BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 24/25 pp942–963
DOI: 10.26599/BDMA.2024.9020015
Volume 7, Number 3, September 2024

Analysis and Classification of Fake News Using Sequential Pattern Mining

M. Zohaib Nawaz, M. Saqib Nawaz, Philippe Fournier-Viger*, and Yulin He

Abstract: Disinformation, often known as fake news, is a major issue that has received a lot of attention lately. Many researchers have proposed effective means of detecting and addressing it. Current machine and deep learning based methodologies for classification/detection of fake news are content-based, network (propagation) based, or multimodal methods that combine both textual and visual information. We introduce here a framework, called FNACSPM, based on sequential pattern mining (SPM), for fake news analysis and classification. In this framework, six publicly available datasets, containing a diverse range of fake and real news, and their combination, are first transformed into a proper format. Then, algorithms for SPM are applied to the transformed datasets to extract frequent patterns (and rules) of words, phrases, or linguistic features. The obtained patterns capture distinctive characteristics associated with fake or real news content, providing valuable insights into the underlying structures and commonalities of misinformation. Subsequently, the discovered frequent patterns are used as features for fake news classification. This framework is evaluated with eight classifiers, and their performance is assessed with various metrics. Extensive experiments were performed and obtained results show that FNACSPM outperformed other state-of-the-art approaches for fake news classification, and that it expedites the classification task with high accuracy.

Key words: disinformation; fake news; sequential pattern mining (SPM); frequent patterns; classification

1 Introduction

People can now quickly receive news and information through various online sources. However, easy and cheap access to information has made disinformation (fake news)[1–3] not only widespread but also a great threat to our society and everyday life. Unlike conventional news outlets like television and newspapers, users can now easily use online social networking (OSN) platforms and messaging services to create, publish content, and spread it quickly[4–6]. According to the report published in November 2023, at least half of American adults received most of their news from OSN platforms as compared to television, radio, and paper publications※. Analysis and identification of fake news are critical for many reasons, for example, (1) individuals or organizations create and spread fake news for personal, financial, or political gains. (2) Fake news can mislead the general public and make them adopt false beliefs. (3) Fake news has the power to change the way the public

• M. Zohaib Nawaz, M. Saqib Nawaz, and Philippe Fournier-Viger are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060. E-mail: nawazmuhammadzohaib2022@email.szu.edu.cn; msaqibnawaz@szu.edu.cn; philfv@szu.edu.cn.
• Yulin He is with the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen 518107, China. E-mail: yulinhe@gml.ac.cn.
• M. Zohaib Nawaz is also with the Department of Computer Science, Faculty of Computing and Information Technology, Univesity of Sargodha, Sargodha 40100, Pakistan.
* To whom correspondence should be addressed.
Manuscript received: 2024-01-28; accepted: 2024-03-11

※https://pewresearch.org/journalism/fact-sheet/social-media-and-news-fact-sheet/

responds to true news and can undermine the credibility of the whole news ecosystem[7–9]. Thus, it is important to analyze and detect fake news on OSN and other platforms.

Many manual tools and websites for fact-checking (e.g., PolitiFact©, FactCheck ‡, Snopes§, and Fiskkit¶) are currently available for the analysis, evaluation, and recognition of fake news. However, the problem of fake news analysis and detection is far from being solved. Now it is not possible to manually assess and verify every news or information due to the enormous amount of online data generated every minute, particularly on OSN platforms[1]. Moreover, determining the credibility of online news articles is difficult as fake news frequently contains wrong or false information mixed with certain facts[10]. In the last decade, computational approaches for fake news classification/detection have drawn a lot of interest. Fake news classification/detection methods, based on machine learning (ML) and deep learning (DL), can be broadly classified into two main groups: (1) content-based methods and (2) propagation-based methods[11–14]. Content-based methods detect fake news by analyzing the news content or information present in articles by either relying on a knowledge-based system[15, 16] or finding latent[13, 17] and non-latent (hand-crafted) features[16, 18] in the content.

Knowledge-based fake news detection methods can only detect false news but not fake news[1, 11]. Non-latent features are style-based and self-defined at various language levels, and various embedding and encoding techniques are used for these features. Latent features are features that are automatically generated by using matrix or tensor factorization, or DL techniques (for more details about non-latent and latent features, see Section 2). Selecting features or extracting non-latent features requires expertise, and some discovered linguistic clues might not be applicable to news or information. Latent features perform well, but they are difficult to comprehend. Moreover, content-based methods often face problems of computational efficiency, interpretability, scalability, and generalization because they are tested on limited datasets. As far as we are aware, no study has been published yet for fake news classification or detection based on pattern mining that focuses on a diverse set of datasets.

This study's two primary objectives are to (1) examine the application of sequential pattern mining (SPM)[19] for the reliable and accurate classification and detection of fake news from datasets in textual format, and (2) evaluate the SPM-based fake news classification approach on multiple datasets, and their combination, to get insights into its effectiveness and generalization ability across different data sources and characteristics. In the past, SPM is used extensively in various applications such as tourist movements analysis[20], bioinformatics[21, 22], market basket[23], text analysis[24], energy reduction in smarthomes[25], malware analysis[26], proof sequence analysis[27], and webpage click-stream analysis[28]. However, no one has explored its applicability for fake news analysis and classification yet. Based on the analysis of online news and information contents, we present a new content-based framework, called fake news analysis and classification using sequential pattern mining (FNACSPM) that provides:

- One approach based on SPM to analyze news contents. Using this approach, the datasets are first transformed into an appropriate learning format. Second, SPM techniques are employed to find frequent sequential patterns in the transformed datasets. Additionally, frequent sequential rules among fake and real news are identified.
- One fake news detection approach that uses frequent patterns (FreqP), discovered by using SPM algorithms. These patterns are then utilized in the fake news classification process. For classification, eight classifiers are utilized and comprehensive experiments are conducted by using various evaluation metrics to evaluate the effectiveness of the detection approach.

The proposed framework is evaluated on six datasets, and their combination for both binary and multi-class fake news classification. Obtained results indicate that using the FNACSPM to identify frequent sequential patterns in news and using these patterns yields improved classification results as compared to using all the news. It is also observed that logistic regression (LR) performed well, overall, for both types of classification. Using all the news in the classification process provided less accurate results and took more time. FNACSPM also outperformed state-of-the-art approaches for fake news classification/detection. By

©https://www.politifact.com/
‡https://www.factcheck.org/
§https://www.snopes.com/
¶https://fiskkit.com/

utilizing frequent patterns, this study offers valuable insights into the linguistic and semantic structures present in fake news. This aids in a deeper and better understanding of the characteristics and commonalities of misinformation, potentially assisting in the development of faster and more reliable strategies and models for detection.

