A survey of uncover misleading and cyberbullying on social media for public health

Omar Darwish1 · Yahya Tashtoush2 · Amjad Bashayreh2 · Alaa Alomar2 · Shahed Alkhaza’leh2 · Dirar Darweesh2

Received: 18 July 2022 / Revised: 18 July 2022 / Accepted: 8 August 2022 / Published online: 23 August 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Misleading health information is a critical phenomenon in our modern life due to advances in technology. In fact, social media facilitated the dissemination of information, and as a result, misinformation spread rapidly, cheaply, and successfully. Fake health information can have a significant effect on human behavior and attitudes. This survey presents the current works developed for misleading information detection (MLID) in health fields based on machine learning and deep learning techniques and introduces a detailed discussion of the main phases of the generic adopted approach for MLID. In addition, we highlight the benchmarking datasets and the most used metrics to evaluate the performance of MLID algorithms are discussed and finally, a deep investigation of the limitations and drawbacks of the current progressing technologies in various research directions is provided to help the researchers to use the most proper methods in this emerging task of MLID.

Keywords Misleading information · Misinformative · Deep learning · Machine learning · Disinformation · COVID-19 · BERT

1 Introduction
Misleading information is a critical problem that affects controversial topics such as human health. Misinformation lacks scientific evidence since the information exchange is primarily happened by a clash of disparate and emotionally charged narratives. Social media networks such as Facebook, Twitter, and Instagram are widely regarded as one of the most essential ways for people to communicate [1].

In the last decade, social media has evolved into a significant instrument for gathering information and developing solutions in a variety of disciplines, including business, entertainment, and crisis management in health care, research, and politics [2].

Social media is regarded as one of the most important sources of information for health monitoring [3, 4]. Open source to the health information on the Internet and other easy-to-access online databases has made up new opportunities for medical professionals, but most of the information is susceptible to manipulation because contextual conventions and the veracity of provenance are constantly questioned. False health information is based on opinions,
mythologies, and oral histories. Any post, tweet, or shared resource that misinterprets the knowledge of medicine accepted by experts is considered misleading health information. This contains fake news articles, memes, and posts about things that are false.

The problem of obtaining health information from social media is determining what is reliable. That was one of the main topics of “Freedom of Information,” a BioMed Central-sponsored conference held on the 6th and 7th of July at New York’s Academy of Medicine. One of the attendees mentioned a widely publicized study published in the professional journal Cancer. J Sybil Biermann and her colleagues at the University of Michigan discovered that one webpage documented the mortality rate for a particular type of bone cancer as 5%, when in fact it was closer to 75%. According to the attendee, such misinformation could be disastrous. The term “misleading Information” is defined in the literature as fake news, misinformation, disinformation, false news, or non-informative news which is published to mislead the users and make them realize that it is truthful and reliable. According to the Federal Trade Commission, more than 25 million individuals will use the Internet to search for health information. The number of medically related websites on the Internet is estimated to be at least 100,000 doctors only review about half of these sites’ content [5].

There are two main types of misleading information on social media: “Disinformation,” which is purposefully spread by nefarious people. The other type is “Misinformation,” which spreads unintentionally but is still false. Also if we look at the psychological side of the problem, we can say that the so-called online cyberbullying, abusive language and hate speech is one of the most important aspects of the problem. According to the above mentioned, the urgency of detecting this type of information was first felt by researchers. Therefore, they started using various types of machine learning (ML) and deep learning (DL) algorithms to help people distinguish between informative and non-informative information by collecting datasets from public posts on social media sites, first and foremost from Twitter, and conducting their scientific experiences on them.

The main contributions of this comprehensive study are:

(1) Presenting some statistics for the previous work that demonstrate the interest of researchers in the MLID field and provide recommendations for future works.
(2) Introducing a detailed discussion of the main phases of this framework to show the generic adopted approach that is used by the plurality of proposed MLID techniques.
(3) Presenting the main ML and DL methods utilized in the literature to review the recent MLID approaches.
(4) Discussing the benchmarking datasets and the most used metrics to evaluate the performance of MLID algorithms.
(5) Providing a deep investigation of the limitations and drawbacks of the current progressing technologies in various research directions to help the researchers to use the best methods in this emerging task of MLID.

The paper is organized as follows: Sect. 2 introduces the research scope and some statistics for the previous works, Sect. 3 describes the standard approach of MLID, and Sect. 4 presents and summarizes the latest methods applied in MLID, Sect. 5 discusses the benchmarking datasets used during MLID, Sect. 6 shows the most used metrics to evaluate the performance of MLID algorithms, and Sect. 7 highlight the main limitations and drawbacks in MLID. Section 8 concludes this research study.

## 2 Research scope

Social media websites permeate our culture by evolving into ideal platforms for information access. As a result, these platforms have an important role in persuading people’s decisions and opinions. Everywhere now, most people spend hours on social websites to communicate with the rest of the world. Also, they use social platforms to read news and gather information rather than conventional media because information dissemination on social networks takes a shorter time and lower cost. Misleading news is a major problem and a global issue since it has become a daily expression and an essential component of any media discourse over the last 4 years. It has huge popularity among stakeholders such as non-governmental organizations, journalists, politicians, civil society and researchers. Furthermore, misinformation has significant repercussions, such as influencing people’s decisions, opinions, and attitudes. As a result, the detection of fake information is critical. Fake information is typically associated with time-sensitive and evolving events that cannot be accurately verified by previous studies. On the flip side, data generated by misinformation are noisy, unorganized, and incomplete [6].

There are a variety of human-based websites that introduce pre-checked data for misleading information detection (MLID). These data are analyzed manually by expert analysts knowledgeable about this research topic. The most popular websites are Snopes.com, PolitiFact.com, and Factcheck.org. Due to the huge amount of data on social media, the human approach becomes lifeless, costly, subjective, and unworkable [7–9].

On the other hand, the technical approach classifies the data automatically using numeric machines that make the
process easier, less expensive, and more efficient. The Internet is a massive repository of information. With the advent of technology, every user has become a self-publisher, with no proofreading, no fact-checking, and no accountability. People have the freedom to publish anything they want, when they want, and wherever they want. Due to the lack of verifiability, users can spread their opinions across social media platforms. These opinions represent many types of misleading information that have been listed in Table 1 [10].

We generated a list of the most recent research studies on the misleading information task from (2015 to 2022) as shown in Fig. 1. We extracted the results from the Association for Computing Machinery (ACM) explorer using “misleading information” “fake news” “misinformation” and “disinformation” keywords.

Also, we evaluated the publishers’ interest in this research topic and got the top three names of journals and conferences that are famous in this field. Concerning journals and magazines, the rapid evaluation of misleading information topics is highlighted in Proceedings of the ACM on Human–Computer Interaction with 90 articles, Communications of the ACM with 69 articles, and ACM Computing Surveys with 26 articles on this topic.

As for Proceedings (Conferences), Proceedings of the ACM Web Conference 2022 published 28 papers, Companion Proceedings of The Web Conference 2018 had 26 papers, and Companion Proceedings of The 2019 World Wide Web Conference published 26 papers.

Our research study differs from previous surveys in that it addresses a comparison between the latest studies of MLID and provides a comprehensive overview of previous developments and algorithms that will introduce a deep understanding to inspire the researchers to use the appropriate methods that would improve their contributions in this field. It concentrates on the MLID approach’s ML and DL techniques, architectures, and models.

In addition, benchmarking datasets and evaluation metrics are discussed. Many limitations and thought-provoking research directions are presented in detail. This research study will motivate the researchers to add more improvements to the task of MLID.

3 The MLID approach

This section demonstrates the MLID phases, beginning with the stages through which all training data should pass before being used to create misleading-information detection models.

The following stages make it easier to handle the large amount of data required to build a detection model. As shown in Fig. 1, the generic methodology is primarily based on ML and DL models, which are widely used to learn distinguishing characteristics of misleading information.

As seen in the pipeline, several critical steps are typically implemented in the development of the final detection system. The summary of MLID approaches are also explained in Table 2.

3.1 Data preprocessing and feature engineering

3.1.1 Preprocessing

Text preprocessing is a technique for cleaning up text data to feed it into a model. Text data includes different types of noise such as emotions, emojis, punctuation, characters, and text in different scenarios. Hence, there are numerous ways to express the same concept, and this is only the most serious problem we face because machines don’t understand words; they require numbers, so we must encode and transform the text into numbers to pass it to all subsequent processing stages [11].

| Type                  | Description                                                                                                                                 |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Fabricated content    | Totally fake content                                                                                                                      |
| Manipulated content   | Real information or imagery is distorted. For example, a headline is created to provoke public interest, which is often promoted by ‘clickbait’ |
| Imposter content      | Impersonation of authentic sources, such as using the branding of a well-known news organization                                             |
| Misleading content    | Misleading information use, such as presenting a statement as a reality                                                                      |
| False context of connection | Content that is factually correct but is attended by fake contextual information, such as when an article’s headline does not accurately reflect the content, such as when an article’s headline does not accurately reflect the content |
| Satire and parody     | Presenting amusing but fake information as if it were real even though not typically classified as disinformation, this may unintentionally mislead readers |
This stage is critical for reducing the indexing (or data) file size of the text documents and improving the IR system’s quality and productivity [12].

The most used techniques for data cleaning are lower case, expand contractions, remove punctuation, words, and digits, remove stop-words. Text rephrasing, lemmatization, and stemming, and white spaces removing. Contraction is the shortening of a word; for example, veggie stands for Vegetarian, and limo stands for Limousine.

For better analysis, we must broaden this compression in the text data. Lower case is important because a machine can easily interpret the words when the text is in the same case since the machine treats the lower and upper cases differently.

Another text processing technique is punctuation removal. There are a total of 32 main punctuations that must be addressed. The string module can be used with a regular expression to replace every punctuation in the text with an empty string. Some people tend to write characters and digits together to form words in the text, such as amjad12 or amjad37bash. This type of word must be eliminated because it causes a problem for machines to understand and process.

As a result, the best solution is to remove this type of text or replace it with an empty string. Stop-words such as these are common words in a text that do not provide any helpful information. Therefore, the researchers use the NLTK library to remove these types of words. The process of minimizing a word to its root stem is known as stemming. There are numerous stemming algorithms, such as the porter stemmer and the snowball stemmer. Porter stemmer is a popular tool in the NLTK library. The stemming method is not used in production because it is inefficient, and most of the time it stems from unwanted words.

As a result, another technique known as Lemmatization entered the market to solve the problem. Lemmatization is comparable to stemming. since it is used to stem words into root words, but it works differently. Lemmatization is a systematic method of decreasing words to their lemma by comparing them with a language glossary [12].

Also, the researchers tend to use word embedding as an essential part of the preprocessing step to represent the words that will be used in the text analysis process. Word embeddings are working to form real-valued vectors that encode the meaning of the words. As a result, the words that have similar meanings will be closer in the vector space [13]. Our suggested framework is presented on Fig. 2.

3.1.2 Feature selection

Even though there are many current classifiers for text categorization, the large dimensionality of the feature space is a serious difficulty [14].

A document typically comprises hundreds or thousands of distinct words that are considered features; however, many of them may be noisy, less useful, or redundant in terms of class labels. This may cause classifiers to be misled and, as a result, decrease their overall performance [15, 16].

As a result, feature selection must be utilized to eliminate noisy, less useful, and redundant features, reducing the feature space to a manageable level and boosting the efficiency and accuracy of the classifiers used. A feature selection approach, in general, consists of four basic steps: feature subset generation, subset evaluation, halting condition, and classification result confirmation [17].

