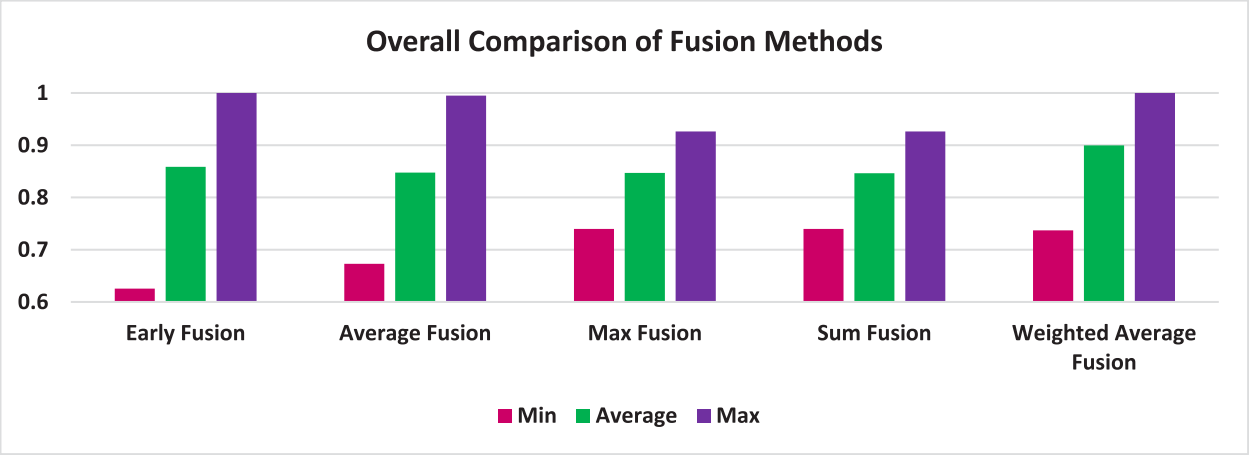

Fig. 38. Overall performance comparison of fusion methods.

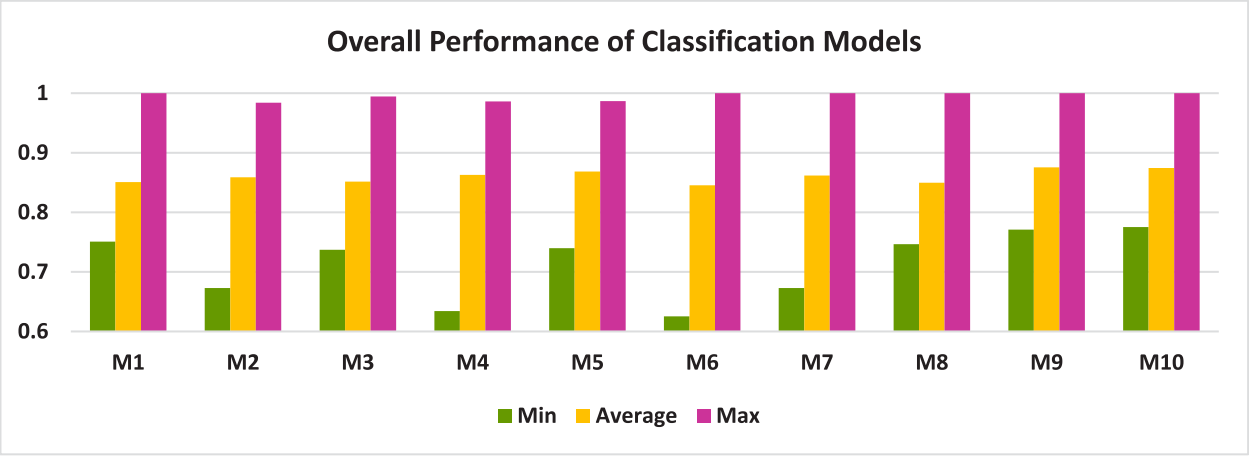

Fig. 39. Overall performance comparison of classification models used.

average fusion, but their average and mins were less significant than the weighted average. Ranking all fusion methods according to their performance, weighted-average fusion stood apart as the best, with early fusion second and average fusion third. Max fusion and sum fusion were on the same level with similar results.

The trends obtained by classification models M1–M10 showed the highest maximum performance (100%) for M1 and M6–M10 (Fig. 39). Models M6–M10 use Bi-LSTM for text classification, which makes better choices than LSTM. The averages showed that XceptionNet and MobileNetV2 were the best choices for image classification on our datasets. Also, considering the minimum results provided by each model, M10 and M9 were best followed by M1, M3, M5, and M8. Therefore, the proposed ARCNN model with specified hyperparameters gave excellent fake news detection provided by any pre-trained classification models.

Different ranges of result scores were obtained for different datasets (Fig. 40). For datasets containing tweets as text, scores were higher compared to datasets with news articles. This is because tweets are short statements, whereas articles are long and complex posts. Accuracy scores also differed for different sizes of datasets. Datasets in which classes were balanced offered more effective predictions than for unbalanced datasets. The size and quality of corpora played a significant part in building an effective classification mechanism.

5.5. Ablation study

Ablation study is the procedure of systematic framework analysis by the removal of its components. This helps in separately identifying the usefulness of each component of the framework. We performed the ablation study to examine the contribution of text classification and image classification models. We experimented with the individual techniques, LSTM, Bi-LSTM, Proposed CNN, VGG-16, InceptionV3, MobileNetV2, and XceptionNet on the six real-world datasets. The parameter settings were kept identical to those of the overall ARCNN framework. The accuracy percentages of the ablation study on six datasets are shown in Table 7. The last row in Table 7 illustrates the highest performance observed by the ARCNN framework wherein text and image components were combined.

5.6. Baseline comparison

To authenticate the worthiness of the proposed model, we compared our approach with three existing baselines on six datasets in terms of accuracy, precision, recall and f1-scores. The proposed approach performed better than the existing ones (Table 8). Since these methods have not been applied to COVID-specific fake news datasets, we reproduced their performance by providing their models with a similar setup to their own. All

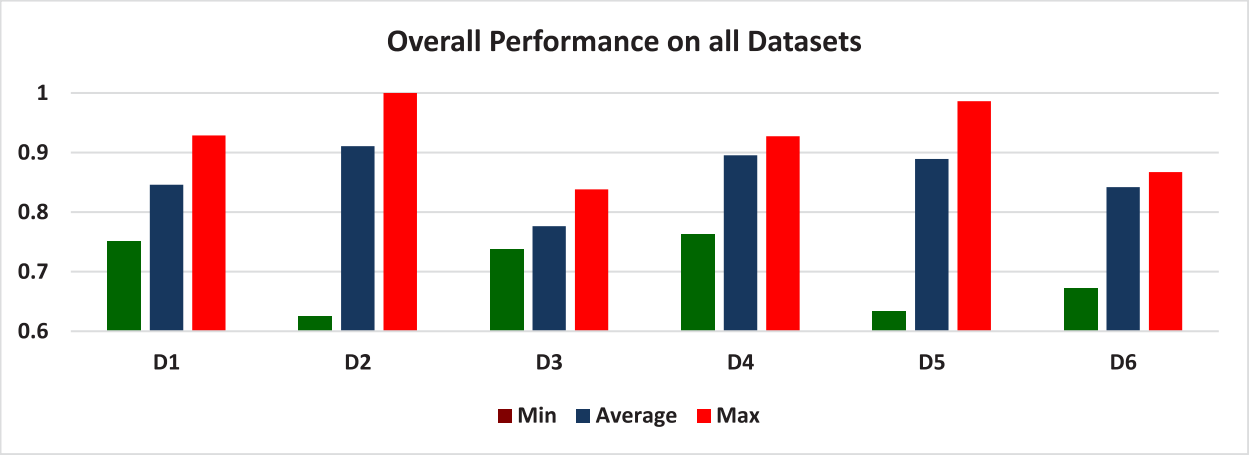

Fig. 40. Overall performance comparison on all datasets.