The rest of the paper contains five sections. Section 2 examines the previous research on the analysis and classification of fake news by using ML and DL. Section 3 provides the details for the six datasets that are used in this study. FNACSPM is presented in Section 4 which offers approaches for fake news analysis and classification. The experimental results and comparison of FNACSPM with recent fake news classification/detection approaches is presented in Section 5. Finally, the paper is concluded with some remarks in Section 6.

2 Related Work

The two main categories of fake news detection techniques are content-based and propagation (or social context) based. Content-based approaches for fake news detection evaluate online news/information by examining textual information, visual information, or both. Content-based approaches use three common textual representations to analyze news: knowledge, style information (non-latent or general), and latent information[13, 15–18]. Propagation- or network-based methods analyze and identify fake news by investigating how news/information spreads over social networks. As the second category of propagation-based techniques is not relevant to this work, those are not discussed further.

The first representation, knowledge, is a subject, predicate, object (SPO) tuples set that is obtained from the text of online news. To identify fake news, knowledge-based methods assess the news authenticity by evaluating the knowledge discovered in news content that needs verification. One way to identify true knowledge is by comparing the obtained SPO from a news article with a knowledge graph (KG)[15, 16]. Generally, knowledge-based systems access the credibility of a given news, but they also face challenges related to the authenticity of the source(s) from which the KG is constructed. For fact-checking online news, it is necessary to identify not only parts of the news that are worth checking but also to have or create a KG that has all the possible "valuable" information and facts[1].

Style-based approaches for fake news detection, as opposed to knowledge-based systems, examine the news contents. To differentiate fake news from the truth, these methods use various general self-defined (non-latent) features that represent the writing style of online news. Non-latent features describe the style of the news (or content) at four language levels: (1) lexicon[11, 16, 17], (2) syntax[11, 29], (3) discourse[30, 31], and (4) semantic[18]. At the lexicon level, these approaches compute the lexicon frequency statistics with models such as bag-of-words (BOW)[11]. Part-of-speech (POS) taggers are used for shallow syntax tasks at the syntax level to compute the frequencies of POS[11, 29, 32]. Moreover, probabilistic context-free grammar (PCFG) can be used in style-based methods to examine and compute the rewrite rules frequencies[18, 29]. The rhetorical structure theory (RST) and tools for rhetorical parsing are used at the discourse level to compute the frequencies of rhetorical relations among sentences as features[30, 31]. In the fourth language level (semantic), phrases or lexicons that fit into each category of psycho-linguistic (like those that are described in linguistic inquiry and word count (LIWC)[18]) or that fit into each self-described psycho-linguistic feature are assigned frequencies. Experience and associated deception theories can be used to learn such features. Style-based approaches can also use term frequency–inverse document frequency (TF-IDF) and *n*-grams at various language levels to capture features of sequences of words (POS tagging, rewrite rules, etc.)

Latent textual features are generally used to create embeddings of the news content. These features can be extracted at the word, sentence, or document level. Embeddings are vectors that can be fed to classifiers within a traditional ML framework for fake news detection. In a DL framework, such embeddings can also be incorporated into neural networks and transformers[1, 11]. In theory, a latent representation can also be generated automatically by processes such as matrix or tensor factorization. The selection or extraction of general (non-latent) features is heavily influenced by experience and is weakly supported by fundamental theories from other disciplines. Latent features are difficult to comprehend and thus make it difficult to educate the public about fake/real news.

Content-based approaches do not take into account auxiliary information that plays a role in news propagation, such as news spreaders. Moreover, these approaches are sensitive to news content. A malicious entity can also manipulate the detection results by disguising their writing styles[14].

Next, we review style-based fake news detection studies published in the last seven years, based on traditional ML and DL.

The semi-supervised learning method[5] to detect breaking news rumors combined unsupervised and supervised learning objectives. Sitaula et al.[10] assessed the veracity of fake news, and they found that the total authors and the link for the creator of a news article with false information play important roles in identifying fake news. The theory-driven method[11] represented news articles with various manual features that captured content structure and writing style. A multi-modal approach was used in SAFE[13] to identify fake news that relied on similarities between news text and visual information. Reis et al.[33] used various supervised classifiers for fake news classification on some features from the literature and also on a new set of features. Some studies[34, 35] have compared various ML classifiers on different datasets for fake news detection. Ahmad et al.[36] investigated various textual properties of news and used an ensemble approach to detect fake news. TF-IDF and 23 classifiers were used in Ref. [37] to detect fake news in three datasets. Shu et al.[38] examined fake news datasets from various contexts to understand their characteristics and used various standard ML classifiers and social article fusion models for classification.

A hybrid framework, named BerConvoNet[12], combined bidirectional encoder representations from transformers (BERT) embeddings and convolutional neural network (CNN) to detect fake news. Two-level CNN with user response generator (TCNN-URG) framework[39] for fake news detection represented online articles at sentence and word levels for extraction of semantic information. The BERT model was applied in Ref. [40] to examine how the news title and the text (body) relate to fake news. Shu et al.[41] proposed dEFEND, an explainable fake news detection method that was based on recurrent neural network (RNN) and co-attention-based techniques. Reference [42] proposed a co-attention sub-network explainable detection model based on sentence-comment. Sastrawan et al.[43] combined CNN and RNN to identify fake news. Similarly, the approach[44] examined news headlines using BERT and a long short-term memory (LSTM) network. To classify fake news on OSN, the FakeBERT[45] approach combined CNN with BERT. An ensemble learning model based on BERT and text sentiment analysis was employed in Ref. [46] for improved detection of harmful news. Reference [47] used various word vector representation techniques with feed-forward neural network (FNN) and LSTM for fake news identification. FNDNet[48] is a deep CNN for fake news identification. Until now, the majority of the literature has focused on fake and real news identification as a binary classification problem. Some studies[49–59] worked on multi-class fake news identification. Recently, DL and neural network based techniques have been proposed and developed for fake news detection that incorporated multi-modal data such as social context[60], text, and image[13, 17, 61–65] and text with users' behavior and profiles[66].

3 Dataset

This study uses six publicly available datasets to analyze and validate the effectiveness of the proposed framework. Fact-checking experts provided the ground truth labels of true (real) or false (fake) for news articles in each of these six datasets. The George McIntire Dataset[67] is the first dataset, referred to as Dataset-1 (DS-1), containing 2291 fake news and 2285 real news. The second and third datasets are from FakeNewsNet Repository[38]. The websites (GossipCop and PolitiFact) for fact-checking were used to get both fake and true news. The GossipCop dataset, referred to as Dataset-2 (DS-2), contains 5335 (16 819) fake (real) news. The PolitiFact dataset, referred to as Dataset-3 (DS-3), contains 474 (798) fake (real) news stories.