The researchers employ a search approach to select a candidate feature subset in the first phase, which is then evaluated using a goodness criterion in the second step. When stopping requirements are met in the third phase, subset formation and evaluation are terminated, and the best feature subset from all candidates is picked. The feature subset will be validated using a validation set in the last stage. Feature selection approaches are classified into
| References | Model | Method | Specifications | Dataset | Size (text) | Topic |
|------------|-------|--------|----------------|---------|-------------|-------|
| [82]       | SVM, KNN, RF, NB, DistilBERT, DistilRoBERTa | Cost-weighting settings | NVIDIA Tesla V100S 32 GB GPU, 240 GB of storage, 32 GB of RAM, an Intel(R) Core(TM) i7-9750H CPU | CLEF 2018 Consumer Health Search | 5,535,120 | Public Health |
| [83]       | KG-Miner TransE, text-CNN, CSI, dEFEND, GUpdater, HGAT, DETERRENT | Knowledge guided graph attention network | An attention mechanism, the knowledge guided article embeddings | Diabetes Cancer | 2269 6099 | Public Health |
| [92]       | RF | Recursive feature selection | Linguistic features, LIWC feature | Collected dataset | 2225 | Public Health |
| [95]       | MLP, NN, SVM, RT, MAAd and RF | The graph theory and the social influence models | Identifies two structural levels (named as user level and network level in the following) that lead to the extraction of representative feature | Collected dataset | 709 | Public Health |
| [96]       | GB, LR, NB, RF, BiLSTM, CNN | L2 regularization | Textual representation, linguistic–stylistic, linguistic emotional, linguistic–medical, propagation-network, and user-profile features | CoAID, ReCOvery, FakeHealth (release) FakeHealth (Story) | 3555 2029 606 1690 | Public Health |
| [100]      | DT, kNN, MNB, NN, BNB, L SVM, LR, ERF, XGBoost | Voting ensemble | Feature Engineering, TF-IDF, N-gram | Collected dataset | 7486 | COVID-19 |
| [101]      | SBERT, BiLSTM, SBERT (DA) | BiLSTM and SBERT are jointly trained with the linear classifier | Bi-encoder, forward/backward LSTM, concatenate flatten | COVIDLIES, SNLI, MultiNLI MedNLI | 6761 570,000 433,000 14,049 | COVID-19 |
| [106]      | Random Forest, DT | Two RF and combination of multiple decision trees | TF-IDF, Count vectorizer, Bag of word | COVID19FN | 2800 | COVID-19 |
| [109]      | DT, kNN, SVM, RF, NB, LSTM, LR, GRU | Grid search with cross validation, Keras_tuner | TF-IDF, N-gram, word embedding | CoAID PolitiFact Disasters Gossip cop | 926 1050 7613 10,650 | COVID-19 |
| [108]      | SVM, PAC, MLP, LSTM with FastText, CNN with FastText, LSTM + CNN, BiLSTM + Attention, Ensemble Model (BERT, ALBERT, XLNet) | Transformer model using HuggingFace library | English Glove word embeddings, PyTorch transformer library, Transformer-XL | Collected dataset | 10,700 | COVID-19 |
| [112]      | BERT | Cluster analysis | Special character removal, stemming and lemmatization, TF-IDF | Collected dataset | 6731 | COVID-19 |
| [116]      | kNN–BSSA, kNN–BGA with feature selection, kNN–BPSO, and kNN | Wrapper feature selection methods | Reduce the number of features, TF, TF-IDF, and bag-of-words | Koirala | 3002 | COVID-19 |
| [118]      | Hybrid model (CNN and LSTM) | Hybrid model | TF-IDF, word embedding and hyperparameter optimization | Dataset1 Dataset2 Dataset3 | 1100 10,202 3001 | COVID-19 |
| [119]      | DT (C4.5), RF, NB, SVM, kNN, Bayes Net + kNN | Stacking method | Data annotation and feature extraction | Collected dataset | 409,484 | COVID-19 |
| References | Model | Method | Specifications | Dataset | Size (text) | Topic |
|------------|-------|--------|----------------|---------|-------------|-------|
| [120]      | LSTM, BiLSTM, CNN, hybrid of LSTM-CNN | GloVe pre-trained word embedding features | Python regular expressions and NLTK | COVID-19 Fake News | 21,379 | COVID-19 |
| [122]      | Attention-based BiLSTM-CRF model | Conditional Random Field (CRF) | PubMed pretrained embeddings | Collected dataset | 20,137 | Cancer |
| [124]      | SVM | LinearSVC class based on the LibLinear library | User engagement features | Collected dataset | 250 | Cancer |
| [125]      | LSTM | GloVe, Word2Vec (CBOW and skip-gram), FastText (CBOW and skip-gram) | Word embeddings | Collected dataset | 140,000 | Influenza |
| [129]      | SVM, RF, RUSBoost, XGBoost, CNN | Sequence Alignment-free Methods | Word encoding, word embeddings | GI-SAID | 60,087 | Influenza |
| [131]      | DNN, SVM, J48, NB | CFS reduction method | WEKA, Sklearn library | Collected dataset | 7000 | Heart Attack |
| [133]      | RF, NB, J48, NN | Session-based model | User-based features, text-based features, and network-based features | Collected dataset | 1.6M | Cyberbullying (Psychological Health) |
| [134]      | RF, NB, J48 | SentiStrength, Indico API | Big Five, Dark Triad models, psychological features such as personalities, sentiment, and emotion | Collected dataset | 9484 | Cyberbullying (Psychological Health) |
| [135]      | RF + Big Five and Dark Triad Models | Ensemble technique | Big Five, Dark Triad models to determine user personality | Collected dataset | 9484 | Cyberbullying (Psychological Health) |
| [138]      | LR, SVM | SVM with two new hypothesis for feature extraction | N-gram, Counting, TF-IDF Score and hypotheses for feature extraction (Capturing pronouns, Skip-grams) | Collected dataset | 6547 | Cyberbullying (Psychological Health) |
| [139]      | kNN, SVM, NB, DT, RF | SMOTE technique | Network-Based Features, Activity Features, User Features, Content-Based Features, Personality Features, Pointwise Mutual Information-Semantic Orientation (PMI-SO) | Collected dataset | 14,495 | Cyberbullying (Psychological Health) |
| [141]      | NB, LibSVM, RF, kNN | SMOTE technique | Network features, Activity features, User features, Content features, Pearson correlation, chi-square test, and information gain | Collected dataset | 10,007 | Cyberbullying (Psychological Health) |
| [142]      | CapsNet–ConvNet | Hybrid deep technique of Capsule network (CapsNet) and convolution neural network (ConvNet) | Google Lens of Google Photos App | Collected dataset | 10,000 | Cyberbullying (Psychological Health) |
| [143]      | NB | NB Lexicon Based features | Bag of Word features and Lexicon Based features | Collected dataset | 350 | Cyberbullying (Psychological Health) |
four types based on how they create feature subsets: filter model [18–20], wrapper model [21, 22], embedding model, and hybrid model [23, 24]. Most feature selection approaches for text classification are filter-based, owing to their simplicity and efficiency. [15, 16, 25, 26] provide a detailed analysis and comparison

| References | Model | Method | Specifications | Dataset | Size (text) | Topic |
|------------|-------|--------|----------------|---------|-------------|-------|
| [144]      | BoW, sBoW, LSA, LDA, EBoW | BoW and latent semantic features | BoW features, latent semantic features and bullying features | Twitter dataset | 1762 | Cyberbullying (Psychological Health) |
| [149]      | Bi-LSTM , CNN, Bi-LSTM with attention, CNN–LSTM combined | Automatically identifying abusive language on Arabic | For Bayesian hyperparameter optimization, the Structured Parzen Estimator (TPE) algorithm is used | Collected dataset | 15,050 | Cyberbullying (Psychological Health) |
| [151]      | SVM, NB | WEKA | Tweet To Senti Strength Feature Vector filter | Arabic dataset, English dataset | 35,273 | Cyberbullying (Psychological Health) |
| [154]      | NB, LR, SVM, XGBoost, CNN, LSTM, BLSTM and GRU | Youtube API | TF-IDF, Word Embedding, Sklearn, Tensorflow, NLTK, matplotlib | Dataset1 Dataset2 Dataset3 | 5000 7000 12,000 | Cyberbullying (Psychological Health) |
| [156]      | SVM, Logistic Regression, NB, Random Forest | Ensemble | Label Encoder | Dataset1, OLID, Dataset3, Dataset4 | 1990 14,100 8817 24,784 | Cyberbullying (Psychological Health) |
| [157]      | GBDT, Random Forest, SVM, XGB_CTD | Fuzzy C-Means (FCM) | Scikit-learn, Keras, FCM and XGBoost Libraries, Intel i7-8500H 3.60 GHz CPU and a laptop with a 12 GB RAM | Collected dataset | 542 | Cyberbullying (Psychological Health) |
| [158]      | FFNN | 4 Hidden layers | Hot encoding | Dataset1 Dataset2 | 4913 34,890 | Cyberbullying (Psychological Health) |
| [159]      | NB | Predefined list | BoW | Training Testing | 1,600,000 359 | Cyberbullying (Psychological Health) |
| [161]      | SVM, CNN-CB | Twitter streaming API | Spyder environment, 12 GB of RAM | Collected dataset | 39,000 | Cyberbullying (Psychological Health) |
| [162]      | CNN, LRCN | Skip-gram | four NVIDIA GTX 1080 servers | Collected dataset | 8815 | Cyberbullying (Psychological Health) |
| [163]      | SVM, DT (C4.5), NB, and kNN | Information gain and chi-square | Tokenization | Collected dataset | 900 | Cyberbullying (Psychological Health) |
| [164]      | CNN and PCNN | TM (threshold moving), CFA (cost function adjusting), and a hybrid solution (TM CFA) | Removing non-alphanumeric characters tokens | Dataset1 Dataset 2 | 1313 13,000 | Cyberbullying (Psychological Health) |
| [166]      | SVM | n-gram | Tokenization, Normalization | Collected dataset | 15,050 | Cyberbullying (Psychological Health) |
of alternative feature selection techniques for generic data [27].

3.1.3 Feature extraction

In-text categorization, there are two main types of feature extraction methods: n-gram and termset.

**n-Gram** the n-gram extraction procedure requires sliding across an entire corpus with a window of length n [28–31].

Then, in each window, all the sets of consecutive words or characters should be extracted. The goal of the n-gram is to obtain the composite features that emerge repeatedly to reduce the ambiguity of individual words. Bigram and trigram are two often used n-grams. Nonetheless, the impact of text structure, such as punctuation and stopwords, is not considered.

**Termset** a termset is not the same as an n-gram in that composite characteristics are retrieved solely based on their co-occurrence, regardless of the order and position of the individual terms [28, 32, 33]. Specifically, termsets are randomly matched combinations of vocabulary. However, this combination has one drawback: a combination explosion, even for 2-termsets. It indicates that with a vocabulary size of n, there will be 2n different types of pairings.

3.2 Data splitting

When starting a modeling project, one of the first decisions to make is how to use the available data. One popular method is to divide the data into training set and testing set. Models and feature sets are created using the training set. The training set serves as the foundation for parameter estimation, model comparison, and all other operations required to arrive at a final version of the model. At the end of these operations, the testing set is used to estimate a final, unbiased evaluation in terms of the model’s effectiveness. Looking at the results of the test sets would skew the results because the testing data would have become part of the outcome. How much data should be kept for testing purposes? It is hard to create a consistent guideline.

Many factors can influence data proportion, including the length of the initial sample pool and the actual amount of predictors. The importance of this decision is reduced
with a large pool of samples once the training set contains sufficient samples. Alternative options to a simple original data partition may also be a smart idea in this situation [34].

### 3.3 Model selection and building

Plenty of well methods have been proposed to detect misleading information. The researcher used DL methods such as BERT [35], SBERT [36], ALBERT [37], LSTM [38], BI-LSTM [39], BSSA [40], SSA [41], CNN [42], and ML models such as SVM [43], DT [44], RF [45], NB [46], kNN [47], XGBoost [48], and GA [49].