Table 7
Ablation study of proposed ARCNN framework.

Feature	Accuracy (%)					
	D1	D2	D3	D4	D5	D6
Individual techniques						
Text-based techniques						
LSTM	79.43	96.24	81.51	89.67	96.29	83.71
Bi-LSTM	82.56	98.32	81.92	91.33	96.61	84.01
Image-based techniques						
Proposed CNN	79.98	90.64	65.37	87.14	89.03	76.54
VGG-16	87.83	92.65	67.82	89.33	92.54	79.03
InceptionV3	87.91	95.81	74.12	90.12	96.82	84.61
MobileNetV2	89.25	90.73	73.46	88.65	96.54	83.02
XceptionNet	88.44	96.06	77.10	91.31	91.46	81.09
Overall ARCNN framework						
Text + Image	92.86	100.00	83.82	92.73	98.62	86.73

Table 8
Baseline comparison.

Dataset	Method	Deep Learning Model	Accuracy	Precision	Recall	F1-Score
D1	Att-RNN (Jin et al., 2017)	LSTM + VGG19	73.08	25.00	66.67	36.36
	EANN (Wang et al., 2018)	TextCNN + VGG19	81.32	63.01	86.79	73.02
	TI-CNN (Yang et al., 2018)	Text CNN + Image CNN	85.22	78.15	91.63	84.35
	ARCNN	**Bi-LSTM + MobileNetV2**	**92.86**	**84.09**	**96.52**	**89.88**
D2	Att-RNN (Jin et al., 2017)	LSTM + VGG19	83.03	78.07	85.88	81.79
	EANN (Wang et al., 2018)	TextCNN + VGG19	85.34	73.79	91.57	81.72
	TI-CNN (Yang et al., 2018)	Text CNN + Image CNN	96.76	95.95	97.08	96.51
	ARCNN	**BiLSTM + MobileNetV2**	**100.0**	**100.0**	**100.0**	**100.0**
D3	Att-RNN (Jin et al., 2017)	LSTM + VGG19	67.92	37.01	74.84	52.04
	EANN (Wang et al., 2018)	TextCNN + VGG19	74.24	37.21	69.57	48.48
	TI-CNN (Yang et al., 2018)	Text CNN + Image CNN	80.1	41.02	**84.39**	48.71
	ARCNN	**BiLSTM + MobileNetV2**	**80.98**	**53.85**	58.33	**56.00**
D4	Att-RNN (Jin et al., 2017)	LSTM + VGG19	74.58	51.24	75.11	70.13
	EANN (Wang et al., 2018)	TextCNN + VGG19	78.85	57.47	83.4	73.33
	TI-CNN (Yang et al., 2018)	Text CNN + Image CNN	82.94	59.78	**85.28**	74.01
	ARCNN	**BiLSTM + MobileNetV2**	**91.91**	**75.52**	81.92	**78.59**
D5	Att-RNN (Jin et al., 2017)	LSTM + VGG19	76.28	72.2	74.01	77.66
	EANN (Wang et al., 2018)	TextCNN + VGG19	82.29	70.97	78.41	79.52
	TI-CNN (Yang et al., 2018)	Text CNN + Image CNN	89.76	79.7	79.81	81.18
	ARCNN	**BiLSTM + MobileNetV2**	**95.39**	**89.47**	**100.0**	**94.44**
D6	Att-RNN (Jin et al., 2017)	LSTM + VGG19	73.55	10.52	78.82	18.26
	EANN (Wang et al., 2018)	TextCNN + VGG19	78.25	08.26	72.36	17.26
	TI-CNN (Yang et al., 2018)	Text CNN + Image CNN	80.30	10.25	**79.02**	19.26
	ARCNN	**BiLSTM + MobileNetV2**	**84.83**	**15.15**	55.56	**23.81**

experiments were performed by training these models on all six datasets, and the results evaluated.

Att-RNN: Jin et al. (2017) proposed a fusion architecture that incorporates textual, visual, and social features and combines them using an attention mechanism. For a fair comparison, we combined only textual and visual features. As proposed in their approach, we used LSTM for text and VGG-19 pre-trained on the Imagenet dataset for the images. The hidden layer dimension for text was set to 32, and the tanh activation function was used. The entire network was trained for 100 epochs with early stopping and a batch size of 128.

EANN: In this approach, the Text-CNN model was used for textual feature extraction, and VGG-19 used for visuals (Wang et al., 2018). Features from both streams were concatenated as an early fusion to form a single set of feature maps, and the model was trained thereafter. We eliminated the event discriminator used in EANN. We trained the model for 100 epochs using early stopping and a batch size of 64.

TI-CNN: Yang et al. (2018) utilizes implicit and explicit text and image features and then combines them with early fusion. We used implicit features pre-existing in text and image and trained the model after early fusion. The textual branch consisted of one-dimensional convolution, while the visual branch used three-dimensional convolutional layers. Features from both CNNs were joined using concatenation, and the model trained for 100 epochs with early stopping and a batch size of 64. As observed from Table 8, results achieved by the ARCNN framework are higher as compared to the baselines, which is attributed to the model selection, hyper-parameter settings, fine-tuning and the fusion mechanisms. To maintain a fair comparison, we select a single combination from the ARCNN framework. The BiLSTM + MobileNetV2 fusion remains consistent throughout the experiments with each dataset, yielding high results. It is also the best performing model for D1, D2, and D5 datasets. Analyzing the reasons causing huge variations between the baseline and ARCNN results, it is noticeable that the baselines use different RNN and CNN combinations than those implemented in the ARCNN framework. The Att-RNN model (Jin et al., 2017) relies on a LSTM + VGG19 combination, incorporating attention mechanism for obtaining contextual understanding of the text. The EANN mechanism (Wang et al., 2018) builds a Text-CNN combining it with VGG19 while the TI-CNN model (Yang et al., 2018) incorporates a Text-CNN and Image-CNN combination where both networks are self-designed. As discussed by (Wang et al., 2018), the multi-modal feature representations are event-dependent in the Att-RNN mechanism and thus cannot be generalized for incoming fake news. They proposed EANN to overcome this limitation but their approach substantially performs lower in the absence of the event discriminator. ARCNN's BiLSTM + MobileNetV2 network differs in architecture from the baselines. The proposed approach highlights the effectiveness of a Bi-directional LSTM over a unidirectional network as used in all of the baselines. This underlines the requirement of forward and backward processing of textual information. ARCNN's visual classifier, MobileNetV2 surpasses VGG16's performance due to its simplified model architecture, fewer operations, and higher efficacy. The residual block and depth-wise separable convolution architecture is superior to VGG16, eliminating the need of a deep neural network consisting of higher number of layers. Added accuracy is attributed to the ARCNN's fusion mechanism. Although, early fusion and weighted average fusion have obtained comparable scores, weighted average is slightly an edge over owing to its weight adaptation ability. Hyper-parameter optimization and fine-tuning following successive experimentations adds to the ARCNN's overall performance. Additionally, ARCNN eliminates the requirement of explicit features such as those used in the Att-RNN or TI-CNN models by solely relying upon implicit textual and visual features.