The next three datasets are originally sourced from the Kaggle data science community. The BuzzFeed dataset[68], called Dataset-4 (DS-4), comprises both 91 real and fake news articles. Another dataset known as Fake News Classification[69], referred to as Dataset-5 (DS-5), contains 23 503 (21 418) fake (real) news articles. The last dataset used in this study is known as Fake and Real News Dataset[70] and is here called Dataset-6 (DS-6). It contains 34 980 (35 208) fake (real) news articles. The authors of this dataset integrated various famous datasets (i.e., McIntire,

Kaggle, BuzzFeed Political, and Reuters).

Statistical details about the six datasets are given in Table 1. Furthermore, the data present in the aforementioned six datasets are combined into one large dataset which is called the whole dataset (WDataset). In each of the six datasets, the articles vary in nature. WDataset goal is to access and evaluate the classifiers on a whole dataset that includes news and information from a wide range of diverse domains.

These datasets contain various attributes such as title, body, subject, video, and image. To prepare the data for analysis, we combined only text-based data (i.e., title and body) into a single attribute called "Text". For the datasets with only a title attribute, we simply used the title as the text. For the datasets with both title and body attributes, we concatenated the two attributes with a separator (e.g., a space or newline character) to form the text. For the datasets with additional attributes (such as subject, timestamps, video, and image links), we ignored them. This process of combining the attributes into a single "Text" attribute enabled us to easily feed the data into pattern mining tools for analysis and, consequently, our ML models for classification. It also helped to standardize the input format across all datasets and to make the modeling process less complex.

4 Methodology

The proposed FNACSPM framework (Fig. 1) for the analysis and detection/classification of fake news consists of four main parts:

(1) Datasets pre-processing and abstraction: The first step is to pre-process the datasets to put them into a suitable format for SPM. This is carried out by converting each sequence into a discrete sequence, where each distinct word is transformed into a distinct positive integer.

(2) Learning via SPM: The second step entails applying various algorithms for SPM on the abstracted datasets to find frequent words, their frequent patterns, and the sequential relationships among discovered frequent patterns.

Table 1 Datasets statistics.

Dataset	Fake news	True news	Feature	MiL	MaL	MeL
DS-1	2291	2285	T, B	23	32 674	4379.5
DS-2	5335	16 819	NURL, T, TID	10	204	69.5
DS-3	474	798	NURL, T, TID	10	340	60.7
DS-4	91	91	T, B, URL ¤	62	32 641	3257.3
DS-5	23 503	21 418	T, B, subject	30	32 888	2553.49
DS-6	34 980	35 208	T, B	15	33 026	3138.40

Note: T: title, B: body, NURL: news URL, TID: tweet ID, MiL: minimum length, MaL: maximum length, MeL: mean length, and ¤: top_img, authors, source, publish_date, movies, images, canonical_link, meta_data.

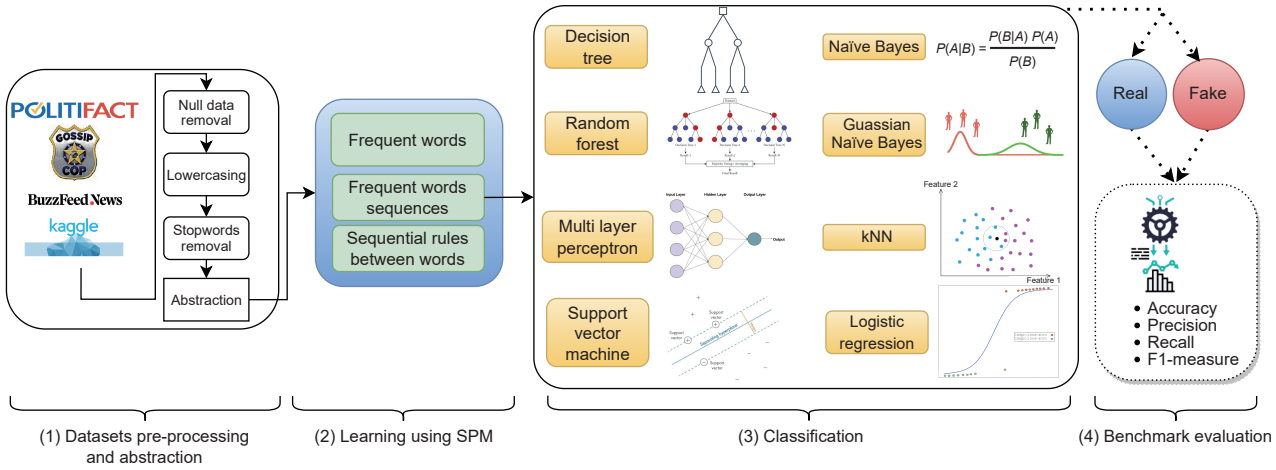


Fig. 1 FNACSPM framework, for fake news analysis and classification, consisting of four main steps: (1) Datasets pre-processing and abstraction, (2) learning using SPM, (3) classification via discovered frequent sequential patterns of words in the datasets by training various classifiers, and (4) evaluation of the framework by performing extensive experiments.

(3) Classification via frequent patterns: The third step is to use frequent patterns, discovered in Step (2), for the classification/detection of fake news. Various classifiers are utilized, and their performance is evaluated with various evaluation measures.

(4) Evaluation: Comprehensive experiments are carried out to access FNACSPM's performance and compare it with recent approaches for fake news detection.

In the next subsections, the first three parts of FNACSPM are explained in greater detail.

4.1 Dataset pre-processing and abstraction

The first step is data pre-processing, where cleaning operations such as lemmatization and stemming, and eliminating special characters, punctuation, and stop words are performed to prepare the data for further analysis. After the pre-processing, the sequences of words in the datasets are represented in an appropriate format. Table 2 provides the statistical details of six datasets after pre-processing. After pre-processing, the datasets are reduced, approximately, as follows: DS-1 (29%), DS-2 (24%), DS-3 (23%), DS-4 (35%), DS-5 (32%), and DS-6 (31%). For example, DS-1 (29%) indicates that the size of DS-1 is reduced to 29% of its original size as a result of the cleaning operations performed in pre-processing.