#### 3.3.1 BERT

The BERT framework consists of two steps: pre-training and fine-tuning. On unlabeled data, the model is trained across many pre-training activities during the pre-training phase. To fine-tune the BERT model, begins with the pre-trained parameters. The parameters are then fine-tuned utilizing labeled data from downstream jobs. Despite the fact that they are all started with identical pre-trained parameters, each downstream job has its own fine-tuned model. BERT’s model structure as described in the original implementation in [50] is a bidirectional multi-layer transformer encoder available in the tensor2tensor library.

The BERT was trained on two activities: language modeling (15% masked tokens predicted from the context by trained BERT) and the following statement prediction (trained BERT was trained used to predict whether a selected following statement was possible or not depending on the first statement).

BERT learns word contextual embeddings as a result of the training procedure. Following the computationally expensive pretraining, BERT can be finetuned with fewer dataset resources with small sizes to enhance its performance on detailed tasks [35, 51].

#### 3.3.2 CNN

A convolutional neural network (CNN) is used for image processing [52] but in other circumstances, researchers prefer to work with TextCNN for text classification, which necessitates the addition of a word embedding layer and a one-dimensional convolutional network to the model’s original structure [53].

CNN is made up of three layers: an input layer, hidden layers, and an output layer. A feed-forward neural network’s intermediary layers are referred to as hidden because their model parameters (model’s input and model’s outcomes) are veiled by the final convolution and activation function.

The hidden layers of a CNN include convolutional layers. Generally, this comprises a layer that does a dot product of the convolution kernel and the input matrix of the layer.

This is typically the inner product of Frobenius and its activation function is typically ReLU. As the convolution kernel slipped along the input matrix for the layer, a feature map is generated, which then plays an important role to the following layer’s input. Following that, additional layers such as fully connected layers, normalization layers, and pooling layers are added. [54].

#### 3.3.3 RNN

A recurrent neural network (RNN) connects the nodes to create a graph that follows a time sequence. As a result, it can show dynamic behavior. RNNs, which are derived from feedforward neural networks, can use their internal memory to perform variable-length input sequences [55–57].

Consequently, they can be used for tasks like unsegmented, handwriting recognition [58], or speech recognition [59, 60]. RNNs are turning complete in theory and may execute arbitrary programs to perform arbitrary input data [61].

Fully RNNs (FRNN) connect all neurons’ outputs to all neurons’ inputs. Because all other topologies may be replicated by reducing the weights of some connections to zero to imitate the lack of connections between particular neurons, this is the most general neural network architecture.

Many people may be misled by the figure on the right because practical neural network topologies are typically organized in “layers,” and the illustration conveys that impression. What looks to be layered, however, are various time steps of the same FRNN. The arc labeled ‘v’ on the left-hand side of the illustration depicts the recurring connections. It “unfolds” in time to provide the illusion of layers [62].

#### 3.3.4 LSTM

LSTM is an abbreviation for long-short term memory. In terms of memory, LSTM is a sort of RNN that outperforms traditional RNNs. LSTMs perform far better when it comes to learning specific patterns.

LSTM, like any other NN, can have several hidden layers, and as it travels through each layer, the relevant information is retained while the irrelevant information is discarded in each cell [63].

A cell, an input gate, an output gate, and a forget gate comprise the LSTM model. The cell retains values over
arbitrary time intervals, and the three gates control data flow in or out of the cell [64].

### 3.3.5 GRU

GRU is an abbreviation for gated recurrent unit, which is a method of gating in RNNs proposed by [65].

The GRU functions similarly to a LSTM with a forget gate [66], but with fewer parameters because it is missing an output gate. [67].

GRU outperformed LSTM on specific tasks such as natural language processing (NLP), polyphonic music modeling, and speech signal modeling [68, 69].

GRUs have been found to outperform other methods on unduplicated datasets of small size [70, 71].

### 3.4 Machine learning models

#### 3.4.1 Decision tree (DT)

A DT is a data flow structure that contains nodes that represent the test attributes. Each branch in the DT reflects the result of the test, and each leaf node provides a class label (decision reached after calculating all attributes). Classification rules are defined by the pathways from root to leaf.

As a visual and analytical decision assistance tool, a DT and its closely related effect diagram are used in decision analysis, where competing alternatives’ expected values are determined [72].

#### 3.4.2 Random forest (RF)

Random decision forests are another name for RFs that are well-suited for coping with high-dimensional noisy data in text classification [73].

RF is an ensemble learning method that can be used for classification, regression, and other issues that work by generating a plethora of DTs during training. For classification problems, the class selected by the most of trees is the random forest output. For regression problems, the mean or average prediction of each tree is returned [74, 75].

#### 3.4.3 Naive Bayes (NB)

NB is a classification algorithm based on Bayes’ Theorem and the assumption of predictor independence.

A NB classifier, in simple terms, assumes that the presence of one feature in a class is independent of the values of the other features. NB has three types: Gaussian, Multinomial, and Bernoulli [76].

#### 3.4.4 Support vector machine (SVM)

SVM is a supervised learning system with an accompanying algorithm that is used for classification issues.

Each data object is represented graphically in n-dimensional space in this procedure, with the value and coordinate determining the item’s value.

Classification is accomplished by determining the hyper-plane that best divides the categories. SVMs may conduct non-linear classification by converting inputs straight into high-dimensional feature spaces [77–79].

#### 3.4.5 K-nearest neighbor (kNN)

kNN is among the most basic and extensively used algorithms that is based on its k value. The k value is used to specify how many neighbors in the algorithm.

This technique works by selecting the number k value as the first step. Then it uses Euclidean distance to determine the kNN of a new data point. Following that, it counts how many data points there are in each category among the K-neighbors before assigning the new data point to the category with the most neighbors [80, 81].

### 4 Literature review

With the advancement of technology and social media addiction, avoiding misleading information becomes an essential part of our daily life due to the easy dissemination of any information especially when it comes to the health fields.

This section presents the latest works proposed for MLID regarding public health fields such as COVID-19, cancer disease, influenza, heart attacks, and psychological health (cyberbullying).

We selected these research papers because they were of the best research in the MLID field from different aspects such as high results, benchmarking datasets usage, obtaining various types of algorithms, and well documentation.

#### 4.1 Public health

Fernández-Pichet et al. [82], provided a comprehensive comparison of recent deep NLP models, such as newer BERT-based models (DistilBERT, DistilRoBERTa), and traditional algorithms, such as SVM, RF, NB, and kNN, for detecting health-related misinformation, and identifying low-quality online content (web pages that are unreliable and hard to read). The CLEF 2018 Consumer Health Search task dataset was chosen by the authors. To compare the mentioned models, the researchers examined them in
terms of trustworthiness, readability, usefulness (both trustworthiness and readability), and the effect of the training set size. Yet, SVM, RF, and DistilRoBERTa achieved the highest F1-score (93%).

Cui et al. [83], proposed the DETERRENT approach, which uses a relational graph attention network to represent various positive and negative relationships in the medical knowledge graph. The authors manually construct two datasets, diabetes and cancer, using a public medical knowledge network KnowLife3 [84], which contains 25,334 entity names and 591,171 triples. The authors collect six positive and four negative relationships. Also, the authors compared DETERRENT with KG-cutting-edge Miner’s misinformation detection algorithms [85], TransE [86], text-CNN [87], CSI [88], deFEND [89], HGAT [90], GUpdater [91]. After the comparison, DETERRENT got the highest score in the cancer dataset with an accuracy of 0.9652, precision of 0.9469, recall of 0.9153 and F1-score of 0.9309.

Kinsora et al. [92], created a medical misinformation-labeled dataset to build ML classifiers for detecting false medical information automatically. This dataset was derived from a dataset created by Vydiswaran et al. [93]. The dataset is a combination of disinformation and correct facts collected from online health forum comments consisting of 2225 comments from MedHelp were labeled as misinformative or non-misinformative. The authors accomplished this by employing information retrieval techniques such as linguistic inquiry and word count (LIWC) psycho-linguistic lexicon feature [94] to obtain a wide variety of natural language features. They later devised a coding technique for labeling and annotating the dataset. The scientists created a classifier that can recognize medical misinformation with an accuracy of 90.1% by using nine designed characteristics in the produced dataset.

Sicilia et al. [95], provided a method for detecting rumor in each post in a specific topic area connected to health news. The authors introduced new descriptors influenced by graph theory and social influence models, such as the likelihood of a tweet being retweeted or a URL being shared, conversation size, a fraction of users’ followers of a root, and a fraction of tweets with URLs. The authors gathered the dataset from Twitter using the hashtags #zikavirus and #zikamicrocephaly, which this was one of the major healthcare trends in 2016. On February 1st, 2016, the World Health Organization (WHO) classified Zika virus disease as a Public Health Emergency of International Concern (PHEIC). The authors used multiple classifiers including multi-layer perceptron (MLP), nearest neighbour (NN), SVM, random tree (RT), multiclass Adaboost (MAa), and RF as an ensemble of trees and obtained the following results: the overall accuracy is equal to 73.63%, the average precision per class is 72.80%, the average recall per class is 73.60% and the average AUC per class is 89.00%.

Di Sotto and Viviani [96], introduced an analysis of unique characteristics groups and ML algorithms that can be useful in assessing misinformation in online health-related material, in the form of websites or social media content. The authors used three well-known datasets; CoAID (COVID-19 heAlthcare mlsinformation Dataset) [97], ReCOVery [98], and FakeHealth [99]. The authors identify six types of health misinformation textual representation features, linguistic–stylistic features, linguistic emotional features, linguistic–medical features, propagation-network features, and user-profile features. In particular, several ML and DL algorithms have been considered. The results show that DL solutions are efficient when word embedding selected features from appropriate medical vocabulary training are used, without the need for other types of features. When “classical” ML classifiers are used, however, the impact of considering other types of features grows. The best results in the CoAID are obtained by CNN. AUC equals 0.973, and F-measure of 0.953.

4.2 COVID-19 misinformation

With the COVID-19 pandemic, Elhadad et al. [100] developed a model for misleading-information detection in the English language. The authors used Google Fact Check Tools API to gather public ground truth from various fact-checking websites. The used dataset is collected from WHO, UNICEF, and UN websites from (February 4 to March 10) 2020. The NN classifier provides the best ACC, ERR, and AUC evaluations, with values ranging from 93.75% to 99.68%, 0.32% to 6.25%, and 89.46% to 99.47%, respectively.

Hossain et al. [101] evaluated the performance of COVID-19 misinformation on social media by working on a dataset of 4.8K expert annotated social media posts that contain general misconceptions and their misinformative and informative expressions on Twitter. The dataset contains 86 misconceptions classified into three labels: 465 Misinformative, 164 Informative, and 4161 Irrelevant. The authors performed their analysis on Misinformative and Informative classes and achieved the best results using Avg.Glove Embedding. The authors updated their research and achieved remarkable results compared to the previous one by training the used linear classifiers on three datasets: SNLI [102], MultiNLI [103], and MedNL [104] and applying SBERT (DA) and BiLSTM models on COVID-LIES dataset [105]. Amer and Siddiqui [106] detected COVID-19 Fake News using COVID19FN dataset [107]. The authors applied RF and DT algorithms and got an accuracy score equal to 94.49%.
Gundapu and Mamidi [108] presented a framework for analyzing the credibility of information disseminated on social media about the COVID-19 pandemic in this research. The best strategy for identifying false news is based on an ensemble of three transformer models (BERT, ALBERT, and XLNET). This model was developed and tested as part of the ConstraintAI 2021 shared problem “COVID-19 Fake News Detection in English.” The author compared a group of ML models and a group of DL models with transformer models based on the ensemble method. On the test set, the proposed model received a 0.9855 F1-score and finished fifth out of 160 teams. This infodemic has made it more difficult to access and identify trustworthy information, as well as rumors spread more quickly, putting public health at risk by making effective preventative tactics impossible to implement.

The authors in [109] employed a new system for COVID-19 Fake News Detection on social media by applying ML and modified deep neural network (DNN) methods. In this study, four datasets were used: CoAID [97], The disaster dataset [110], The PolitiFact dataset [111], The gossip cop dataset [111]. The authors achieved accuracy equals 98.57%.