6. Conclusion

In this work, we propose the ARCNN architecture for fake news detection. Our framework uses 10 combinations of text and image classification models to detect fake news based on two modalities: text and image. We provide a generic architecture that can incorporate a pre-trained classification model of choice. We use LSTM and Bi-LSTM in the RNN component of the framework for text classification. In the CNN component, we use the proposed CNN, fine-tuned VGG-16, MobileNetV2, InceptionV3, and XceptionNet for image classification. We have conducted experiments on six COVID-19 fake news datasets alternating with various text and image classification models. Our introduced datasets, CovID I and CovID II, are publicly available. The source codes for the performed experiments are publicly available on GitHub[6]. The two streams of data are combined using early fusion and four types of late fusion techniques. We presented vast experimentation and study in fake news detection. The proposed architecture outperforms various state-of-the-art fake news detection models. Results are calculated in terms of eight evaluation metrics for all conducted experiments. For easier understanding of results, the data are neatly represented in various graphs. Trends are observed and analyzed for fellow researchers providing a deep study that can be readily utilized to build fake news detection models. Our work leverages deep learning and combines various techniques to develop a novel and scalable fake news detection mechanism. To demonstrate plausibility, we provide a helpful experimentation study for fake news detection.

6.1. Future work

In the present scenario, there is a lack of infodemic datasets and detection mechanisms. This leads to the challenge of distinguishing fake news from real, and hence, dealing with it becomes problematic. Coronavirus-related fake news datasets are still limited to textual information. Datasets containing various other information like visual data or meta-data, which could be helpful in detection, are very few. We propose an architecture that uses two modalities. Our proposed architecture is flexible in accepting more data streams from different modalities that can be fused. Due to the low availability of versatile data, we exploit text and images used in social media posts and news articles. We intend to utilize video features for detection based on fake news videos. We also encourage fellow researchers to build a holistic fake news detection framework that could capture most of the possible details in a piece of news and exploit them for efficient fake news detection. Future work includes the building of our proposed framework as an application or browser plugin. Collection of versatile and balanced real-world datasets and designing better mechanisms to detect fake news in real-time is promoted.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

See Tables A.1–A.6.

[6] https://github.com/chahatraj/TIPNet-Infodemic-Detection

Table A.1
Results obtained on D1 (CovID I).

Fusion	Model	F1	Accuracy	Precision	Recall	FPR	ROC	Spec.	MCC
Early Fusion	M1	0.8145	0.8343	0.8604	0.7733	0.1115	0.9119	0.8885	0.6686
	M2	0.8477	0.8943	0.7803	0.9279	0.1213	0.9581	0.8787	0.7745
	M3	0.8049	0.8629	0.75	0.8684	0.1398	0.9249	0.8602	0.7045
	M4	0.8661	0.9029	0.8333	0.9016	0.0965	0.9588	0.9035	0.7916
	M5	0.8315	0.8657	0.8788	0.7891	0.0788	0.9427	0.9212	0.7233
	M6	0.7739	0.8286	0.6937	0.875	0.1948	0.8286	0.8052	0.65
	M7	0.7734	0.8343	0.75	0.7984	0.146	0.9176	0.854	0.6438
	M8	0.7612	0.8029	0.8333	0.7006	0.114	0.9034	0.886	0.602
	M9	0.8988	**0.9286**	0.8409	0.9652	0.0894	**0.9625**	0.9106	0.8488
	M10	0.8327	0.8657	0.8864	0.7852	0.0746	0.9388	0.9254	0.7249
Average Fusion	M1	0.8008	0.8296	0.6849	0.9638	0.2444	0.8296	0.7556	0.6887
	M2	0.8632	0.8886	0.8662	0.8601	0.0918	0.885	0.9082	0.7692
	M3	0.7489	0.8314	0.6197	0.9462	0.2101	0.7978	0.7899	0.6622
	M4	0.8955	0.92	0.8451	0.9524	0.0982	0.9081	0.9018	0.835
	M5	0.8529	0.8857	0.8169	0.8923	0.1182	0.8748	0.8818	0.7618
	M6	0.8595	0.8748	0.7658	0.9793	0.1923	0.8748	0.8077	0.7681
	M7	0.8429	0.8829	0.8271	0.8594	0.1036	0.8721	0.8964	0.7499
	M8	0.7459	0.8229	0.6842	0.8198	0.1757	0.796	0.8243	0.6175
	M9	0.8178	0.8714	0.7594	0.886	0.1356	0.8497	0.8644	0.7245
	M10	0.8095	0.8629	0.7669	0.8571	0.1342	0.8443	0.8658	0.7056
Max Fusion	M1	0.6765	0.7532	0.516	0.9817	0.3283	0.7532	0.6717	0.5753
	M2	0.7232	0.8229	0.5704	**0.9878**	0.2276	0.7828	0.7724	0.6557
	M3	0.6087	0.7686	0.4437	0.9692	0.2772	0.717	0.7228	0.5481
	M4	0.7032	0.8143	0.5423	1	0.2381	0.7711	0.7619	0.6428
	M5	0.682	0.8029	0.5211	0.9867	0.2473	0.7582	0.7527	0.6179
	M6	0.7255	0.7818	0.5766	0.978	0.3002	0.7818	0.6998	0.618
	M7	0.7611	0.8457	0.6466	0.9247	0.1829	0.8072	0.8171	0.6751
	M8	0.6763	0.8086	0.5263	0.9459	0.2283	0.7539	0.7717	0.6037
	M9	0.729	0.8343	0.5865	0.963	0.2045	0.7863	0.7955	0.6591
	M10	0.7315	0.8343	0.594	0.9518	0.2022	0.7878	0.7978	0.6568
Sum Fusion	M1	0.6725	0.7509	0.5114	0.9816	0.3304	0.7509	0.6696	0.5716
	M2	0.7232	0.8229	0.5704	0.9878	0.2276	0.7828	0.7724	0.6557
	M3	0.6087	0.7686	0.4437	0.9692	0.2772	0.717	0.7228	0.5481
	M4	0.7032	0.8143	0.5423	1	0.2381	0.7711	0.7619	0.6428
	M5	0.682	0.8029	0.5211	0.9867	0.2473	0.7582	0.7527	0.6179
	M6	0.7087	0.7722	0.5541	0.9828	0.3105	0.7722	0.6895	0.605
	M7	0.7611	0.8457	0.6466	0.9247	0.1829	0.8072	0.8171	0.6751
	M8	0.6763	0.8086	0.5263	0.9459	0.2283	0.7539	0.7717	0.6037
	M9	0.729	0.8343	0.5865	0.963	0.2045	0.7863	0.7955	0.6591
	M10	0.7315	0.8343	0.594	0.9518	0.2022	0.7878	0.7978	0.6568
Weighted Average	M1	0.8871	0.8885	0.8767	0.8978	0.1205	0.8885	0.8795	0.7772
	M2	0.8614	0.8943	0.8099	0.92	0.12	0.8809	0.88	0.7807
	M3	0.7782	0.8486	0.6549	0.9588	0.1937	0.8178	0.8063	0.6974
	M4	0.8955	0.92	0.8451	0.9524	0.0982	0.9081	0.9018	0.835
	M5	0.8571	0.8886	0.8239	0.8931	0.1142	0.8783	0.8858	0.7677
	M6	**0.8996**	0.9005	0.8919	0.9075	0.1063	0.9005	0.8937	0.8011
	M7	0.8664	0.8943	**0.9023**	0.8333	**0.0631**	0.8958	0.9369	0.7809
	M8	0.8401	0.8771	0.8496	0.8309	0.0935	0.8702	0.9065	0.7405
	M9	0.8593	0.8943	0.8496	0.8692	0.0909	0.8856	0.9091	0.7748
	M10	0.8476	0.8829	0.8571	0.8382	0.0888	0.8779	0.9112	0.7526

Table A.2
Results obtained on D2 (CovID II).