Let $W=\{w_1,w_2,...,w_m\}$ represent the set of words in a dataset. A words set WS is a set of words such that $\mathrm{WS}\subseteq W$. Set cardinality is represented by |WS|. A words set WS has a length of n (known as n-WS) if it contains n words, i.e., $|\mathrm{WS}|=n$. For instance, take $W=\{\text{trump, image, people, featured, via, even}\}$. Then, the set {trump, people, via, even} is a WS containing four words. A total order relation on words is defined, indicated by the $\prec$, to aid in the identification of patterns. In the framework's implementation, this lexicographical order is employed as the processing order for pattern searching.

Table 2 Datasets statistics (after pre-processing).

Dataset	MiL	MaL	MeL
DS-1	17	21 875	3144
DS-2	4	174	53.1
DS-3	10	279	47.1
DS-4	39	20 203	2124.3
DS-5	22	3279	1759.5
DS-6	9	22 831	2189.1

A sequence of words is basically a list of words set $S=\langle \mathrm{WS}_1, \mathrm{WS}_2, ..., \mathrm{WS}_n\rangle$, such that $\mathrm{WS}_i\subseteq\mathrm{WS}$ $(1\leqslant i\leqslant n)$. A words corpus dataset, $\mathrm{WCD}=\langle S_1,S_2,...,S_n\rangle$, is a list of words sequences. In WCD, a sequence is associated with an ID. Table 3 shows a WCD containing four word sequences. According to the first sequence, the word "trump" is followed by "featured", then "via", and "show".

The word sequences are transformed into integer sequences. This is done to prepare the datasets in a format that SPM algorithms can process more easily. Each line in the final transformed datasets denotes a word sequence for a fake/real news. In sequences, a unique positive integer is used to replace each unique word type. For instance, the words "trump" and "featured" are changed to 1 and 3, respectively. A single space and the negative number −1 are used to separate the words in sequences from one another. News (sequence) ends when a negative number (−2) appears at the end of a line. Table 3 also provides the conversion of four word sequences into integer sequences.

4.2 Learning via SPM

WCD is analyzed, in the second step, to discover frequent patterns. Suppose that $S_a=\langle a_1,a_2,...,a_n\rangle$ and $S_b=\langle b_1,b_2,...,b_m\rangle$ are two sequences of words. S_b contains S_a $(S_a\sqsubseteq S_b)$ if and only if there exists integer $1\leqslant k_1<k_2<\cdots<k_n\leqslant m$, s.t., $a_1\subseteq b_{k1}, a_2\subseteq b_{k2},...,$ $a_n\subseteq b_{kn}$. S_a is considered to be S_b's subsequence if S_b contains S_a. The importance and interestingness of a subsequence in SPM can be found via various

Table 3 Sample of WCD.

ID	Sequence	Representation of words sequences as integer sequences
1	⟨{trump},{featured},{via},{show}⟩	1 −1 3 −1 4 −1 5 −1 −2
2	⟨{image},{getty},{image},{image},{said},{president}⟩	6 −1 12 −1 6 −1 6 −1 32 −1 23 −1 −2
3	⟨{one},{donald},{image},{said},{reuters},{release},{image},{american}⟩	7 −1 11 −1 6 −1 32 −1 15 −1 18 −1 6 −1 22 −1 −2
4	⟨{republican},{american},{horror},{image},{one},{republican},{ring},{american},{getty}⟩	19 −1 22 −1 14 −1 6 −1 7 −1 19 −1 31 −1 22 −1 12 −1 −2

measures, in which the support measure is mostly used. In a WCD, the support of S_a is the total number of sequences (S) that contain S_a, which is denoted by the symbol $\sup(S_a)$:

$$\sup(S_a) = |\{S | S_a \sqsubseteq S \wedge S \in \text{WCD}\}| \quad (1)$$

In a sequential dataset, such as WCD, SPM deals with the enumeration problem to find all the frequent subsequences. If support of a sequence S is equal to or greater than a user-provided threshold of minimum support ($\sup(S) \geqslant \text{minsup}$), then S is said to be a frequent sequences. Sequences can have up to $2^n - 1$ distinct subsequences, where n represents the total number of items. For most datasets, finding the support of all potential subsequences using the naive method is not possible[71]. However, over the past two decades, various effective algorithms have been developed that can discover all sequential patterns without having to search through all the potential subsequences.

SPM algorithms use the s-extensions and i-extensions operations to move through the search space of sequential patterns. For an item y, S_b is an s-extension of S_x, if $S_b = \langle x_1, x_2, ..., x_n, \{y\}\rangle$. On the other hand, S_c is an i-extension of S_x if $S_c = \langle x_1, x_2, ..., x_n \cup \{y\}\rangle$. In general, SPM algorithms use a depth-first or breadth-first search with various optimizations and data structures.

Frequent itemset mining (FIM), a special case of SPM, deals with analyzing records where the sequential ordering among items is not considered. The first and best-known FIM method, called Apriori[72], can discover frequent itemsets (like word sets) in large databases. Apriori first discovers items (e.g., words) in databases that occur frequently. Then, discovered items are expanded to discover larger itemsets that often appear adequately. Besides finding itemsets, Apriori can also finds relationships (association rules) among items. Multiple memory efficient and fast algorithms can be used for FIM, which find the same patterns. These new algorithms use different types of data structures, optimization techniques, and search strategies.

One SPM algorithm used in this work is top-k sequential (TKS)[73], which can find the top-k most common sequential patterns in a database (or dataset), where a user sets the parameter k. TKS finds the desired k patterns by applying the sequential pattern mining (SPAM)'s candidate generation procedure and a vertical database representation (VDR). The VRD facilitates the counting of patterns without expensive database scans. Thus, SPM algorithms based on VDR generally work more effectively on dense or long sequences. Other strategies for search space reduction are also used in TKS, along with the data structure of the precedence map (PMAP). These methods allow TKS to lower the number of costly operations like bit vector intersections. Another SPM algorithm used in this work is CM-SPAM[74]. It scans the search space of a dataset or database to find frequent sequential patterns. CM-SPAM uses the data structure of co-occurrence MAP (CMAP) that stores items co-occurrence information. CM-SPAM uses a generic mechanism to prune the search space via the VDR. The reader may refer to Refs. [73, 74] for more details about the two aforementioned algorithms for SPM.