Ng and Carley [112] worked on a study that looks at a corpus of coronavirus-related fact checks gathered from the three main fact-checking organizations (PolitiFact [113], Poynter [114], and Snopes [115]). The authors collected 6731 fact-checked stories, categorized these stories into six clusters, analyzed the temporal trends of story validity and the level of agreement across sites, created a novel pipeline for categorizing tales into more detailed narrative kinds, and applied it to a corpus of COVID-related misleading tweets. The authors utilized the BoW classifier and the BERT model. The BERT model performed well, attaining an average accuracy of 87% in the supervised classification of story validity, accurately detecting an average of 59% and 43% of the tales, respectively.

Al-Ahmad et al. [116], suggested a powerful approach for detecting misleading news. This approach employs kNN–BSSA, kNN–BPSO, kNN–BGA with feature selection, and the standard kNN models to reduce the number of symmetrical features. The authors combined three wrapper feature selection methods: Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Salp Swarm Algorithm (SSA) to get higher accuracy.

Koirala dataset [117] was used to create another six datasets by applying various tokenizers with stemming strategies. Based on the prediction findings, the suggested model showed an accuracy of 75.43% and decreased the number of symmetrical characteristics to 303.

Alouffi et al. [118] offer a hybrid DL model that detects COVID-19 fake news by combining a CNN with a LSTM. The authors used six ML and two DL models. The proposed model for experimental outcomes is tested using three COVID-19 fake news datasets. To validate the results, four matrices are used: accuracy, precision, recall, and F1-measure. Experimental results demonstrate model outperforms the used ML and DL models. As a result, the suggested method produces the best ACC, PRE, REC, and F1 score evaluations, with values ranging from 97.7 to 97.5%, 97.53%, and 97.7%.

Al-Rakhami and Al-Amri [119], designed a novel framework for identifying disinformation on the Twitter network by combining six machine-learning algorithms with ensemble learning, and seven combinations of base models were considered: C4.5 + RF, SVM + RF, C4.5 + kNN, SVM + kNN, SVM + Bayes Net + kNN, C4.5 + Bayes Net + kNN, in the end, all the models combined. The authors used Twitter’s streaming application program interface (API) for tweets to acquire a massive dataset about the COVID-19 outbreak. Human annotators reviewed and labeled the dataset. After that, the author extracted essential COVID-19 traits and used them to develop their system to automatically assess the credibility of tweets. The SVM + RF enabled the ensemble model to produce better results for root relative squared error (RRSE), the area under the receiver operating characteristic (AUROC) curve, accuracy, and Kappa statistic with values ranging from 0.257%, 99.7%, 0.978%, and 97.5%, respectively.

In our previous research [120], to automatically identify and classify misinformation content on social media platforms associated with the COVID-19 pandemic The dataset “COVID-19 Fake News” which includes 21,379 instances of actual and fake news for the COVID-19 pandemic and associated vaccines, was used to train and test these DNNs. In terms of accuracy, the CNN model significantly outperformed the other DNNs in terms of accuracy, scoring 94.2%.

### 4.3 Cancer disease misinformation

Bianchini et al. [121] introduced a comprehensive study on sixteen websites to define various types of falsehoods, evaluate the risk of encountering deceptive information, and examine the variations in expert and layperson assessments. The authors gathered a dataset using if MONITOR and divided it into three classes (correct, incorrect, and undefined).

Bal et al. [122] discussed the spread of misinformation about diseases and health issues via social media, especially cancer, on Twitter. The authors provided a new dataset and used classifiers to distinguish between medically relevant and non-medically relevant tweets. The dataset was collected using the Twitter Streaming API1. The gathered tweets contain specific keywords in
combination with cancer. Also, the authors proposed complex neural techniques to specify the objects/items/techniques suspected of causing cancer or contributing to healing. To distinguish between tweets. The authors obtained an F1 score of 0.7818 for “causes”, 0.8341 for “prevents”, and 0.5106 for “cures” using TF-IDF-weighted PubMed embeddings [123]. The best F1 score of 0.6846 was achieved using the attention-based BiLSRM CRF model. Hou et al. [124], worked on the automatic detection of misinformation on YouTube. The researchers introduced a new collection of 250 prostate cancer movies that had been manually annotated for inaccuracy. To study the use of linguistic, auditory, and user participation characteristics, the researchers built a new dataset of 250 prostate cancer-related videos and manually labeled them. SVM classifiers for misinformative video identification were trained using this dataset. The authors achieved 74% accuracy, 76.5% precision, and 73.2% recall.

Dai et al. [99], created the first complete fake health news datasets. The repository’s extensive properties include news content, news reviews, and social contexts. The authors performed exploratory analysis on the datasets to assess their quality and identify their essential aspects. These datasets perform detection on relatively basic baselines because the goal is not to attain high performance but rather to test the quality of datasets for fake news identification and, offer reference findings for future system evaluations; conduct fake health news detection on the two datasets. using Random guess, Unigram, Unigram + NS, Unigram + Tags, SVM, Random Forest, CNN, Bi-GRU, and SAF, the best result SAF 0.760 Accuracy and getting highest F1-score and AUC using second datasets 0.802, 0.809 respectively.

### 4.4 Influenza misinformation

Jang et al. [125], suggested a plan for extracting training data that is used to represent the hidden features and enhance the performance by filtering and choosing only influenza-related phrases before prediction. The authors utilized the Pearson correlation coefficient, PCC [126] to order the retrieved keywords and the root-mean-square error, RMSE [127] to evaluate each model. The quantity of the collected data was around 761 MB, containing nearly 140,000 words. Using the LSTM model, the best PCC equals 89% and the best RMSE equals 90%. Brainard and Hunter [128], created an agent-based model that simulates independent but related circulating contagious diseases as well as the sharing of health advice (classified as useful or harmful). The scientists used three stages of modeling (1 = no misinformation disseminated, 2 = misinformation causing epidemics to worsen, and 3 = ways to limit the influence of disinformation). Xu and Wojtczak [129], used the GI-SAID [130] database to collect viral protein sequences obtained from avian, swine, and human samples. The authors applied five classifiers (SVM, RF, RUSBoost, XGBoost, and CNN) to a 5-gram model. The results reveal that the PSSM-based model achieved an MCC of approximately 95% and an F1 of around 96. The model achieved MCC of 96%, and F1 of 97%.

### 4.5 Heart disease misinformation

Karajeh et al [131], classified over 7000 heart attack tweets as informative and non-informative. The dataset consists of 11% informative tweets and 89% non-informative tweets. The tweets were classified using DNNs, SVM, DT (J48), and NB with k-cross-validation, k values of 2, 5, and 10. The dataset was divided into 66% training and 34% testing, and all algorithms were tested using the CFS method. DNN had the highest accuracy (95.2%) and the highest F1-score (73.6%) for both informative and non-informational. Based on heart failure fake news.

O’Connor [132], presented a solid case for medical experts to strongly resist overstated treatments, unverified entities, unproven vaccinations, and nutraceuticals, so that may keep a calm yet resolute attitude to the advancement of new chances for the patients they serve.

### 4.6 Cyberbullying (psychological health)

Chatzakou et al. [133] introduced a scalable system for identifying bullying on Twitter and provided a methodology for gathering text, user, and network-based information, investigating the characteristics of bullies and aggressors, and what qualities distinguish them from other users. The authors assessed the technique using a corpus of 1.6 million tweets produced over three months. Two settings are examined to determine the capability of sensing user behavior: four classifications (bully, aggressive, spam, and typical users) and three classifications (bully, aggressive, and normal users). Three ML methods [RF, NV, and DTs (J48)] are employed for detection. When employing 3-class classification, RF got an accuracy of 91.08%, kappa value of 0.5284, RMSE of 0.2117, precision of 0.899%, recall of 0.917%, and AUC of 0.907%.

Balakrishnan et al. [134] provided an automatic cyberbullying detection approach based on Twitter users’ psychological features such as personalities, sentiment, and emotion. In particular, it constructs a detection model based on existing empirical evidence on user personalities and cyberbullying perpetration, and then enhances it with user sentiment and emotion. The dataset contained 5453 tweets from Twitter and was manually annotated by experts. The annotated dataset was obtained from [133] with 9484 annotated tweet IDs. For detection three ML
algorithms are used [RF, NV, and DTs (J48)]. The authors obtain high results with J48 algorithms and with an accuracy of 91.88%, a weighted AUC of 0.97%, F-score 0.92, kappa 0.840, and RMSE 0.178.

Balakrishnan et al. [135] proposed a cyberbullying detection technique based on user personality as indicated by the Big Five [136] and Dark Triad [137] models. The model seeks to identify bullying patterns within Twitter groups based on correlations between personality factors and cyberbullying. The authors used RF for cyberbullying categorization in addition to a baseline approach comprising seven Twitter variables. The best result was found when using baseline + the key personalities from Big Five and Dark Triad with values equal 96% (precision), 95% (recall), and 0.929% (F-measure).

Chavan and Shylaja [138] used supervised ML algorithms like SVM and LR to identify swear words and obnoxiousness in comments on social networks. The datasets used for experiments are collected from the Kaggle website. The best result achieved is logistic regression with the occurrence of pronouns and skip-gram as features of accuracy of 86%, AUC of 86.92%, recall of 0.71%, and precision of 0.769%.

Talpur and O’Sullivan [139] developed a feature-based approach that takes characteristics from a tweet’s content to generate a machine learning classification model for determining whether a tweet is low, non-cyberbullied, medium, or high-level cyberbullied. Enabling feature extraction (age, sexual identity, and personal characteristics) and Twitter API capabilities to integrate pointwise semantic orientation as a new input feature to indicate the degree of cyberbullying in a tweet. To test the effectiveness of each feature and classifier performance, feature selection approaches such as information gain, chi-square, and correlation were applied in different combinations of features. For detection, the author used NB, SVM, KNN, DT, and RF models. Tweets were gathered between November and December of 2019 with 14,495 tweets and annotated to become 11,904 tweets classified. The authors employ the SMOTE technique [140] by oversampling the minority classes to balance the data. RF achieved the highest Kappa with a value of 84%, F-Measure of 92%, and Accuracy of 93% when parameters were set to Base Classifier + SMOTE + Cost Adjusted + Predicted Features + PMI.

Al-Garadi et al. [141] created a functionality model based on tweet features such as network, user, activity, and tweet content. To assess the performance of the four chosen classification models, specifically, NB, LibSVM, RF, and KNN and to evaluate the most important features, three feature selection methods are used: the information gain, chi-square test, and Pearson correlation. The dataset was gathered from Twitter during January and February of 2015 and contains 10,007 tweets. SMOTE was applied to deal with the data set’s unequal class distribution. RF using SMOTE alone showed the best AUC (0.943), F-measure (0.936), Precision (0.941), Recall (0.939).

Kumar and Sachdeva [142] developed a deep neural model for detecting cyberbullying in three different social data modalities (textual, visual, and info-graphic). Caps-Net–ConvNet is an architecture that includes a capsule network (CapsNet) DNN with dynamic routing for predicting textual bullying content and a convolution neural network (ConvNet) for predicting visual bullying content.

Adikara et al. [143] used NB Classifier with Lexicon Based Features to identify cyberbullying comments on Instagram by integrating Bag of Word features and Lexicon Based features. The data used is drawn at random from Bahasa Indonesia Instagram comments (Indonesian).

Zhao et al. [144] proposed a framework for detecting cyberbullying called embeddings enhanced Bag-of-Words model (EBoW), constructed a collection of predefined insulting terms using word embeddings and gave varying weights to obtain bullying features, which are then concatenated using Bag-of-Words and Latent semantic features (LSA) [145] and utilized to generate the final representation before putting it into a linear SVM classifier. The authors used the Twitter dataset [146] to evaluate the proposed model EBoW. Several learning models are used for detection such as BoW Model, Semantic-enhanced BoW Model [147], LSA, LDA [148], and EBoW. The proposed model obtained the best result with a precision equal to 76.8%, recall of 79.4%, and F1-score of 78.0%.