Fusion	Model	F1	Accuracy	Precision	Recall	FPR	ROC	Spec	MCC
Early Fusion	M1	1	**1**	1	1	0	1	1	1
	M2	0.9124	0.9355	0.8389	1	0.0972	0.9991	0.9028	0.8703
	M3	0.9932	0.9946	0.9866	1	0.0089	1	0.9911	0.9888
	M4	0.9544	0.9651	0.9128	1	0.0551	0.9999	0.9449	0.9287
	M5	0.9829	0.9866	0.9664	1	0.0219	0.9999	0.9781	0.9722
	M6	0	0.6254	0	0	0.3746	1	0.6254	0
	M7	0.9544	0.9651	0.9128	1	0.0551	0.9945	0.9449	0.9287
	M8	0.614	0.7769	0.443	1	0.2712	0.9999	0.7288	0.5682
	M9	0.8405	0.8898	0.7248	1	0.1553	0.9999	0.8447	0.7825
	M10	0.9864	0.9892	0.9732	1	0.0176	1	0.9824	0.9778
Average Fusion	M1	0.9951	0.9951	0.9902	1	0.0097	0.995	0.9903	0.9902
	M2	0.7536	0.8629	0.6047	1	0.1735	0.8023	0.8265	0.7069
	M3	0.783	0.8763	0.6434	1	0.1592	0.8217	0.8408	0.7355
	M4	0.843	0.9059	0.7287	1	0.1259	0.8643	0.8741	0.7981
	M5	0.8684	0.9194	0.7674	1	0.1099	0.8837	0.8901	0.8265
	M6	0.9951	0.9951	0.9902	1	0.0097	0.9951	0.9903	0.9902
	M7	0.7826	0.8522	0.6644	0.9519	0.1866	0.821	0.8134	0.7019
	M8	0.7623	0.8306	0.6779	0.8707	0.1875	0.8053	0.8125	0.6459
	M9	0.8487	0.8898	0.7718	0.9426	0.136	0.8702	0.864	0.7728
	M10	0.8284	0.8763	0.745	0.9328	0.1502	0.8545	0.8498	0.7449
Max Fusion	M1	0.807	0.8383	0.6765	1	0.2444	0.8382	0.7556	0.7149
	M2	0.7228	0.8495	0.5659	1	0.1873	0.7829	0.8127	0.6782
	M3	0.7656	0.8683	0.6202	1	0.1678	0.8101	0.8322	0.7184
	M4	0.8219	0.8952	0.6977	1	0.1383	0.8488	0.8617	0.7754
	M5	0.8482	0.9086	0.7364	1	0.1227	0.8682	0.8773	0.8038
	M6	0.7965	0.8309	0.6618	1	0.2527	0.8309	0.7473	0.7032
	M7	0.7984	0.8656	0.6644	1	0.1832	0.8322	0.8168	0.7367
	M8	0.808	0.871	0.6779	1	0.1771	0.8389	0.8229	0.7469
	M9	0.8712	0.9086	0.7718	1	0.1323	0.8859	0.8677	0.8184
	M10	0.8538	0.8978	0.745	1	0.1456	0.8725	0.8544	0.7978
Sum Fusion	M1	0.807	0.8383	0.6765	1	0.2444	0.8382	0.7556	0.7149
	M2	0.7228	0.8495	0.5659	1	0.1873	0.7829	0.8127	0.6782
	M3	0.7656	0.8683	0.6202	1	0.1678	0.8101	0.8322	0.7184
	M4	0.8219	0.8952	0.6977	1	0.1383	0.8488	0.8617	0.7754
	M5	0.8482	0.9086	0.7364	1	0.1227	0.8682	0.8773	0.8038
	M6	0.7965	0.8309	0.6618	1	0.2527	0.8309	0.7473	0.7032
	M7	0.7984	0.8656	0.6644	1	0.1832	0.8322	0.8168	0.7367
	M8	0.808	0.871	0.6779	1	0.1771	0.8389	0.8229	0.7469
	M9	0.8712	0.9086	0.7718	1	0.1323	0.8859	0.8677	0.8184
	M10	0.8538	0.8978	0.745	1	0.1456	0.8725	0.8544	0.7978
Weighted Average	M1	0.9975	0.9976	0.9951	1	0.0049	0.9975	0.9951	0.9951
	M2	0.9762	0.9839	0.9535	1	0.0241	0.9767	0.9759	0.9646
	M3	0.9721	0.9812	0.9457	1	0.028	0.9729	0.972	0.9588
	M4	0.9762	0.9839	0.9535	1	0.0241	0.9767	0.9759	0.9646
	M5	0.9762	0.9839	0.9535	1	0.0241	0.9767	0.9759	0.9646
	M6	1	**1**	1	1	0	1	1	1
	M7	1	**1**	1	1	0	1	1	1
	M8	1	**1**	1	1	0	1	1	1
	M9	1	**1**	1	1	0	1	1	1
	M10	1	**1**	1	1	0	1	1	1

Table A.3
Results obtained on D3 (ReCOVery news articles).