The aforementioned algorithms have the main drawback that they may discover too many sequential patterns, most of which are not interesting or important for users. Sequential patterns appearing frequently in a database with low confidence are of no value in tasks of prediction or decision-making. Due to this, there is another pattern type known as sequential rules. A pattern as a sequential rule considers both the confidence (conditional probability) that some events (words in this work) will follow or be followed by others in addition to the support of events. A sequential rule $X \rightarrow Y$ in this work represents a relationship between two WSs $X, Y \subseteq W$, s.t., $X, Y \neq \varnothing$ and $X \cap Y = \varnothing$. According to the rule $r: X \rightarrow Y$, if words from X appear in a series, then words from Y will follow in the same sequence. S_x contains X if and only if $X \subseteq \bigcup_{i=1}^{n} x_i$. Similarly, S_x contains the rule r if an integer k exists s.t., $1 \leqslant k < n$, $X \subseteq \bigcup_{i=1}^{k} x_i$ and $Y \subseteq \cup_{i=k+1}^{n} x_i$. In a dataset WCD, the confidence of a rule r is

$$\text{conf}_{\text{WCD}}(r) = \frac{|\{S | r \sqsubseteq S \wedge S \in \text{WCD}\}|}{|\{S | X \sqsubseteq S \wedge S \in \text{WCD}\}|} \quad (2)$$

Similarly, in WCD, the support of a rule r is

$$\sup_{\text{WCD}}(r) = \frac{|\{S | r \sqsubseteq S \wedge S \in \text{WCD}\}|}{|\text{WCD}|} \quad (3)$$

For a WCD and a user-specified minimum support threshold (minsup), a rule r is considered a frequent sequential rule if and only if $\sup_{\text{WCD}}(r) \geqslant \text{minsup}$. Similarly, for a user-specified minimum confidence threshold (minconf), a rule r is considered a valid sequential rule if (1) it is frequent and (2) $\text{conf}_{\text{WCD}}(r) \geqslant \text{minconf}$. Enumerating all the valid

sequential rules in a dataset is the goal of sequential rule mining. In this work, the ERMiner algorithm[75] is used to discover frequent sequential rules in fake news datasets. A VDR is employed by ERMiner. The rules search space is investigated by the use of equivalency classes of rules with identical antecedent and consequent. Moreover, the search space of sequential rules is investigated by using two procedures (called left and right merges). ERMiner is more effective than earlier algorithms for mining sequential rules because it uses the sparse count matrix (SCM) approach for search space pruning. In summary, SPM algorithms are different from each other on the basis of (1) the use of a depth-first or breadth-first search, (2) the use of a VDR or horizontal representations and of particular data structures, and (3) how the support measure is calculated to find those frequent patterns that satisfy minsup constraint.

4.3 Classification

The third step of the framework involves the fake news classification using the sequential frequent patterns discovered with SPM. The lengths of news articles are generally long, for example, see Tables 1 and 2. A close inspection of the WCD reveals that the majority of the sequences (both real and fake) contain the same words repeated multiple times. This word repetition in online news can be avoided during the classification process by treating contiguous identical words as a single word.

More precisely, FNACSPM uses the sequential frequent patterns, found with the SPM algorithms, to classify fake and real news in datasets. For classification, two methods (binary and multi-class (MC)) are employed. Two types of binary classification are considered for training a classifier so that it classifies each fake or real news.

Type 1: Each dataset is considered separately in the first type. For a separate dataset, binary classification assigns "fake" or "real" labels to each sequence (news) corresponding to that class.

Type 2: In this type, all datasets are combined together to create one dataset, that is used in training a model for the classification of news (sequence) type. This classification type labels "1" to sequence(s) that originally belonged to that dataset type and labels "Others" (or 0) to all other sequences.

Definition 1 Assume that NT denotes the set of all news types (classes). A sequence S is labeled with regard to c for a chosen class $c \in \mathrm{NT}$:

$$S_c = \begin{cases} 1, & \text{if } s \in c; \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

The news class labels, according to Eq. (4), are labeled to 1 for those that belong to c, while others are labeled as 0 (or "Others"). A simple example is provided to illustrate this process. For instance, if the news type of interest in DS-1 is fake, then Eq. (4) assigns 1 to all the DS-1 sequences belonging to the fake type and 0 to all other sequences in DS-1 and other sequences in the whole dataset.

A second way to train and test classifiers for fake news classification/detection is to use MC classification. In the context of this work, each sequence in the whole dataset (DS-7 or WDS), which combines all six datasets, is labeled with its respective class name. There are 12 classes in total, as shown in Table 4. In MC classification, a classifier is trained to correctly label sequences according to those classes.

For classification, seven standard ML algorithms and one DL algorithm are used, which are: (1) Bernoulli Naive Bayes (BNB), (2) Gaussian Naive Bayes (GNB), (3) decision tree (DT), (4) random forest (RF), (5) support vector machine (SVM), (6) k-Nearest Neighbors (kNN), (7) LR, and (8) multi layer perceptron (MLP). We chose these eight classifiers for this work because most previous studies on fake news analysis and detection also used them.

The performance of classifiers is assessed using the following seven metrics: accuracy (ACC), precision (P), F1 score, recall (R), Matthews correlation

Table 4 MC labeling of sequences from the whole dataset.

Dataset	Class	MC class
DS-1	Fake	DS1-F
	Real	DS1-R
DS-2	Fake	DS2-F
	Real	DS2-R
DS-3	Fake	DS3-F
	Real	DS3-R
DS-4	Fake	DS4-F
	Real	DS4-R
DS-5	Fake	DS5-F
	Real	DS5-R
DS-6	Fake	DS6-F
	Real	DS6-R

coefficient (MCC), area under the receiver operating characteristic curve (AUC) and area under the precision-recall curve (AUPRC). The seven measures are defined as follows:

$$\text{ACC} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{5}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{7}$$

$$\text{F1} = 2\times\frac{P\times R}{P+R} \tag{8}$$

$$\text{MCC} = \frac{\text{TP}\times\text{TN}-\text{FP}\times\text{FN}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}} \tag{9}$$

$$\text{AUC} = \int_0^1 \text{TPR}(\text{dFPR}) \tag{10}$$

$$\text{AUPRC} = \sum_{i=1}^{n}\frac{(R_i-R_{i-1})\times(P_i+P_{i-1})}{2} \tag{11}$$

TP = true positive, TN = true negative, FP = false positive, and FN = false negative. In Eq. (10), TPR represents the recall (R) and dFPR is the derivative of the false positive rate (FPR), that is equal to $\frac{\text{FP}}{\text{FP}+\text{TN}}$. P_i and R_i in Eq. (11) represent the values for precision and recall, respectively, at the i-th decision threshold.