The authors in [149] addressed the issue of automatically identifying abusive language on Arabic social media platforms using CNN, Bi-LSTM, Bi-LSTM with the attention mechanism, and a combined CNN–LSTM. The researchers used the authors’ dataset in [150]. The best recall is achieved by the combined CNN–LSTM network with a value equal to 83.46% while the best accuracy, precision, and F1-score are achieved by the CNN with values equal to 87.84%, 86.10%, and 84.05% respectively.

Haidar et al. [151] created two customized tools for data acquisition. The author used these tools for scraping data from social networks such as Facebook and Twitter. Twitter Scrapper was written in PHP [152], but Facebook Scrapper [153] was written in Python. Both scrapers were linked to a MongoDB server. The tweets were mostly from Lebanon, Syria, the Gulf region, and Egypt. The total size of the tweet database was 4.93 GB. The data was cleaned and preprocessed using WEKA. The TweetToSentiStrengthFeatureVector filter was used as part of the preprocessing techniques, which converts strings to word vectors and normalizes them. After deduplication, the Arabic dataset comprised 35,273 distinct tweets. There were 91,431 tweets in the English data set. The data were subjected to a NB classifier and SVM analysis. In the
bullying instances, 801 tweets were classified correctly, while 1395 tweets were classified as not bullying content. Finally, the SVM model had the highest recall with a value of 94.1%.

Ahmed et al. [154], created a model to detect cyberbullying in Bangla and Romanized Bangla writings. The authors manually chose YouTube videos of several well-known Bangladeshi social media personalities and collected comments using YouTube API version 3.0 [155]. The texts were divided into two databases by the writers. The first dataset had 5000 Bangla texts, whereas the second contained 7000 Romanized Bangla texts. Following that, a third dataset was formed by bringing the first two datasets together, which totaled 12,000 texts. Following that, the authors manually labelled all three datasets as either bullying or not bullying. The authors applied NB, SVM, LR, XGBoost, CNN, LSTM, BLSTM, and GRU algorithms on three datasets. The authors got an accuracy of 84%. CNN outperformed all other algorithms in the dataset, including Bangla texts.

Ali and Syed [156], used LR, SVM, RF, NB, and the Ensemble approach to detect cyberbullying. Four different datasets were used. Because ML methods require numerical input for training, the text was first transformed into a numerical form using a label encoder. The results showed that SVM, Logistic Regression, and Ensemble outperformed the other classifiers, with an average accuracy of 92%.

Süzen and Duman [157], designed a neural network to estimate the type of cyberbullying that young people have been exposed to. It is an XGBoost-based method from the model community learning algorithm XGB-CTD. A survey was created from the cyber security scale that was used as a sample to construct the data collection. To achieve high accuracy, XGB CTD and other methods such as SVM, RF, and gradient boosting DTs (GBDT) were trained by identifying optimal hyperparameters. XGB-CTD has the highest accuracy rating of 91.75%.

Haidar et al. [158], introduced a research to detect cyberbullying in Arabic. The authors used two datasets, the first of which had 4913 records. The second dataset had 34,890 records. The dataset was divided into 80% for training and 20% for testing, and the FFNN model was used on these datasets, yielding a 94.56% accuracy rate on the second dataset.

Sanchez and Kumar [159], applied the NB model to recognize the power of sentiment analysis in detecting bullying on Twitter. The used dataset comprised 1,600,000 training tweets and 359 testing tweets. The authors used Amazon’s Mechanical Turk (crowdsourcing) [160] to categorize unlabeled data and to verify and validate freshly labeled data as part of the outcome evaluation. The NB model achieved a maximum accuracy of 67.3%.

The authors in [161], proposed a CNN cyberbullying detection (CNN-CB) algorithm and used an SVM approach to compare it with the proposed system. The dataset was acquired by the Twitter API and contained 39,000 tweets taken from the Twitter public timeline. Many trials revealed that the CNN-CB algorithm outperforms standard text cyberbullying identification with a 95% accuracy, precision of 93%, and recall of 73%.

The authors in [162] suggested a hybrid architecture combining character-level CNN and word-level LRCN, with LRCN implemented with Tensorflow. The two models were applied to a dataset containing 8815 comments that were binary tagged as 0 (neutral) and 1 (positive) (cyberbullying). Using LRCN, the accuracy rate was 87.22%.

The authors in [163] created a dataset from Instagram and Twitter posts written in Turkish. It contains 900 messages, 450 of which contain cyberbullying content and 450 of which do not. To detect cyberbullying, the authors used ML approaches such as SVMs, DTs (C4.5), NB, and kNNs classifiers. For the dataset utilized, the NB improved accuracy to 84%.

The authors in [164] employed the CNN model and presented a novel pronunciation-based CNN (PCNN), and then they used two datasets. The first dataset was gathered from the Twitter platform, which has only 1313 messages. The second dataset was gathered from the social networking site Formspring.me, included 13,000 messages that were labeled by a web service called Amazon Mechanical Turk. After the datasets were collected, the authors applied CNN and PCNN classifiers, and the final results were an accuracy of 0.968%, a precision of 74%, a recall of 0.453%, and F1-score of 0.562% when using the PCNN classifier.

Malmasi and Zampieri [165], used text categorization algorithms to discriminate between hate speech, profanity, and other texts. To construct a baseline for this job, the authors used typical lexical characteristics and a linear SVM classifier. A character 4-gram model yielded the best results, with an accuracy equal to 78%.

Alakrot et al. [166], gathered and labeled a large dataset of Arabic YouTube comments that included both offensive and inoffensive comments. This dataset was used to train an SVM classifier, and the authors experimented using a variety of N-gram features, word-level features, and pre-processing approaches. The authors then summarized the pre-processing methods and attributes that allow training a more accurate classifier, with accuracy of 90.05%, than earlier studies on Arabic text classifiers.
5 Standard datasets

This section presents the most used benchmarking datasets in the literature for evaluating MLID methods.

CLEF 2018 Consumer Health Search Task [82] focuses on the efficacy of health-related information offered by search engines. The collection includes 5,535,120 web pages retrieved from CommonCrawl. The CLEF eHealth dataset was labeled as Trustworthiness (10,405 positive, 3820 negative), Readability (3102 positive, 12,455 negative), and Useful (1567 positive, 11,488 negative). The Stanford Natural Language Inference (SNLI) dataset [102] contains 570k sentence pairs that have been labeled as neutral, contradiction, or entailment by hand. It has Flickr30k image captions and hypotheses generated by crowd-sourced annotators. It also has annotators who can judge the relationship between sentences that represent the same event.

Each pair is labeled as “entailment,” “neutral,” “contradiction,” or “-” with “-” indicating that an agreement could not be gained. The Multi-Genre Natural Language Inference (MultiNLI) dataset [103] contains 433K sentence pairs. Its dimensions and collection method are very similar to SNLI. MultiNLI presents ten different written and spoken English features (Face-to-face, 9/11, Telephone, Letters, Travel, Oxford University Press, Verbatim, Government, Slate, and Fiction). There are matches between the development and testing datasets that are obtained from the same sources as the training dataset and mismatches between the datasets that do not closely reflect any seen all through training.

The Medical Natural Language Inference (MedNLI) dataset [104] has sentence pairs created from the past medical history section of MIMIC-III clinical notes labeled as Definitely True, Maybe True, and Definitely False by Physicians. The dataset consists of 11,232 training, 1395 developments, and 1422 test samples. COVIDLIES [105] includes 6761 annotated tweet–misconception pairs. The dataset labeled as 670 agree, 343 disagree, and 5748 no stance. COVID19FN [107] is a dataset of misinformation news collected from Poynter and other fact-checking websites during the Infodemic. It comprises roughly 2800 news articles that are labeled as real or fake news.

CoAID (COVID-19 Healthcare Misinformation Dataset) [97] collected from websites and social media fake news related to COVID-19 healthcare. This dataset includes 296,000 user engagements, 4251 news mentions, 3555 tweets, and a ground-truth label.

The disaster dataset [110] contains 7613 tweets labeled as a real disaster (1) or unreal disaster (0). Specifically, it includes 4342 tweets representing real disasters and 3271 representing the opposite. This dataset consists of five features: id, text, keyword, target, and location.

In PolitiFact [111] there are two CSV files. The first one holds 432 tweets related to real news and the second file contains 618 tweets about fake news. The gossip cop dataset [111], like the PolitiFact dataset, consists of two CSV files. Both datasets have five features: id, tweet-id, title, URL, and label. The Koirala dataset [117] was collected using the Webhose.io tool. This dataset consists of 3002 tweets which are divided into three labels: fake news, real news, and partially fake news. COVID-19 Fake News dataset [167] was built from the ground up using a subset of the Zenodo dataset [168]. The summary of the used datasets has been demonstrated in Table 3.

The dataset was collected from the official accounts of global health organizations, as well as their vaccines, from the websites of the WHO [169], International Committee of the Red Cross (ICRC) [170], United Nations (UN) [171], and United Nations Children’s Fund (UNICEF) [172], as well as their official Twitter accounts. After preprocessing the dataset, 1379 new data points were available, with 80% of them committed to training and 20% for usage in the testing phase. The training dataset contains 17,103 tweets. 9179 Tweets were labeled as true, and 7924 tweets were labeled as false. The testing dataset contains 4276 tweets. 2186 Tweets were labeled as true, and 2090 were labeled as false.

ReCOVery [98] is a set of COVID-19 news written in English language, gathered from various websites. An automated approach is used to evaluate the news without involving the experts’ domain. To evaluate the news’
reliability, two famous fact-checking sites were used. These websites are NewsGuard [173], and Media Bias/Fact Check, MBFC [174]. These websites create trustworthiness ratings: the news is labeled as trustworthy if it exceeds certain thresholds, and as unreliable if not.

FakeHealth [99] is a set of reviews written in English by specialists about medical interventions, health, treatments, etc. The dataset published on the HealthNewsReview website [175] which is a web page represents a project supported by the Informed Medical Decisions Foundation [176] that ran from 2005 to 2018. GI-SAID [130] is a global science project and primary source founded in 2008 that gives free access to genomic data of influenza viruses and the COVID-19 coronavirus. It has 60,087 features.

### 6 Standard evaluation metrics

1. **Accuracy (ACC):** is a popular metric for analyzing MLID problems. Based on the input, this metric is used to decide which model is better at defining patterns and relations between samples in a dataset. It decides if a set of measurements are correct on average [177]. This metric can be calculated as follows:

\[
\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)}.
\]

A comparison of the accuracy values reached by MLID approaches have been explained in Table 4.