Fusion	Model	F1	Accuracy	Precision	Recall	FPR	ROC	Spec	MCC
Early Fusion	M1	0.5164	0.8012	0.3901	0.7639	0.1928	0.6725	0.8072	0.4439
	M2	0.4774	0.7666	0.4022	0.5873	0.1937	0.721	0.8063	0.3438
	M3	0.4098	0.7925	0.2717	0.8333	0.2114	0.7588	0.7886	0.396
	M4	0.4615	0.7781	0.3587	0.6471	0.1993	0.7455	0.8007	0.3592
	M5	0.4394	0.7867	0.3152	0.725	0.2052	0.7867	0.7948	0.3761
	M6	0.5887	0.8166	0.4823	0.7556	0.1706	0.712	0.8294	0.498
	M7	0.3972	0.755	0.3043	0.5714	0.2148	0.7179	0.7852	0.2814
	M8	0.56	0.8098	0.4565	0.7241	0.173	0.789	0.827	0.4659
	M9	0.4397	0.771	0.337	0.6327	0.2061	0.727	0.7939	0.3367
	M10	0.3276	0.7752	0.2065	0.7917	0.226	0.7569	0.774	0.3252
Average Fusion	M1	0.5164	0.8012	0.3901	0.7639	0.1928	0.6725	0.8072	0.4439
	M2	0.4409	0.7948	0.3733	0.5385	0.1599	0.6424	0.8401	0.3284
	M3	0.0632	0.7428	0.0333	0.6	0.2551	0.5128	0.7449	0.0938
	M4	0	0.737	0	0	0.2609	0.498	0.7391	−0.0319
	M5	0.3932	0.7948	0.2556	0.8519	0.21	0.62	0.79	0.3924
	M6	0.5887	0.8166	0.4823	0.7556	0.1706	0.712	0.8294	0.498
	M7	0.3731	0.7579	0.3205	0.4464	0.1821	0.6026	0.8179	0.2329
	M8	0.3714	0.7464	0.3333	0.4194	0.1825	0.5998	0.8175	0.2174
	M9	0.3967	0.7896	0.3077	0.5581	0.1776	0.6185	0.8224	0.3003
	M10	0.3529	0.7781	0.2692	0.5122	0.1863	0.5974	0.8137	0.252
Max Fusion	M1	0.25	0.7568	0.1489	0.7778	0.2444	0.5665	0.7556	0.2664
	M2	0.0519	0.789	0.0267	1	0.2122	0.5133	0.7878	0.1449
	M3	0	0.7399	0	0	0.2601	0.5	0.7399	0
	M4	0	0.7399	0	0	0.2601	0.5	0.7399	0
	M5	0	0.7399	0	0	0.2601	0.5	0.7399	0
	M6	0.32	0.7703	0.1986	0.8235	0.2335	0.5913	0.7665	0.3283
	M7	0.3478	0.7839	0.2564	0.5405	0.1871	0.5966	0.8129	0.2613
	M8	0.3768	0.7522	0.2559	0.4333	0.1812	0.6035	0.8188	0.2284
	M9	0.3238	0.7954	0.2179	0.6296	0.1906	0.5904	0.8094	0.2817
	M10	0.3048	0.7896	0.2051	0.5926	0.1938	0.5821	0.8063	0.2559
Sum Fusion	M1	0.2424	0.7587	0.1418	0.8333	0.2449	0.5656	0.7551	0.2779
	M2	0.0263	0.7861	0.0133	1	0.2145	0.5067	0.7855	0.1023
	M3	0	0.7399	0	0	0.2601	0.5	0.7399	0
	M4	0	0.7399	0	0	0.2601	0.5	0.7399	0
	M5	0	0.7399	0	0	0.2601	0.5	0.7399	0
	M6	0.3103	0.7683	0.1915	0.8182	0.2351	0.5878	0.7649	0.32
	M7	0.3478	0.7839	0.2564	0.5405	0.1871	0.5966	0.8129	0.2613
	M8	0.3768	0.7522	0.3333	0.4333	0.1812	0.6035	0.8188	0.2284
	M9	0.3238	0.7954	0.2179	0.6296	0.1906	0.5904	0.8094	0.2817
	M10	0.3048	0.7896	0.2051	0.5926	0.1938	0.5821	0.8063	0.2559
Weighted Average	M1	0.5164	0.8012	0.3901	0.7639	0.1928	0.6725	0.8072	0.4439
	M2	0.541	**0.8382**	0.44	0.7021	0.1405	0.6942	0.8595	0.467
	M3	0.3724	0.737	0.3	0.4909	0.2165	0.5651	0.7835	0.2287
	M4	0.4286	0.7688	0.3333	0.6	0.2027	0.6276	0.7973	0.3184
	M5	0.4167	0.7977	0.2778	0.8333	0.2057	0.6291	0.7943	0.4026
	M6	0.5887	0.8166	0.4823	0.7556	0.1706	0.712	0.8294	0.498
	M7	0.525	0.781	0.5385	0.5122	0.1358	0.6949	0.8642	0.383
	M8	0.3714	0.7464	0.3333	0.4194	0.1825	0.5998	0.8175	0.2174
	M9	0.56	0.8098	0.5385	0.5833	0.1309	0.7135	0.8691	0.4395
	M10	0.5235	0.7954	0.5	0.5493	0.1413	0.6902	0.8587	0.3943

Table A.4
Results obtained on D4 (ReCOVery tweets).

Fusion	Model	F1	Accuracy	Precision	Recall	FPR	ROC	Spec	MCC
Early Fusion	M1	0.7939	0.8231	0.6818	0.9502	0.2481	0.823	0.7519	0.6735
	M2	0.7553	0.9172	0.6614	0.8803	0.0766	0.8199	0.9234	0.7171
	M3	0.7673	0.9243	0.6854	0.8714	0.0669	0.8314	0.9331	0.7303
	M4	0.7415	0.9222	0.6124	0.9397	0.0801	0.8018	0.9199	0.7203
	M5	0.7292	0.9202	0.5899	0.9545	0.0842	0.7918	0.9158	0.7127
	M6	0.8303	0.8517	0.7259	0.9698	0.219	0.8517	0.781	0.7267
	M7	0.756	**0.9273**	0.618	0.9735	0.0787	0.807111	0.9213	0.7413
	M8	0.7822	0.915	0.776	0.7884	0.0546	0.8625	0.9454	0.7294
	M9	0.7893	0.9191	0.7708	0.8087	0.0554	0.8631	0.9446	0.7397
	M10	0.7859	0.9191	0.7552	0.8192	0.0588	0.868	0.9413	0.7371
Average Fusion	M1	0.7776	0.8147	0.6477	0.9725	0.2641	0.8147	0.7359	0.6677
	M2	0.7485	0.914	0.7022	0.8013	0.0646	0.8317	0.9354	0.6991
	M3	0.7357	0.9007	0.7584	0.7143	0.0546	0.8454	0.9454	0.6751
	M4	0.7673	0.9243	0.6854	0.8714	0.0669	0.8314	0.9331	0.7303
	M5	0.7104	0.9007	0.6685	0.758	0.072	0.8105	0.928	0.6527
	M6	0.8223	0.8464	0.7111	0.9748	0.2274	0.8464	0.7726	0.7195
	M7	0.7538	0.9172	0.6561	0.8857	0.0776	0.8179	0.9224	0.7168
	M8	0.6981	0.869	0.7708	0.6379	0.0591	0.8319	0.9409	0.6199
	M9	0.7574	0.9161	0.6667	0.8767	0.077	0.8219	0.923	0.7175
	M10	0.744	0.912	0.651	0.8681	0.0804	0.8134	0.9196	0.7026
Max Fusion	M1	0.6916	0.7635	0.5303	0.9938	0.3203	0.7635	0.6797	0.5958
	M2	0.6716	0.9099	0.5056	1	0.0992	0.7528	0.9008	0.6749
	M3	0.674	0.9089	0.5169	0.9684	0.0975	0.7565	0.9025	0.6685
	M4	0.6716	0.9089	0.5112	0.9785	0.0984	0.7544	0.9016	0.6691
	M5	0.6543	0.9048	0.4944	0.967	0.1016	0.7453	0.8984	0.6516
	M6	0.7715	0.8136	0.6296	0.996	0.2708	0.8136	0.7292	0.6744
	M7	0.7516	0.9223	0.6085	0.9829	0.0859	0.803	0.9141	0.7372
	M8	0.7599	0.9191	0.651	0.9124	0.0798	0.8179	0.9202	0.7276
	M9	0.7524	0.9212	0.6094	0.9832	0.0874	0.8034	0.9126	0.7373
	M10	0.7403	0.9181	0.5938	0.9828	0.0906	0.7956	0.9094	0.7263
Sum Fusion	M1	0.6919	0.7639	0.5303	0.9953	0.3201	0.7639	0.6799	0.597
	M2	0.6716	0.9099	0.5056	1	0.0992	0.7528	0.9008	0.6749
	M3	0.674	0.9089	0.5169	0.9684	0.0975	0.7565	0.9025	0.6685
	M4	0.6716	0.9089	0.5112	0.9785	0.0984	0.7544	0.9016	0.6691
	M5	0.6543	0.9048	0.4944	0.967	0.1016	0.7453	0.8984	0.6516
	M6	0.7695	0.8099	0.6222	0.996	0.2747	0.8099	0.7253	0.6686
	M7	0.7516	0.9223	0.6085	0.9829	0.0859	0.803	0.9141	0.7372
	M8	0.7599	0.9191	0.651	0.9124	0.0798	0.8179	0.9202	0.7276
	M9	0.7524	0.9212	0.6094	0.9832	0.0874	0.8034	0.9126	0.7373
	M10	0.7403	0.9181	0.5938	0.9828	0.0906	0.7956	0.9094	0.7263
Weighted Average	M1	0.7939	0.8231	0.6818	0.9502	0.2481	0.823	0.7519	0.6735
	M2	0.756	**0.9273**	0.618	0.9735	0.0787	0.807111	0.9213	0.7413
	M3	0.7415	0.9222	0.6124	0.9397	0.0801	0.8018	0.9199	0.7203
	M4	0.7673	0.9243	0.6854	0.8714	0.0669	0.8314	0.9331	0.7303
	M5	0.7292	0.9202	0.5899	0.9545	0.0842	0.7918	0.9158	0.7127
	M6	0.8303	0.8517	0.7259	0.9698	0.219	0.8517	0.781	0.7267
	M7	0.7553	0.9172	0.6614	0.8803	0.0766	0.8199	0.9234	0.7171
	M8	0.7822	0.915	0.776	0.7884	0.0546	0.8625	0.9454	0.7294
	M9	0.7859	0.9191	0.7552	0.8192	0.0588	0.868	0.9413	0.7371
	M10	0.7893	0.9191	0.7708	0.8087	0.0554	0.8631	0.9446	0.7397