5 Result

A computer equipped with 16 GB RAM and an 11th-generation Core i5 processor was utilized for carrying out experiments. A JAVA-based open-source library, called SPMF[76], was used to examine and find patterns in the datasets. Implementations of over 250 data and pattern mining algorithms are available in this library. For classification purposes, Python is used, where a variety of libraries are utilized, including scikit-learn[77] for ML algorithms, NumPy for numerical computations, and Pandas for data manipulation. In the text pre-processing phase, the TF-IDF was used, utilizing the "TfidfVectorizer" module from the scikit-learn library. To ensure reliable model evaluation, the dataset is split into training and testing sets (80% training and 20% testing) by using the train_test_split function from the scikit-learn. This function facilitated the random partitioning of the data, allocating a specified proportion for training the models and the remaining portion for evaluating their performance. Next, we discuss the obtained results by using the SPM algorithms on the abstracted datasets.

5.1 Discovered pattern and rule

The Apriori is first applied to the transformed datasets to find frequent words. Both fake and real news contain many similar words (Fig. 2). We found that in the first 3000 frequent words discovered by Apriori in fake and real news, approximately 93% are similar to each other. However, frequent sets of words are unordered. Besides, Apriori does not guarantee that words from a word set (WS) occur in a sequence consecutively. As a result, Apriori's long patterns are not interesting or important and offer no helpful information. Apriori is unable to identify sequential patterns because it ignores the relationships between words in order. Next, we present the outcomes for SPM algorithms that improve upon Apriori.

More important and meaningful patterns can be discovered in data using SPM algorithms like TKS, CM-SPAM, and ERMiner. The top-k sequential patterns of words in the datasets are discovered using the TKS algorithm. CM-SPAM algorithm needs the minsup threshold to be set, unlike TKS. Table 5 lists some frequent sequential patterns of words that are found in six datasets with TKS and CM-SPAM. From discovered patterns, one can find useful and interesting details about the frequent occurrences of words in fake and real news. The bold patterns represent fake ones

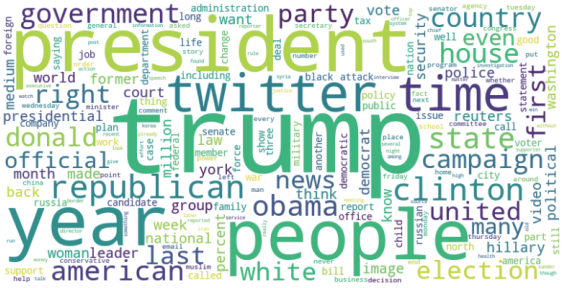


Fig. 2 Frequent words discovered in the whole dataset.

Table 5 Frequent sequential patterns in six datasets.

Dataset	Frequent sequential pattern	
	TKS	CM-SPAM
DS-1	new york	**fbi email**
	cruz kasich	clinton win
	trump vote	donald trump
	america news	hiliray clinton
	trump delegate	**onion america**
	general election	general election
	fbi investigation	**trump woman problem**
	onion finest news source	**america finest news source**
DS-2	**image via image**	tv scoop award
	would featured via	**kim kanye west**
	cut tie jazz jennings	**brad angelina jolie**
	khloe kardashian baby	**kim kardashian west**
	trump one getty image	**prince harry meghan**
	selena gomez justin bieber	blake shelton stefani
	getty donald trump president	linkin park bennington
	shannon open relationship beador	jennifer aniston justin theroux
DS-3	tax	tax
	obama	obama
	debate	debate
	breaking	**breaking**
	transcript	transcript
	michelle obama	**michelle obama**
	president trump	**president trump**
	news latest video	**news latest video**
DS-4	**get life**	**get life**
	like one	**like one**
	people one	**people one**
	new get time one	hillary clinton said
	donald trump said	donald trump story
	hillary clinton said	also one trump donald
	trump continued trump	**like one hillary clinton**
	story campaign donald trump	trump continued trump
DS-5	reuters said said	**trump said trump**
	said republican said	**donald twitter image**
	york state house said	said republican would
	donald twitter image	washington u said house
	washington reuters said	president washington said
	donald trump one said	**president donald trump like**
	president donald trump image	**donald trump image feature**
	twitter one trump featured image	**trump one trump featured image**
DS-6	**york new one**	state said one
	state said would	**one would said**
	trump said make	donald trump year
	president year said	featured image said
	president new said	news president said
	state hillary clinton	**american make trump**
	donald trump image	reuters said washington
	american people trump	**republican hillary clinton**

while others represent real ones. We find that some fake and real patterns are similar to each other and there is some difference among the patterns found in the six datasets. Overall, it was observed that using pattern mining on news was quite fast. However, for datasets that contain long news sequences, we need to fine-tune some parameters of both algorithms to find frequent sequential patterns.

Table 6 shows the relationships between frequent words that are identified in each dataset via the ERMiner algorithm. It was observed that different dataset requires different parameter settings (minsup and minconf) before they start giving sequential rules. For example, for DS-1, minconf = 15%. This indicates that rules should therefore have a minimum of 15% confidence. The third rule in DS-1 indicates that the word "campaign" is followed by the word "clinton". Similarly, the last rule indicates that the words "new" and "people" are followed by "hillary" and "state", respectively. ERMiner offered some useful dependencies and relationships that are present among frequent words. On six datasets, the three SPM algorithms performed effectively. The obtained results showed a clear association between the total number of words in news sequences and the effectiveness of algorithms for sequential patterns. In Table 6, X in DS-6 represents that ERminer was unable to find rules in the set of fake news due to running out of time or memory.

Table 6 Extracted sequential rules by using the ERMiner algorithm.

Dataset	Extracted sequential rule
DS-1	donald→republican
	state → hillary, clinton
	campaign → clinton
	state → time, year
	party, campaign → president, said, state
	trump, donald → said, clinton, hillary, campaign
	clinton, state → people
	new, people → hillary, state
DS-2	**brad → pitt**
	miley, cyrus → liam
	kim → kardashian
	selena, gomez → justin, bieber
	beyonce → jay, z
	wedding, prince, harry → meghan, markle
	jennifer, leaving → biggest, mistake
	brad, pitt → angelina, jolie
DS-3	week → transcript
	news, latest → video
	office → news, breaking
	donald → paid
	senate, call, vote → congress
	trump, executive → order
	queen → say, elizabeth
	kim, jong → trump, north
DS-4	**new → president**
	get, life → people
	thing, get → short, life
	hillary → trump
	hillary, clinton → donald, trump
	hillary → said, clinton, trump
	donald, continued → trump
	thing, know → get, trial
DS-5	reuters → president
	washington → persident, trump
	official, house → state, year
	featured → image
	government, last, president → new, republican
	trump → people, president
	obama, president → time, image
	twitter, pic, country → white, house
DS-6	**X**
	video → trump
	X
	president, donald → trump
	donald → image, trump
	X
	trump → image, featured
	X

5.2 Result for classification

The experimental results for both binary and MC classification on six datasets are presented in this section. The eight classifiers were used for two cases:

Case 1: All the words, after prepossessing, in news sequences are used in the classification process.