2. **Error Rate (ERR):** is calculated by dividing the total number of wrong predictions by the total number of samples in the dataset. The ideal error rate is zero and the worst is one [178]. It can be described as follows:

\[
\text{ERR} = 1 - \text{Accuracy}.
\]

3. **Precision:** represents the number of true positives divided by the total number of true positives and false positives [179]. It can be calculated as follows:

\[
\text{Precision} = \frac{TP}{(TP + FP)}.
\]

4. **Recall:** This metric is also known as sensitivity. It represents the ratio of a specific class identified in a correct manner using all the instances of that class as the following formula describes [179]:

\[
\text{Recall} = \frac{TP}{(FN + TP)}.
\]

5. **F1-score:** It’s among the most important metrics for evaluating ML and DL models. It combines precision and recall to calculate the F Score or F Measure by summarizing the model’s performance predictions as described below [180]:

| References | Model        | ACC (%) |
|------------|--------------|---------|
| [83]       | DETERRENT    | 96.00   |
| [92]       | RF           | 90.10   |
| [95]       | RF           | 73.63   |
| [100]      | NN           | 99.68   |
| [106]      | RF           | 94.49   |
| [109]      | Modified LSTM| 98.57   |
| [110]      | Ensemble Model| 98.55  |
| [112]      | BERT         | 87.00   |
| [116]      | kNN-BGA      | 75.40   |
| [120]      | CNN          | 94.20   |
| [131]      | DNN          | 95.20   |
| [133]      | RF           | 91.08   |
| [134]      | J48          | 91.88   |
| [138]      | LR           | 86.00   |
| [139]      | RF           | 93.00   |
| [142]      | ConvNet      | 97.05   |
| [149]      | CNN          | 87.84   |
| [154]      | CNN          | 84.00   |
| [156]      | SVM, LR, Ensemble | 92.00 |
| [157]      | XGB-CTD      | 91.75   |
| [158]      | FFNN         | 94.56   |
| [159]      | NB           | 67.30   |
| [161]      | CNN-CB       | 95.00   |
| [131]      | DNN          | 95.20   |
| [133]      | RF           | 91.08   |
| [134]      | J48          | 91.88   |
| [138]      | LR           | 86.00   |
| [139]      | RF           | 93.00   |
| [142]      | ConvNet      | 97.05   |
| [149]      | CNN          | 87.84   |
| [154]      | CNN          | 84.00   |
| [156]      | SVM, LR, Ensemble | 92.00 |
| [157]      | XGB-CTD      | 91.75   |
| [158]      | FFNN         | 94.56   |
| [159]      | NB           | 67.30   |
| [161]      | CNN-CB       | 95.00   |
| [162]      | LRCN         | 87.22   |
| [163]      | NB           | 84.00   |
| [164]      | PCNN         | 96.00   |
| [165]      | SVM          | 78.00   |
| [166]      | SVM          | 90.50   |
| References | Model | Metric | (%) |
|------------|-------|--------|-----|
| [82]       | RF, SVM, DistilRoBERTa | F1-score | 93.00 |
| [83]       | DETERRENT | Precision | 94.00 |
| [83]       | DETERRENT | Recall | 91.00 |
| [83]       | DETERRENT | F1-score | 93.00 |
| [95]       | RF | Precision | 72.80% |
| [95]       | RF | Recall | 73.60 |
| [95]       | RF | AUC | 89.00 |
| [96]       | CNN | AUC | 97.30 |
| [96]       | CNN | F1-score | 95.30 |
| [100]      | NN | ERR | 32.00 |
| [100]      | NN | AUC | 99.47 |
| [101]      | BERTSCORE (DA) + SBERT (DA) | Precision | 63.30 |
| [101]      | BiLSTM | Recall | 94.20 |
| [101]      | BiLSTM | F1-score | 89.50 |
| [106]      | RF | Precision | 95.00 |
| [106]      | RF | Recall | 95.00 |
| [106]      | RF | F1-score | 95.00 |
| [108]      | Ensemble Model | Precision | 98.55 |
| [108]      | Ensemble Model | Recall | 98.55 |
| [108]      | Ensemble Model | F1-score | 98.55 |
| [109]      | Modified LSTM | Precision | 98.55 |
| [109]      | Modified LSTM | Recall | 98.60 |
| [109]      | Modified LSTM | F1-score | 98.50 |
| [116]      | kNN–BGA | Precision | 66.22 |
| [116]      | kNN | Recall | 69.57 |
| [116]      | kNN–BSSA | F1-score | 61.96 |
| [120]      | CNN | Precision | 93.60 |
| [120]      | CNN | Recall | 93.90 |
| [120]      | CNN | F1-score | 93.70 |
| [120]      | CNN | Specificity | 93.90 |
| [120]      | CNN | Error rate | 5.80 |
| [120]      | CNN | Miss rate | 5.50 |
| [120]      | CNN | FPR | 6.00 |
| [131]      | DNN | F1-score | 73.60 |
| [134]      | J48 | AUC | 97.00 |
| [134]      | J48 | F-score | 92.00 |
| [134]      | J48 | Kappa | 84.00 |
| [134]      | J48 | RMSE | 17.00 |
| [133]      | RF | Kappa | 59.00 |
| [133]      | RF | RMSE | 14.00 |
| [133]      | RF | AUC | 81.00 |
| [133]      | RF | Precision | 90.00 |
| [133]      | RF | Recall | 91.00 |
| [135]      | RF + Big Five and Dark Triad | Precision | 96.00 |
| [135]      | RF + Big Five and Dark Triad | Recall | 95.00 |
| [135]      | RF + Big Five and Dark Triad | F1-score | 92.00 |
| [138]      | LR | AUC | 86.92 |
| [138]      | LR | Recall | 71.00 |
\[
F1-score = 2 \cdot \frac{(\text{Precision} \cdot \text{Recall})}{(\text{Precision} + \text{Recall})}.
\]  
(5)

**Area under curve (AUC):** is another name for c-statistics. It calculates the total two-dimensional area beneath the entire ROC curve. It is computed by adding Concordance Percentage and 0.5 times of Tied Percentage [181].

\[
AUC = \int_0^1 TPR(FPR^{-1}(x))dx.
\]  
(6)

**False-positive rate (FPR):** is a reliable and valid instrument used to identify the ratio of incorrectly classified negative samples to the overall number of negative samples [52].

\[
FPR = \frac{FP}{(FP + TN)}.
\]  
(7)

**Specificity:** is a metric used to assess the accuracy of negative instances [52].

\[
\text{Specificity} = \frac{TN}{(TN + FN)}.
\]  
(8)

**Miss Rate:** It is the percentage of positive samples that are misclassified [100].

\[
\frac{FN}{(FN + PN)}.
\]  
(9)

**Root Mean Square Error (RMSE):** is the residuals’ standard deviation (prediction errors).
indicates how narrowly the data is centered near the line of best fit \([182, 183]\).

\[
\sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}.
\]  

(10)

A comparison of the most popular performance metrics obtained using MLID approaches is shown in Table 5.

## 7 Data limitations and challenges

All research has challenges and/or limitations associated to the dataset. So, this chapter considers the most common of them.

### 7.1 Lack of a particular form

Detecting misleading information is difficult manually and automatically because it comes in many different forms in the online environment. For example, clickbait headlines are used to entice users to open potentially biased articles to profit from views. When the title reflects an important subject for the reader while the content is irrelevant and completely contrary to the reader’s expectations. Fake news most often exists on malicious websites that spread it on purpose. They hire malicious users to do this job or use social media bots. As a result, distracted users who fail to verify the article’s source before will help them by sharing it. On the other hand, Fake news can also exist on trustworthy websites. This might happen by accident or in the rush to publish shocking news without checking the source first.

### 7.2 Reliable data

In data collection, fact-checking is a critical aspect to produce a reliable dataset. The researcher in [184] introduced fact-checking techniques used in modeling and identifying fake news. Fact-checking techniques depend on human professionals to evaluate data integrity. This methodology is applied by websites such as FactCheck.org, and Snopes. The main problem with these techniques is time and resources consuming.

### 7.3 Untrusted sources

Despite the advent of numerous professional fact-checking sites and tools, there are still flaws and issues. Because of the absence of consistency and coherence among different fact-checking sites, the sites may not be as dependable or credible as intended. For example, Fact Checker [115] and Politifact rarely verify the same facts, and when they do, they frequently disagree [185]. This may further perplex people and possibly compel them not to reevaluate their preexisting impressions of the truthfulness of a tale. In fact, fact-checking websites are not available to the general public as originally anticipated. Escaping our complex misinformation networks may be more difficult than we believe [186].

### 7.4 Unbalanced data

Data collection is an expensive and difficult process, so gathering unbalanced data consider a big problem for researchers because, after this effort and in most cases, they have to drop part of this collected data. To apply most of the classification algorithms, we are required to balance uneven data by equalizing the size of label classes, which leads in dealing with significantly less data than we may wish or dealing with data redundancy. As a result, this will affect our ability to achieve a large enough and representative sample of examples from the minority class negatively. Karajeh et al. [131], faced this problem by using an unbalanced dataset that contains a higher percentage of non-informational tweets than informational tweets.

### 7.5 Limited data sources

Misleading information can be found anywhere, but the researchers tend to use Twitter data because the tweets are considered open source and their use are legal, while other platforms such as Facebook and LinkedIn are considered private and illegal content. This problem has faced a lot of researchers such as Karajeh et al. [131] since Twitter was the only platform used in this work.

## 8 Conclusion

This paper proposed a detailed study of the most recent works developed for MLID in health fields based on ML and DL techniques. It also introduced a detailed discussion of the main phases of the generic adopted approach for MLID. Many critical topics that affect MLID techniques have been discussed, including text preprocessing, feature engineering, data splitting, and model building. Moreover, the benchmarking datasets and the most used metrics to evaluate the performance of MLID algorithms are explored. A deep investigation of the limitations and drawbacks of the current technological advancements in various research directions is provided to help the researchers to choose the most suitable for this emerging task of MLID.
Detecting misleading information in the text is a difficult task with numerous applications. More DL transformers should be used to get better results. Further, expanding the annotated datasets by including data from different domains and languages are fertile research areas for future works paths.

### Appendix

See Table 6.

**Table 6** The complete list of abbreviations used in this survey

| Abbreviations | Definition | Abbreviations | Definition |
|---------------|------------|---------------|------------|
| MLID          | Misleading information detection | ERF          | Ensemble random forest |
| DT            | Decision tree | XGBoost       | Extreme gradient boosting |
| kNN           | k-Nearest neighbor | ACC          | Accuracy |
| ML            | Machine learning | SVM          | Support vector machine |
| PAC           | Passive aggressive classifier | Bi-LSTM     | Bidirectional long short-term memory |
| MLP           | Multi-Layer perceptron | BERT        | Bidirectional encoder representations from transformers |
| ALBERT        | A Lite BERT | NB            | Naive Bayes |
| CNN           | Convolutional neural network | LSTM        | Long short-term memory |
| LSVM          | Linear support vector machines | ERR         | Error rate |
| NLI           | Natural language inference | SBERT       | Sentence-BERT |
| PSO           | Particle swarm optimization | GA          | Genetic algorithm |
| BNB           | Bernoulli Naïve Bayes | AUC         | Area under The curve |
| SSA           | Salp swarm algorithm | BSSA        | Binary salp swarm algorithm |
| RRSE          | Root relative squared error | PRE         | Precision |
| EBoW          | Embeddings-enhanced Bag-of-Words | REC         | REC |
| BPSO          | Binary particle swarm optimization | BGA        | Binary-coded genetic algorithm |
| NN            | Neural network | CAM          | Complementary and alternative medicine |
| PMI-SO        | Pointwise semantic orientation of each word and phrases | SMOTE       | Synthetic Minority Over-sampling Technique |
| LDA           | Latent Dirichlet allocation | LSA         | Latent semantic features |
| LibSVM        | Library for support vector machines | CapsNet     | Capsule network deep neural network |
Author contributions The authors declare that each author has contributed to the realization of this work.

Funding This research received no external funding.

Data availability Data utilized in this article must be approved by the corresponding author.

Code availability Not applicable.

Consent for publication Not applicable.

Declarations

Conflict of interest The authors declare there were no conflicts of interest.

Ethical approval Not applicable.

Informed consent Not applicable.