Table A.5
Results obtained on D5 (CoAID).

Fusion	Model	F1	Accuracy	Precision	Recall	FPR	ROC	Spec	MCC
Early Fusion	M1	0.8601	0.8735	0.7925	0.9403	0.1737	0.9602	0.8263	0.7552
	M2	0.823	0.8009	0.9901	0.7042	0.0135	0.965	0.9865	0.657
	M3	0.776	0.8102	0.703	0.8659	0.2239	0.919	0.7761	0.6244
	M4	0.368	0.6343	0.2277	0.9583	0.4063	0.9421	0.5938	0.3477
	M5	0.84	0.8519	0.8317	0.8485	0.1453	0.9142	0.8547	0.7022
	M6	0.8845	0.8735	0.9874	0.801	0.0156	0.9846	0.9844	0.768
	M7	0.8791	0.8981	0.7921	0.9877	0.1556	0.9855	0.8444	0.8074
	M8	0.8731	0.8843	0.8515	0.8958	0.125	0.9532	0.875	0.7677
	M9	0.8479	0.8472	0.9109	0.7931	0.09	0.943	0.91	0.7026
	M10	0.91	0.912	0.9505	0.8727	0.0472	0.9715	0.9528	0.8272
Average Fusion	M1	0.9789	0.9785	0.9759	0.9818	0.025	0.9785	0.975	0.9569
	M2	0.8	0.8065	0.8155	0.785	0.1727	0.8069	0.8273	0.613
	M3	0.7727	0.7696	0.8252	0.7265	0.18	0.7723	0.82	0.5455
	M4	0.8455	0.8433	0.9029	0.7949	0.1	0.8462	0.9	0.6936
	M5	0.8	0.788	0.8932	0.7244	0.122	0.7931	0.8778	0.5941
	M6	0.9498	0.9538	0.9342	0.966	0.0562	0.9527	0.9438	0.9076
	M7	0.7834	0.7834	0.8947	0.6967	0.1053	0.7957	0.8947	0.5915
	M8	0.785	0.788	0.8842	0.7059	0.1122	0.7987	0.8878	0.5955
	M9	0.839	0.8479	0.9053	0.7818	0.0841	0.8543	0.9159	0.7031
	M10	0.8351	**0**.8579	0.8526	0.8182	0.1186	0.8525	0.8814	0.7023
Max Fusion	M1	0.8678	0.88	0.7711	0.9922	0.1939	0.8824	0.8061	0.7814
	M2	0.8817	0.8986	0.7961	0.988	0.1567	0.8937	0.8433	0.809
	M3	0.8681	0.8894	0.767	1	0.1739	0.8835	0.8261	0.796
	M4	0.9167	0.9263	0.8544	0.9888	0.1172	0.9228	0.8828	0.8585
	M5	0.914	0.9252	0.85	0.9884	0.1172	0.9228	0.8828	0.8561
	M6	0.7809	0.8308	0.6447	0.9899	0.2389	0.8195	0.7611	0.6927
	M7	0.8953	0.9171	0.8105	1	0.1286	0.9053	0.8714	0.8404
	M8	0.8706	0.8986	0.7789	0.9867	0.1479	0.8854	0.8521	0.804
	M9	0.908	0.9263	0.8316	1	0.1159	0.9158	0.8841	0.8574
	M10	0.869	0.8986	0.7684	1	0.1528	0.8842	0.8472	0.8069
Sum Fusion	M1	0.8639	0.8769	0.7651	0.9922	0.198	0.8794	0.802	0.7763
	M2	0.8817	0.8986	0.7961	0.988	0.1567	0.8937	0.8433	0.809
	M3	0.8681	0.8894	0.767	1	0.1739	0.8835	0.8261	0.796
	M4	0.9167	0.9263	0.8544	0.9888	0.1172	0.9228	0.8828	0.8585
	M5	0.914	0.9252	0.85	0.9884	0.1172	0.9228	0.8828	0.8561
	M6	0.7711	0.8246	0.6316	0.9897	0.2456	0.8129	0.7544	0.6824
	M7	0.8953	0.9171	0.8105	1	0.1286	0.9053	0.8714	0.8404
	M8	0.8706	0.8986	0.7789	0.9867	0.1479	0.8854	0.8521	0.804
	M9	0.908	0.9263	0.8316	1	0.1159	0.9158	0.8841	0.8574
	M10	0.869	0.8986	0.7684	1	0.1528	0.8842	0.8472	0.8069
Weighted Average	M1	0.9849	0.9846	0.9819	0.9879	0.0188	0.9847	0.9813	0.9692
	M2	0.9754	0.977	0.9612	0.99	0.0342	0.9762	0.9658	0.9541
	M3	0.9758	0.977	0.9806	0.9712	0.0177	0.9771	0.9823	0.9539
	M4	0.9854	**0.9862**	0.9806	0.9902	0.0174	0.9859	0.9826	0.9723
	M5	0.9709	0.9724	0.9709	0.9709	0.0263	0.9723	0.9737	0.9446
	M6	0.9699	0.9723	0.9539	0.9864	0.0393	0.9712	0.9607	0.9447
	M7	0.9405	0.9493	0.9158	0.9667	0.063	0.9456	0.937	0.8974
	M8	0.963	0.9677	0.9579	0.9681	0.0325	0.9667	0.9675	0.9344
	M9	0.9444	0.9539	0.8947	1	0.0758	0.9474	0.9242	0.9094
	M10	0.9399	0.9493	0.9053	0.9773	0.0698	0.9444	0.9302	0.8981

Table A.6
Results obtained on D6 (MediaEval 2020).