Case 2: The frequent sequential patterns, found with two SPM algorithms are used in the classification process.

TKS and CM-SPAM algorithms are used in Case 2 to find frequent 100, 200, 400, and 600 patterns of words in each dataset. Four different numbers of patterns were considered to investigate whether or not the number of patterns affects how well classification models perform. After discovery, the frequent patterns are further pre-processed to ensure that in each pattern there are at least 3 distinct frequent words. For the classification in both cases, the default hyperparameters for algorithms were as follows:

- BNB with an α value of 1.0;
- GNB with no significant hyperparameters to tune;
- DT with a criterion of "gini", a splitter of "best", no maximum depth limit, a minimum samples split, and leaf of 2 and 1, respectively;
- RF with 100 estimators, "gini" criterion, no maximum depth limit, minimum samples split of 2, and minimum samples leaf of 1;
- MLP with a hidden layer size of 600, "tanh" activation function, "adam" solver, an α value of 0.0001, invscaling learning rate, and learning rate initialization of 0.001;
- SVM with a C value of 1.0, a "radial basis function (rbf)" kernel, a degree of 3, and a "scale" γ value;
- kNN with 2 neighbors, "uniform" weight scheme, "auto" algorithm, leaf size of 30, and Euclidean distance metric ($p = 2$).
- LR with a C value of 1.0, "Limited-memory Broyden-Fletcher-Goldfarb-Shanno (lbfgs)" solver, and a maximum iteration limit of 100.

5.2.1 Binary classification result

Table 7 provides the results of binary classification for Case 1 (all words are used for classification). The format $\frac{\text{Acc}}{\text{Time}}$ is used for classifiers. For example, the entry $\frac{0.84}{9.5}$ represents that BNB achieved an accuracy of 0.84 on DS-1 and took 9.5 s to terminate. For Case 1, 10 000 random and fake news articles were used in DS-5 and DS-6 in the classification. For the WDS, proportionate sampling was used to deal with data imbalance. It involves selecting a subset of data from each dataset in a way that maintains the original class distribution. This helps to ensure that each class is represented proportionally in the 10 000 randomly selected articles. LR achieved the highest accuracy of 0.83 on average, on all datasets. On DS-5, DT and RF achieved the highest accuracy of 0.99, followed by LR (0.98). The ranking of classifiers based on average accuracy is in the order LR > RF > DT > kNN > BNB > GNB. SVM and MLP are not included in the ranking as they were unable to produce results within 5 h on various datasets.

kNN performed best in terms of computational time, followed by LR. The ranking of classifiers based on time is kNN > LR > BNB > GNB > RF > DT. RF and LR performed better overall but RF was slow compared to LR. The accuracy of classifiers, except RF, was low on DS-4 compared to the other five datasets. This is because DS-4 contained few fake and real news. For the whole dataset, the results for all the classifiers decreased significantly. In Case 1, interestingly we find that classifiers achieved the highest accuracy, except kNN, on the Fake

Table 7 Classifiers' accuracy and running time for binary classification (Case 1).

Dataset	Result	BNB	GNB	DT	RF	MLP	SVM	kNN	LR
DS-1	Accuracy	0.84	0.81	0.82	0.90	0.93	**0.93**	0.84	0.92
	Running time (s)	9.5	17.8	55.8	28.8	3402.4	2845.6	3.5	8.4
DS-2	Accuracy	0.84	0.65	0.80	0.84	0.77	–	0.81	**0.85**
	Running time (s)	17.4	23.4	1677.8	943.4	6390.5	–	26.7	10.9
DS-3	Accuracy	0.75	0.75	0.76	0.75	0.79	0.81	**0.86**	0.81
	Running time (s)	0.1	0.1	0.9	1.8	34.9	4.9	0.3	0.07
DS-4	Accuracy	0.62	0.59	0.76	**0.80**	0.65	0.70	0.73	0.69
	Running time (s)	0.2	0.1	0.5	0.7	27.7	0.7	0.4	0.08
DS-5	Accuracy	0.85	0.90	**0.99**	**0.99**	–	–	0.66	0.98
	Running time (s)	28.2	58.1	60.6	72.6	–	–	8.6	19.4
DS-6	Accuracy	0.84	0.75	0.91	0.90	0.92	–	0.86	**0.91**
	Running time (s)	62.9	329.2	329.2	118.3	3749.1	–	15.1	30.7
WDS	Accuracy	0.58	0.57	0.57	0.61	–	–	0.62	**0.65**
	Running time (s)	83.6	167.7	217.7	212.5	–	–	12.9	43.1
Average	Accuracy	0.76	0.71	0.80	0.82	–	–	0.76	**0.83**
	Running time (s)	28.8	85.2	334.6	196.8	–	–	**9.64**	16.1

News Classification dataset (DS-5). In previous studies[10–14, 35–38, 41, 42, 44, 60, 63] that used multiple datasets, classifiers performed better on the PolitoFact dataset (DS-3). Figure 3 shows the overall results for LR, which performed best for Case 1.

Classifiers in Case 2 performed significantly better than classifiers in Case 1 (Table 8). Overall for varying pattern lengths, it is observed that classifiers achieved better results on patterns found with TKS as compared to CM-SPAM. Moreover, classification models achieved the highest accuracy with 400 patterns, followed by 600, 200, and 100, respectively. The ranking of classifiers based on accuracy on patterns found by using TKS (CM-SPAM) is in the order LR (LR) ≈(>) SVM (MLP) >(>) MLP (BNB) >(>) RF (SVM) >(>) DT (kNN) >(>) BNB (RF) >(>) kNN (DT) >(>) GNB (GNB). Table 9 lists the overall results for LR, which performed best on patterns discovered using TKS. LR performed best on DS-2 (GossipCop), followed by DS-3 (PolitiFact). Obtained results so far clearly indicate that using all the words not only provides less accurate results, compared to using frequent patterns, but also takes much time and memory.

5.2.2 MC classification result

Tables 10 and 11 provides the obtained MC classification results for both cases. For Case 1, LR and SVM performed better accuracy (0.77) compared to others. The ranking of classifiers based on accuracy is in the order LR ≈ SVM > RF > MLP > kNN > DT ≈ BNB > GNB. Interestingly, the classifiers achieved high accuracy for MC as compared to binary classification for Case 1. Computation-wise, kNN performed better, followed by BNB, while SVM performed worst, followed by MLP.