References

1. WHO: When and How to Use Masks. Word Health Organization. https://tinyurl.com/56763k99. Accessed June 2022
2. Coronavirus Disease (COVID-19). https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/index.html. Accessed 15 Sep 2021
3. Ngan, M.L., Grother, P.J., Hanaoka, K.K.: Ongoing Face Recognition Vendor Test (FRVT) Part 6A: Face Recognition Accuracy with Masks Using Pre-COVID-19 Algorithms; NIST Interagency/Internal Report (NISTIR). National Institute of Standards and Technology, Gaithersburg (2020)
4. Ngan, M.L., Grother, P.J., Hanaoka, K.K.: Ongoing Face Recognition Vendor Test (FRVT) Part 6B: Face Recognition Accuracy with Face Masks Using Post-COVID-19 Algorithms; NIST Interagency/Internal Report (NISTIR). National Institute of Standards and Technology, Gaithersburg (2020)
5. Gottlieb, S.: Health information on Internet is often unreliable. Br. Med. J. 321(7254), 136 (2000)
6. Granik, M., Mesyura, V.: Fake news detection using Naive Bayes classifier. In: 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), May 2017, pp. 900–903. IEEE (2017)
7. Kumar, R., Verma, R.: KDD techniques: a survey. Int. J. Electron. Comput. Sci. Eng. 1(4), 2042–2047 (2008)
8. Shastri, A., Deshpande, M.: A review of big data and its applications in healthcare and public sector. Big Data Anal. Healthc. 66, 55–66 (2019). https://doi.org/10.1007/978-3-030-31672-3_4
9. Mukherjee, S., Shaw, R., Haldar, N., Changdar, S.: A survey of data mining applications and techniques. Int. J. Comput. Sci. Inf. Technol. 6(5), 4663–4666 (2015)
10. Beebe, M.: Types of Misinformation and Disinformation. Shawneesu.libguides.com, 7 October 2021. https://tinyurl.com/52vmry95. Accessed 20 Aug 2022
11. Must Known Techniques for text preprocessing in NLP. Analytics Vidhya, 14 June 2021. https://tinyurl.com/5ccaya5t. Accessed 20 Aug 2022
12. Kannan, S., Gurusamy, V., Vijayarani, S., I lamathi, J., Nithya, M., Kannan, S., Gurusamy, V.: Preprocessing techniques for text mining. Int. J. Comput. Sci. Commun. Netw. 5(1), 7–16 (2014)
13. Jurafsky, D., Martin, H.J.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, Upper Saddle River (2000)
14. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: ICML, vol. 97, pp. 412–420 (1997)
15. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. Knowl. Data Eng. 17(4), 491–502 (2005)
16. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. 34(1), 1–47 (2002)
17. Tang, J., Alelyani, S., Liu, H.: Feature selection for classification: a review. In: Data Classification: Algorithms and Applications, p. 37. CRC Press, Boca Raton (2014)
18. Brown, G.: A new perspective for information theoretic feature selection. In: Artificial Intelligence and Statistics, pp. 49–56 (2009)
19. Song, Q., Ni, J., Wang, G.: A fast clustering-based feature subset selection algorithm for high dimensional data. IEEE Trans. Knowl. Data Eng. 25(1), 1–14 (2013)
20. Yang, J., Liu, Y., Zhu, X., Liu, Z., Zhang, X.: A new feature selection based on comprehensive measurement both in inter-category and intra-category for text categorization. Inf. Process. Manag. 48(4), 741–754 (2012)
21. Dy, J.G., BRODLEY, C.E.: Feature selection for unsupervised learning. J. Mach. Learn. Res. 5, 845–889 (2004)
22. Pudil, P., Novovicová, J., Kitlner, J.: Floating search methods in feature selection. Pattern Recognit. Lett. 15(11), 1119–1125 (1994)
23. Das, S.: Filters, wrappers and a boosting-based hybrid for feature selection. In: ICML, 2001, vol. 1, pp. 74–81 (2001)
24. Xing, E.P., Jordan, M.I., Karp, R.M., et al.: Feature selection for high-dimensional genomic microarray data. In: ICML, 2001, vol. 1, pp. 601–608 (2001)
25. Dash, M., Liu, H.: Feature selection for classification. Intell. Data Anal. 1(1–4), 131–156 (1997)
26. Molina, L.C., Belanche, L., Nebot, A.: Feature selection algorithms: a survey and experimental evaluation. In: Proceedings of 2002 IEEE International Conference on Data Mining, 2002. ICDM 2002, pp. 306–313. IEEE (2002)
27. Deng, X., Li, Y., Weng, J., et al.: Feature selection for text classification: a review. Multimed. Tools Appl. 78, 3797–3816 (2019). https://doi.org/10.1007/s11042-018-6083-5
28. Wan, C., Wang, Y., Liu, Y., Ji, J., Feng, G.: Composite feature extraction and selection for text classification. IEEE Access 7, 35208–35219 (2019)
29. Riseman, E.M., Hanson, A.R.: A contextual postprocessing system for error correction using binary n-grams. IEEE Trans. Comput. C23(5), 480–493 (1974)
30. Bekkerman, R., Allan, J.: Using Bigrams in Text Categorization, pp. 1120–1121. Center of Intelligent Information Retrieval, Amherst (2004)
31. Tesař, R., Strnad, V., Jezeck, K., Poesio, M.: Extending the single words based document model: a comparison of bigrams and 2-itemsets. In: Proceedings of ACM Symposium on Document Engineering, pp. 138–146 (2006)
32. Dim, B., Hakan, A.: A novel framework for termset selection and weighting in binary text classification. Eng. Appl. Artif. Intell. 35(2), 38–53 (2014)
33. Dim, B., Hakan, A.: Termset weighting by adapting term weighting schemes to utilize cardinality statistics for binary text categorization. Appl. Inn. 47(4), 456–472 (2017)
34. Kuhn, M., Johnson, K.: Feature Engineering and Selection: A Practical Approach for Predictive Models. CRC Press, Boca Raton (2019)
35. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint (2018). arXiv:1810.04805
36. Wang, B., Kuo, C.C.J.: SBERT-WK: a sentence embedding method by dissecting BERT-based word models. IEEE/ACM Trans. Audio Speech Lang. Process. 28, 2146–2157 (2020)
37. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: a lite BERT for self-supervised learning of language representations. arXiv preprint (2019). arXiv:1909.11694
38. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. 9(8), 1735–1780 (1997)
39. Liu, G., Guo, J.: Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing 337, 325–338 (2019)
40. Shekhawat, S.S., Sharma, H., Kumar, S., Nayyar, A., Qureshi, B.: bSSA: binary Salp swarm algorithm with hybrid data transformation for feature selection. IEEE Access 9, 14867–14882 (2021)
41. Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M.: Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems. Adv. Eng. Softw. 114, 163–191 (2017)
42. Albawi, S., Mohammed, T.A., Al-Zawi, S.: Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET), August 2017, pp. 1–6. IEEE (2017)
43. Cortes, C., Vapnik, V.: Support vector machine. Mach. Learn. 20(3), 273–297 (1995)
44. Hall, L.O., Chawla, N., Bowyer, K.W.: Decision tree learning on very large data sets. In: SMC’98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218), October 1998, vol. 3, pp. 2579–2584. IEEE (1998)
45. Svetnik, V., Liaw, A., Tong, C., Culberson, J.C., Sheridan, R.P., Feuston, B.P.: Random forest: a classification and regression tool for compound classification and QSAR modeling. J. Chem. Inf. Comput. Sci. 43(6), 1947–1958 (2003)
46. Bayes, T.: Naive Bayes Classifier. Article Sources and Contributors, pp. 1–9 (1968)
47. Jacquez, G.M.: A k nearest neighbour test for space–time interaction. Stat. Med. 15(18), 1935–1949 (1996)
48. Agarwal, A.K., Wadhwa, S., Chandra, S.: XGBoost: a scalable tree boosting system. J. Assoc. Physicians India 42(8), 665 (1994)
49. Jenkins, W.M.: Towards structural optimization via the genetic algorithm. Comput. Struct. 40(5), 1321–1327 (1991)
50. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
51. BERT Explained: State of the art language model for NLP. Medium; Towards Data Science, 10 November 2018. https://towardsdatascience.com/bert-explained-state-of-the-art-langauge-model-for-nlp-188b21a9b6270. Accessed 24 May 2022
52. Alzu’bi, A., Albalsas, F., Al-Hadhrami, T., Younis, L.B., Bashyreh, A.: Masked face recognition using deep learning: a review. Electronics 10(21), 2666 (2021)
53. Guide to Text Classification Using TextCNN, 18 July 2021. https://analyticsindiamag.com/guide-to-text-classification-using-textcnn/. Accessed 20 Aug 2022
54. Wikipedia Contributors: Convolutional Neural Network. Wikipedia; Wikimedia Foundation, 27 February 2019. https://tinyurl.com/2p887vdj. Accessed 20 Aug 2022
55. Dupond, S.: A thorough review on the current advance of neural network structures. Annu. Rev. Control 14, 200–230 (2019)
56. Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A., Arshad, H.: State-of-the-art in artificial neural network applications: a survey. Heliyon 4(11), e00938 (2018)
57. Tealab, A.: Time series forecasting using artificial neural networks methodologies: a systematic review. Future Comput. Inform. J. 3(2), 334–340 (2018)
58. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. IEEE Trans. Pattern Anal. Mach. Intell. 31(5), 855–868 (2008)
59. Zia, T., Zahid, U.: Long short-term memory recurrent neural network architectures for Urdu acoustic modeling. Int. J. Speech Technol. 22(1), 21–30 (2019)
60. Li, X., Wu, X.: Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), April 2015, pp. 4520–4524. IEEE (2015)
61. Miljanovic, M.: Comparative analysis of recurrent and finite impulse response neural networks in time series prediction. Indian J. Comput. Sci. Eng. 3(1), 180–191 (2012)
62. Wikipedia Contributors: Recurrent Neural Network. Wikipedia; Wikimedia Foundation, 3 December 2018. https://tinyurl.com/37c6m6a. Accessed 20 Aug 2022
63. LSTM for Text Classification Beginners Guide to Text Classification, Analytics Vidhya, 14 June 2021. https://tinyurl.com/yc2376cf. Accessed 20 Aug 2022
64. Wikipedia Contributors: Long Short-Term Memory. Wikipedia; Wikimedia Foundation, 22 November 2018. https://en.wikipedia.org/wiki/Long_short-term_memory. Accessed 20 Aug 2022
65. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: encoder–decoder approaches. arXiv preprint (2014). arXiv:1409.1259
66. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with LSTM. Neural Comput. 12(10), 2451–2471 (2000)
67. Britz, D.: Recurrent neural network tutorial, part 4 implementing a GRU/LSTM RNN with Python and Theano. 2015. http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-mm-with-python-and-theano. Accessed 20 Aug 2022
68. Ravanelli, M., Brakel, P., Omonogpo, M., Bengio, Y.: Light gated recurrent units for speech recognition. IEEE Trans. Emerg. Top. Comput. Intell. 2(2), 92–102 (2018)
69. Su, Y., Kuo, C.C.J.: On extended long short-term memory and dependent bidirectional recurrent neural network. Neurocomputing 356, 151–161 (2019)
70. Chang, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint (2014). arXiv:1412.3555
71. Gruber, N., Jockisch, A.: Are GRU cells more specific and LSTM cells more sensitive in motive classification of text? Front. Artif. Intell. 3, 40 (2020)
72. Wikipedia Contributors,: Decision Tree. Wikipedia; Wikimedia Foundation, 28 September 2019. https://en.wikipedia.org/wiki/Decision_tree. Accessed 20 Aug 2022
73. Islam, M.Z., Liu, J., Li, J., Liu, L., Kang, W.: A semantics aware random forest for text classification. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, November 2019, pp. 1061–1070 (2019)
112. Ng, L.H.X., Carley, K.M.: The coronavirus is a bioweapon: classifying coronavirus stories on fact-checking sites. Comput. Math. Organ. Theory 27(2), 179–194 (2021)

113. Fact-checking U.S. politics PolitiFact. PolitiFact (2019). https://tinyurl.com/5886jhrk. Accessed 20 Aug 2022

114. IFCN COVID-19 Misinformation – Poynter- (2022). https://www.poynter.org/ifcn-covid-19-misinformation/. Accessed 20 Aug 2022

115. Fact Check. Snopes.com (2018). https://www.snopes.com/fact-check/. Accessed 20 Aug 2022

116. Al-Ahmad, B., Al-Zoubi, A.M., Abu Khurma, R., Aljarah, I.: An evolutionary fake news detection method for COVID-19 pandemic information. Symmetry 13(6), 1091 (2021)