Fusion	Model	F1	Accuracy	Precision	Recall	FPR	ROC	Spec	MCC
Early Fusion	M1	0.1429	0.8481	0.0851	0.4444	0.1401	0.6498	0.8599	0.1423
	M2	0.2264	0.8057	0.1935	0.2727	0.1323	0.5779	0.8677	0.1212
	M3	0.2326	0.8436	0.1613	0.4167	0.1307	0.5957	0.8693	0.1871
	M4	0.2667	0.8436	0.1935	0.4286	0.1269	0.6956	0.8731	0.2121
	M5	0.1081	0.8436	0.0645	0.3333	0.1415	0.5865	0.8585	0.0901
	M6	0.08	0.8544	0.0426	0.6667	0.1438	0.7021	0.8562	0.1425
	M7	0.0625	0.8578	0.0323	1	0.1429	0.5601	0.8571	0.1663
	M8	0.1176	0.8578	0.0645	0.6667	0.1394	0.6151	0.8606	0.1763
	M9	0.1765	**0.8673**	0.0968	1	0.1346	0.7566	0.8654	0.2894
	M10	0.1212	0.8626	0.0645	1	0.1388	0.5813	0.8612	0.2357
Average Fusion	M1	0.2857	0.8576	0.2045	0.4737	0.1178	0.5839	0.8822	0.2443
	M2	0.303	0.673	0.4545	0.2273	0.1241	0.584	0.8759	0.1316
	M3	0.4	0.8436	0.3333	0.5	0.1164	0.6358	0.8836	0.3227
	M4	0.2963	0.8199	0.2424	0.381	0.1316	0.5847	0.8684	0.2055
	M5	0.2	0.8483	0.1212	0.5714	0.1422	0.5522	0.8578	0.2117
	M6	0.2759	0.8671	0.1818	0.5714	0.1192	0.5799	0.8808	0.2688
	M7	0.303	0.673	0.4545	0.2273	0.1241	0.584	0.8759	0.1316
	M8	0.3137	0.8341	0.2424	0.4444	0.1295	0.5931	0.8705	0.2422
	M9	0.2917	0.8389	0.2121	0.4667	0.1327	0.5836	0.8673	0.2363
	M10	0.2381	0.8483	0.1515	0.5556	0.1386	0.5645	0.8614	0.232
Max Fusion	M1	0	0.8608	0	0	0.1392	0.5	0.8608	0
	M2	0.1111	0.8483	0.0606	0.6667	0.149	0.5275	0.851	0.1687
	M3	0	0.8436	0	0	0.1564	0.5	0.8436	0
	M4	0	0.8436	0	0	0.1564	0.5	0.8436	0
	M5	0	0.8436	0	0	0.1564	0.5	0.8436	0
	M6	0	0.8608	0	0	0.1392	0.5	0.8608	0
	M7	0.0571	0.8436	0.0303	0.5	0.1531	0.5123	0.8469	0.0925
	M8	0	0.8436	0	0	0.1564	0.5	0.8436	0
	M9	0	0.8436	0	0	0.1564	0.5	0.8436	0
	M10	0.0588	0.8483	0.0303	1	0.1524	0.5151	0.8476	0.1603
Sum Fusion	M1	0	0.8608	0	0	0.1392	0.5	0.8608	0
	M2	0.1111	0.8483	0.0606	0.6667	0.149	0.5275	0.851	0.1687
	M3	0	0.8436	0	0	0.1564	0.5	0.8436	0
	M4	0	0.8436	0	0	0.1564	0.5	0.8436	0
	M5	0	0.8436	0	0	0.1564	0.5	0.8436	0
	M6	0	0.8608	0	0	0.1392	0.5	0.8608	0
	M7	0.0571	0.8436	0.0303	0.5	0.1531	0.5123	0.8469	0.0925
	M8	0	0.8436	0	0	0.1564	0.5	0.8436	0
	M9	0	0.8436	0	0	0.1564	0.5	0.8436	0
	M10	0.0588	0.8483	0.0303	1	0.1524	0.5151	0.8476	0.1603
Weighted Average	M1	0.125	0.8671	0.0682	0.75	0.1314	0.5323	0.8686	0.1998
	M2	0.1111	0.8483	0.0606	0.6667	0.149	0.5275	0.851	0.1687
	M3	0.4151	0.8531	0.3333	0.55	0.1152	0.6414	0.8848	0.3506
	M4	0.2381	0.8483	0.1515	0.5556	0.1386	0.5645	0.8614	0.232
	M5	0.2	0.8483	0.1212	0.5714	0.1422	0.5522	0.8578	0.2117
	M6	0.0444	0.8639	0.0227	1	0.1365	0.5114	0.8635	0.1401
	M7	0.1579	0.8483	0.0909	0.6	0.1456	0.5398	0.8544	0.1903
	M8	0.2162	0.8626	0.1212	1	0.1401	0.5606	0.8599	0.3228
	M9	0.2381	0.8483	0.1515	0.5556	0.1386	0.5645	0.8614	0.232
	M10	0.2105	0.8578	0.1212	0.8	0.1408	0.5578	0.8592	0.276

References

Adiba, F. I., Islam, T., Kaiser, M. S., & Mahmud, M. (2020). Effect of corpora on classification of fake news using naive Bayes classifier. *International Journal of Automation, Artificial Intelligence and Machine Learning, 1*(1).

Ajao, O., Bhowmik, D., & Zargari, S. (2018). Fake news identification on Twitter with hybrid CNN and RNN models. In *ACM international conference proceeding series*. http://dx.doi.org/10.1145/3217804.3217917.

Al-Ahmad, B., Al-Zoubi, A. M., Khurma, R. A., & Aljarah, I. (2021). An evolutionary fake news detection method for covid-19 pandemic information. *Symmetry*, *13*(6), http://dx.doi.org/10.3390/sym13061091.

Al-Rakhami, M. S., & Al-Amri, A. M. (2020). Lies kill, facts save: Detecting COVID-19 misinformation in Twitter. *IEEE Access*, *8*, http://dx.doi.org/10.1109/ACCESS.2020.3019600.

Allahverdipour, H. (2020). Global challenge of health communication: Infodemia in the coronavirus disease (COVID-19) pandemic. *Journal of Education and Community Health*, *7*(2), http://dx.doi.org/10.29252/jech.7.2.65.

Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, *31*(2), http://dx.doi.org/10.1257/jep.31.2.211.

Anoop, K., Gangan, M. P., D, P., & Lajish, V. L. (2019). *Leveraging heterogeneous data for fake news detection*. http://dx.doi.org/10.1007/978-3-030-01872-6_10.

Atrey, P. K., Hossain, M. A., el Saddik, A., & Kankanhalli, M. S. (2010). Multimodal fusion for multimedia analysis: A survey. *Multimedia Systems*, *16*(6), http://dx.doi.org/10.1007/s00530-010-0182-0.

Boididou, C., Middleton, S. E., Jin, Z., Papadopoulos, S., Dang-Nguyen, D.-T., Boato, G., & Kompatsiaris, Y. (2018). Verifying information with multimedia content on twitter. *Multimedia Tools and Applications*, *77*(12), http://dx.doi.org/10.1007/s11042-017-5132-9.

Burkhardt, J. M. (2017). How fake news spreads. *Library Technology Reports*, *53*(8).

Cui, L., & Lee, D. (2020). CoAID: COVID-19 healthcare misinformation dataset. http://arxiv.org/abs/2006.00885.

Cui, L., Wang, S., & Lee, D. (2019). Same: Sentiment-aware multi-modal embedding for detecting fake news. In *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining, ASONAM 2019*. http://dx.doi.org/10.1145/3341161.3342894.

Elhadad, M. K., Li, K. F., & Gebali, F. (2021a). An ensemble deep learning technique to detect COVID-19 misleading information. In *Advances in intelligent systems and computing, 1264 AISC*. http://dx.doi.org/10.1007/978-3-030-57811-4_16.

Elhadad, M. K., Li, K. F., & Gebali, F. (2021b). COVID-19-FAKES: A Twitter (Arabic/English) dataset for detecting misleading information on COVID-19. In *Advances in intelligent systems and computing, 1263 AISC*. http://dx.doi.org/10.1007/978-3-030-57796-4_25.

Ferrara, E., Cresci, S., & Luceri, L. (2020). Misinformation, manipulation, and abuse on social media in the era of COVID-19. *Journal of Computational Social Science*, *3*(2), http://dx.doi.org/10.1007/s42001-020-00094-5.

Figueira, Á., & Oliveira, L. (2017). The current state of fake news: Challenges and opportunities. *Procedia Computer Science*, *121*, http://dx.doi.org/10.1016/j.procs.2017.11.106.

Jin, Z., Cao, J., Guo, H., Zhang, Y., & Luo, J. (2017). Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In *MM 2017 - Proceedings of the 2017 ACM multimedia conference*. http://dx.doi.org/10.1145/3123266.3123454.

Jin, Z., Cao, J., Zhang, Y., & Zhang, Y. (2015). Verifying multimedia use with a two-level classification model. *vol. 1436*, In *MCG-ICT at mediaeval CEUR workshop proceedings*.

Kaliyar, R. K., Goswami, A., & Narang, P. (2021). A hybrid model for effective fake news detection with a novel COVID-19 dataset. *vol. 2*, In *ICAART 2021 - Proceedings of the 13th international conference on agents and artificial intelligence*. http://dx.doi.org/10.5220/0010316010661072.

Khattar, D., Gupta, M., Goud, J. S., & Varma, V. (2019). MvaE: Multimodal variational autoencoder for fake news detection. In *The Web Conference 2019 - Proceedings of the world wide web conference, WWW 2019*. http://dx.doi.org/10.1145/3308558.3313552.

Kishore Shahi, G., & Nandini, D. (2020). (n.d.) FakeCovid-A Multilingual Cross-domain Fact Check News Dataset for COVID-19. www.aaai.org.

Krishnamurthy, G., Majumder, N., Poria, S., & Cambria, E. (2018). A deep learning approach for multimodal deception detection. http://arxiv.org/abs/1803.00344.

Lago, F., Phan, Q. T., Boato, G., & Venčkauskas, A. (2019). Visual and textual analysis for image trustworthiness assessment within online news. *Security and Communication Networks*, *2019*, http://dx.doi.org/10.1155/2019/9236910.

Maigrot, C., Claveau, V., Kijak, E., & Sicre, R. (2016). A multimodal system for the verifying multimedia use task. *MediaEval CEUR Workshop Proceedings*, *1739*.

Majumder, S. B., & Das, D. (2020). Detecting fake news spreaders on Twitter using universal sentence encoder notebook for PAN at CLEF 2020.

Meel, P., & Vishwakarma, D. K. (2020). Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. *Expert Systems with Applications*, *153*, http://dx.doi.org/10.1016/j.eswa.2019.112986.

Meel, P., & Vishwakarma, D. K. (2021). HAN, image captioning, and forensics ensemble multimodal fake news detection. *Information Sciences*, *567*, http://dx.doi.org/10.1016/j.ins.2021.03.037.

Mookdarsanit, P., & Mookdarsanit, L. (2021). The covid-19 fake news detection in thai social texts. *Bulletin of Electrical Engineering and Informatics*, *10*(2), http://dx.doi.org/10.11591/eei.v10i2.2745.

Naeem, S. bin., Bhatti, R., & Khan, A. (2021). An exploration of how fake news is taking over social media and putting public health at risk. *Health Information and Libraries Journal*, *38*(2), http://dx.doi.org/10.1111/hir.12320.

Narwal, B. (2018). Fake news in digital media. In *Proceedings - IEEE 2018 international conference on advances in computing, communication control and networking, ICACCCN 2018*. http://dx.doi.org/10.1109/ICACCCN.2018.8748586.

Orso, D., Federici, N., Copetti, R., Vetrugno, L., & Bove, T. (2020). Infodemic and the spread of fake news in the COVID-19-era. *In European Journal of Emergency Medicine*, http://dx.doi.org/10.1097/MEJ.0000000000000713.

Pogorelov, K., Schroeder, D. T., Burchard, L., Moe, J., Brenner, S., Filkukova, P., & Langguth, J. (2020). Fakenews: Corona virus and 5G conspiracy task at MediaEval 2020. *vol. 2882*, In *CEUR Workshop Proceedings*.

Saini, N., Singhal, M., Tanwar, M., & Meel, P. (2020). Multimodal, semi-supervised and unsupervised web content credibility analysis frameworks. In *Proceedings of the international conference on intelligent computing and control systems, ICICCS 2020*. http://dx.doi.org/10.1109/ICICCS48265.2020.9121005.

Shim, J. S., Lee, Y., & Ahn, H. (2021). A link2vec-based fake news detection model using web search results. *Expert Systems with Applications*, *184*, http://dx.doi.org/10.1016/j.eswa.2021.115491.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media. *ACM SIGKDD Explorations Newsletter*, *19*(1), http://dx.doi.org/10.1145/3137597.3137600.

Shu, K., Zhou, X., Wang, S., Zafarani, R., & Liu, H. (2019). The role of user profiles for fake news detection. In *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining, ASONAM 2019*. http://dx.doi.org/10.1145/3341161.3342927.

Singh, V. K., Ghosh, I., & Sonagara, D. (2021). Detecting fake news stories via multimodal analysis. *Journal of the Association for Information Science and Technology*, *72*(1), http://dx.doi.org/10.1002/asi.24359.

Singh, M., Kaur, R., & Iyengar, S. R. S. (2020). Multidimensional analysis of fake news spreaders on Twitter. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), 12575 LNCS*. http://dx.doi.org/10.1007/978-3-030-66046-8_29.

Singhal, S., Shah, R. R., Chakraborty, T., Kumaraguru, P., & Satoh, S. (2019). SpotFake: A multi-modal framework for fake news detection. In *Proceedings - 2019 IEEE 5th international conference on multimedia big data, BigMM 2019*. http://dx.doi.org/10.1109/BigMM.2019.00-44.

Vishwakarma, D. K., & Jain, C. (2020). Recent state-of-the-art of fake news detection: A review. In *2020 International conference for emerging technology, INCET 2020*. http://dx.doi.org/10.1109/INCET49848.2020.9153985.

Wang, Y., Ma, F., Jin, Z., Yuan, Y., Xun, G., Jha, K., Su, L., & Gao, J. (2018). EANN: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining*. http://dx.doi.org/10.1145/3219819.3219903.

Xiao, L., Wang, G., & Zuo, Y. (2018). Research on patent text classification based on Word2Vec and LSTM. *vol. 1*, In *Proceedings - 2018 11th international symposium on computational intelligence and design, ISCID 2018*. http://dx.doi.org/10.1109/ISCID.2018.00023.

Yang, Y., Zheng, L., Zhang, J., Cui, Q., Li, Z., & Yu, P. S. (2018). TI-CNN: Convolutional neural networks for fake news detection. http://arxiv.org/abs/1806.00749.

Zhou, X., Mulay, A., Ferrara, E., & Zafarani, R. (2020). ReCOVery: A multimodal repository for COVID-19 news credibility research. In *International conference on information and knowledge management, proceedings*. http://dx.doi.org/10.1145/3340531.3412880.