117. Koirala, A.: COVID-19 Fake News Dataset. Mendeley Data, vol. V1 (2021). https://doi.org/10.17632/zwfdmp5vyg.1

118. Alouffi, B., Alharbi, A., Sahal, R., Saleh, H.: An optimized hybrid deep learning model to detect COVID-19 misleading information. Comput. Intell. Neurosci. 2021(1), 1–15 (2021)

119. Al-Rakhami, M.S., Al-Amri, A.M.: Lies kill, facts save: towards detecting COVID-19 misinformation in Twitter. IEEE Access 8, 155961–155970 (2020)

120. Tashtoush, Y., Alrababah, B., Darwish, O., Maabreh, M., Alsaeedi, N.: A deep learning framework for detection of COVID-19 fake news on social media platforms. Data 7(5), 65 (2022)

121. Bianchini, C., Truccolo, L., Bidoli, E., C.I.Q.A. Group, Mazzocut, M.: Avoiding misleading information: a study of complementary medicine online information for cancer patients. Libr. Inf. Sci. Res. 41(1), 67–77 (2019)

122. Bal, R., Sinha, S., Dutta, S., Joshi, R., Ghosh, S., Dutt, R.: Analysing the extent of misinformation in cancer related tweets. In: Proceedings of the International AAAI Conference on Web and Social Media, May 2020, vol. 14, pp. 924–928 (2020)

123. Chen, Q., Peng, Y., Lu, Z.: BioSentVec: creating sentence embeddings for biomedical texts. In: 2019 IEEE International Conference on Healthcare Informatics (ICHI), June 2019, pp. 1–5. IEEE (2019)

124. Hou, R., Perez-Rosas, V., Loeb, S., Mihaleca, R.: Towards automatic detection of misinformation in online medical videos. In: 2019 International Conference on Multimodal Interaction, October 2019, pp. 235–243 (2019)

125. Jang, B., Kim, I., Kim, J.W.: Effective training data extraction method to improve influenza outbreak prediction from online news articles: deep learning model study. JMIR Med. Inform. 4(5), e23305 (2021)

126. Benesty, J., Chen, J., Huang, Y., Cohen, I.: Pearson correlation coefficient. In: Noise Reduction in Speech Processing, pp. 1–5. Springer, New York (2009)

127. Chai, T., Draxler, R.R.: Root mean square error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE in the literature. Geosci. Model Dev. 7(2), 1247–1250 (2014). https://doi.org/10.5194/gmd-7-1247-2014

128. Brainard, J., Hunter, P.R.: Misinformation making a disease outbreak worse: outcomes compared for influenza, monkeypox, and norovirus. Simulation 96(4), 365–374 (2020)

129. Xu, Y., Wojtczak, D.: Predicting influenza A viral host using PSSM and word embeddings. In: 2021 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), October 2021, pp. 1–10. IEEE (2021)

130. GISAID-Initiative: Gisaid.org. https://www.gisaid.org. Accessed 13 Dec 2020

131. Karajeh, O., Darweesh, D., Darwish, O., Abu-El-Rub, N., Alsinglawi, B., Alsaeedi, N.: A classifier to detect informational vs. non-informational heart attack tweets. Future Internet 13(1), 19 (2021)

132. O’Connor, C.M.: Heart failure fake news: how do we distinguish the truth? JACC Heart Fail. 7(4), 363–363 (2019)

133. Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., Vakali, A.: Mean birds: detecting aggression and bullying on Twitter. In: Proceedings of the 2017 ACM on Web Science Conference, June 2017, pp. 13–22 (2017)

134. Balakrishnan, V., Khan, S., Arabnia, H.R.: Improving cyberbullying detection using Twitter users’ psychological features and machine learning. Comput. Secur. 90, 101710 (2020)

135. Balakrishnan, V., Khan, S., Fernandez, T., Arabnia, H.R.: Cyberbullying detection on Twitter using Big Five and Dark Triad features. Personal. Individ. Differ. 141, 252–257 (2019)

136. McCrae, R.R., John, O.P.: An introduction to the five-factor model and its applications. J. Personal. 60(2), 175–215 (1992)

137. Paulhus, D.L., Williams, K.M.: The dark triad of personality: Narcissism, Machiavellianism, and Psychopathy. J. Res. Personal. 36(6), 556–563 (2002)

138. Chavan, V.S., Shylaja, S.S.: Machine learning approach for detection of cyber-aggressive comments by peers on social media network. In: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), August 2015, pp. 2354–2358. IEEE (2015)

139. Talpur, B.A., O’Sullivan, D.: Multi-class imbalance in text classification: a feature engineering approach to detect cyberbullying in Twitter. Informatics 7(4), 52 (2020)

140. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357 (2002)

141. Al-Garadzi, M.A., Varathan, K.D., Ravana, S.D.: Cybercrime detection in online communications: the experimental case of cyberbullying detection in the Twitter network. Comput. Hum. Behav. 63, 433–443 (2016)

142. Kumar, A., Sachdeva, N.: Multimodal cyberbullying detection using capsule network with dynamic routing and deep convolutional neural network. Multimed. Syst. (2021). https://doi.org/10.1007/s00530-020-00747-5

143. Adikara, P.P., Adinugroho, S., Insani, S.: Detection of cyber harassment (cyberbullying) on Instagram using Naive Bayes classifier with bag of words and lexicon based features. In: Proceedings of the 5th International Conference on Sustainable Information Engineering and Technology, November 2020, pp. 64–68 (2020)

144. Zhao, R., Zhou, A., Mao, K.: Automatic detection of cyberbullying on social networks based on bullying features. In: Proceedings of the 17th International Conference on Distributed Computing and Networking, January 2016, pp. 1–6 (2016)

145. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. Discourse Process. 25(2–3), 259–284 (1998)

146. Fighting bullying with machine learning. Research.cs.wisc.edu (n.d.). https://research.cs.wisc.edu/bullying/data.html. Accessed 24 May 2022

147. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. 3(Jan), 993–1022 (2003)

148. Nahar, V., Li, X., Pang, C.: An effective approach for cyberbullying detection. Commun. Inf. Sci. Manag. Eng. 3(5), 238 (2013)

149. Mohaouache, H., Mourhir, A., Nikolov, N.S.: Detecting offensive language on Arabic social media using deep learning. In: 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), October 2019, pp. 466–471. IEEE (2019)

150. Alakrot, A., Murray, L., Nikolov, N.S.: Dataset construction for the detection of anti-social behaviour in online communication in Arabic. Procedia Comput. Sci. 142, 174–181 (2018)
151. Haidar, B., Chamoun, M., Serhrouchni, A.: A multilingual system for cyberbullying detection: Arabic content detection using machine learning. Adv. Sci. Technol. Eng. Syst. J. 2(6), 275–284 (2017)

152. Twitter Scraper Apify. Apify (n.d.). https://apify.com/vedrmuta/twitter-scraper#features. Accessed 24 May 2022

153. Facebook Scraper Apify. Apify (n.d.). https://apify.com/pocesar/facebook-pages-scraper#why-scrrape-facebook. Accessed 24 May 2022

154. Ahmed, M. T., Rahman, M., Nur, S., Islam, A., Das, D.: Deployment of machine learning and deep learning algorithms in detecting cyberbullying in Bangla and Romanized Bangla text: a comparative study. In: 2021 International Conference on Advances in Electrical, Computing. Communication and Sustainable Technologies (ICAECT), February 2021, pp. 1–10. IEEE (2021)

155. YouTube API. Wikipedia, 24 May 2022. https://tinyurl.com/5n7c5aak. Accessed 20 Aug 2022

156. Ali, A., Syed, A.M.: Cyberbullying detection using machine learning. Pak. J. Eng. Technol. 3(2), 45–50 (2020)

157. Sizzen, A.A., Duman, B.: Detection of types cyber-bullying using fuzzy C-means clustering and XGBoost Ensemble algorithm (2021)

158. Haidar, B., Chamoun, M., Serhrouchni, A.: Arabic cyberbullying detection: using deep learning. In: 2018 7th International Conference on Computer and Communication Engineering (ICCEC), September 2018, pp. 284–289. IEEE (2018)

159. Sanchez, H., Kumar, S.: Twitter bullying detection. Ser. NSDI 12(2011), 15 (2011)

160. Amazon Mechanical Turk: Mturk.com. Amazon Mechanical Turk (2018). https://www.mturk.com/. Accessed 20 Aug 2022

161. Al-Ajlan, M.A., Ykhlef, M.: Deep learning algorithm for cyberbullying detection. Int. J. Adv. Comput. Sci. Appl. 9(9), 199–205 (2018)

162. Bu, S.J., Cho, S.B.: A hybrid deep learning system of CNN and LRCN to detect cyberbullying from SNS comments. In: International Conference on Hybrid Artificial Intelligence Systems, June 2018, pp. 561–572. Springer, Cham (2018)

163. Özsel, S.A., Saraç, E., Aksemir, S., Aksu, H.: Detection of cyberbullying on social media messages in Turkish. In: 2017 International Conference on Computer Science and Engineering (UBMK), October 2017, pp. 366–370. IEEE (2017)

164. Zhang, X., Zhang, J., Vishwamitra, N., Whittaker, E., Mazer, J.P., Kowalski, R., Dillon, E.: Cyberbullying detection with a pronunciation based convolutional neural network. In: 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), December 2016, pp. 740–745. IEEE (2016)

165. Malmasi, S., Zampieri, M.: Detecting hate speech in social media. arXiv preprint (2017). arXiv:1712.06427

166. Alakrot, A., Murray, L., Nikolov, N.S.: Towards accurate legal jurisdictional claims in published maps and institutional affiliations. Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Omar Darwish received the Ph.D. degree in computer science from Western Michigan University, USA and the M.S. degree from the Jordan University of Science and Technology, Jordan. He has worked as a Visiting Assistant Professor with the West Virginia University Institute of Technology, a Software Engineer with MathWorks, and a Programmer with the Nuqul Group. He is currently an Assistant Professor with the department of Information Security and Applied Computing, Eastern Michigan University. His research interests include Cybersecurity, IoT and Machine Learning.
Yahya Tashtoush is an Associate Professor at the College of Computer and Information Technology, Jordan University of Science and Technology (JUST), Irbid, Jordan. He received his B.Sc. and M.Sc. degrees in Electrical Engineering from JUST in 1995 and 1999, respectively. He received his Ph.D. degree in Computer Engineering from the University of Alabama in Huntsville and the University of Alabama at Birmingham, AL, USA in 2006 (Joint Program). His current research interests are Software Engineering, Information Retrieval, Machine/Deep Learning, Fuzzy Systems and Wireless Networks.

Amjad Bashayreh received her B.S. degree in Software Engineering from Jordan University of Science and Technology (JUST), Irbid, Jordan in 2020. Currently, she is a graduate student in the department of Computer Science, JUST. Her research interests include Cloud Computing, Network Security, Computer Vision, Deep Learning, Machine Learning, Image Processing, Artificial Intelligence.

Alaa M. Alomar is a graduate student in the department of Computer Science, the College of Computer and Information Technology, Jordan University of Science and Technology (JUST), Irbid, Jordan. She received her B.Sc. degree in Computer Science from Yarmouk University in 2020. Her current research interests are Machine Learning, Deep Learning, AI, Natural language processing, Cloud computing, and Edge computing.

Shahed S. Alkhazaleh is a graduate student in the department of Computer Science, the College of Computer and Information Technology, Jordan University of Science and Technology (JUST), Irbid, Jordan. She received her B.Sc. degree in Computer Science from Yarmouk University in 2019. Her current research interests are Machine Learning, Deep Learning, AI, Natural language processing, Cloud computing, and Edge computing.

Dirar Darweesh received the M.S. degree in computer science from Jordan University of Science and Technology, Jordan. He worked as a lecturer with Jordan University of Science and Technology. His research interests include Cybersecurity, IoT and Machine Learning.