For Case 2, LR performed better (average accuracy 0.767) compared to others. For Case 2, the ranking of classifiers based on average accuracy is in the order LR ≈ MLP ≈ SVM > BNB > RF > DT > kNN > GNB. When compared to TKS, all classifiers performed generally better on patterns found with CM-SPAM. Moreover, classifiers achieved high accuracy on 600 patterns, followed by 400, 200, and 100, respectively. Again, the time taken by classifiers reduced significantly in Case 2 as compared to Case 1. For Case 2, GNB performed worst.

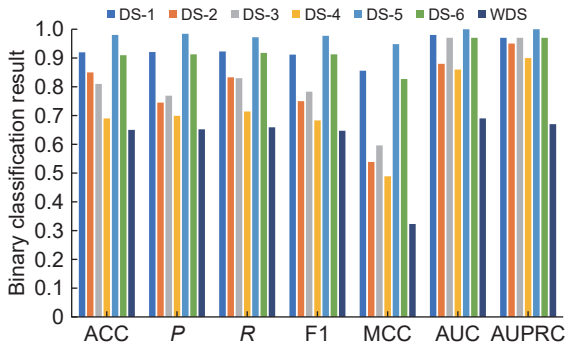


Fig. 3 Binary classification results for LR (Case 1).

Interestingly, the majority of the classifiers, except BNB, GNB, and MLP, performed better in Case 1 as compared to the results obtained by using TKS's 100 and 200 frequent patterns. Conversely, all classifiers performed better with 600 patterns obtained by CM-SPAM. Moreover, the classifiers in Case 1 performed better in some cases than Case 2. Figure 4 provides the overall results of LR for Case 1 and Case 2.

In summary, the obtained results show that frequent patterns discovered in news can be used efficiently to classify and detect fake news instead of providing the whole news sequences. Using all the words (or the entire news) not only provides less accurate results, compared to using frequent patterns, but also takes much more time. From Tables 1 and 2, it is evident that news articles typically consist of thousands of words. However, the sequential patterns discovered by the TKS algorithm contain 74 words at most. For binary classification, it was observed that classification models performed better, overall, on TKS's patterns as compared to CM-SPAM's patterns. The opposite was true for MC classification. However, classifiers performed better on TKS's 600 patterns as compared to CM-SPAM's 600 patterns. For binary (MC) classification, classifiers performed better on 400 (600) patterns.

5.2.3 Comparison

FNACSPM is compared in this section with state-of-the-art approaches (published during 2017–2023) for fake news classification and detection.

Table 12 provides a comparison for binary and MC classification. For binary classification, the majority of prior studies used multiple datasets. The maximum number of 4 datasets was used in Refs. [12, 34, 36, 62]. The bold datasets in column 2 of Table 12 for binary classification are those datasets with which the corresponding learning model achieved the highest results. For example, Ref. [10] achieved the highest F1 of 0.82 on the PolitiFact dataset using linear SVM.

Table 8 Classifiers accuracy and running time for binary classification on frequent sequential patterns discovered by TKS (CM-SPAM).

Dataset	FreqP	Result	BNB	GNB	DT	RF	MLP	SVM	kNN	LR
DS-1	100	Accuracy	0.78 (0.80)	0.80 (0.88)	**0.90** (0.88)	0.80 (0.85)	0.85 (0.82)	0.88 (0.82)	0.80 (0.80)	0.88 (0.80)
		Running time (s)	0.2 (0.2)	0.01 (0.01)	0.07 (0.07)	0.3 (0.2)	0.7 (0.7)	0.02 (0.02)	0.2 (0.4)	0.02 (0.02)
	200	Accuracy	0.85 (0.82)	0.82 (0.81)	**0.96** (0.85)	0.85 (0.75)	0.89 (0.81)	0.90 (0.82)	0.85 (0.81)	0.91 (0.85)
		Running time (s)	0.2 (0.3)	0.01 (0.01)	0.07 (0.07)	0.4 (0.3)	2.2 (2.6)	0.05 (0.06)	0.2 (0.3)	0.01 (0.01)
	400	Accuracy	0.89 (0.86)	0.90 (0.92)	0.89 (0.84)	0.89 (0.85)	0.92 (0.91)	**0.94** (0.88)	0.82 (0.78)	0.94 (0.88)
		Running time (s)	0.4 (0.2)	0.02 (0.01)	0.09 (0.08)	0.5 (0.4)	4.4 (4.03)	0.2 (0.1)	0.3 (0.2)	0.03 (0.02)
	600	Accuracy	0.95 (0.92)	**0.96** (0.92)	0.88 (0.86)	0.93 (0.92)	0.93 (0.95)	0.94 (0.93)	0.85 (0.84)	0.95 (0.94)
		Running time (s)	0.2 (0.3)	0.02 (0.03)	0.08 (0.09)	0.6 (0.5)	5.9 (9.1)	0.2 (0.8)	0.2 (0.2)	0.02 (0.03)
DS-2	100	Accuracy	**1** (0.80)	**1** (0.80)	0.95 (0.68)	**1** (0.72)	**1** (0.70)	**1** (0.68)	**1** (0.65)	**1** (0.70)
		Running time (s)	0.3 (0.2)	0.01 (0.01)	0.07 (0.08)	0.2 (0.3)	0.7 (0.8)	0.01 (0.02)	0.3 (0.2)	0.01 (0.02)
	200	Accuracy	0.94 (0.90)	0.94 (0.89)	**1** (0.88)	**1** (0.88)	0.94 (0.90)	**1** (0.94)	**1** (0.78)	**1** (0.94)
		Running time (s)	0.2 (0.3)	0.01 (0.02)	0.07 (0.08)	0.2 (0.3)	0.7 (2.7)	0.02 (0.05)	0.2 (0.3)	0.01 (0.02)
	400	Accuracy	0.99 (0.94)	0.99 (0.94)	**1** (0.91)	**1** (0.92)	0.99 (0.94)	**1** (0.94)	**1** (0.92)	**1** (0.94)
		Running time (s)	0.2 (0.3)	0.01 (0.02)	0.07 (0.08)	0.3 (0.4)	1.8 (4.1)	0.08 (0.3)	0.3 (0.2)	0.02 (0.02)
	600	Accuracy	0.99 (0.95)	0.99 (0.96)	**1** (0.93)	**1** (0.95)	0.99 (0.95)	**1** (0.97)	**1** (0.91)	**1** (0.96)
		Running time (s)	0.2 (0.3)	0.01 (0.02)	0.07 (0.08)	0.3 (0.4)	1.8 (4.7)	0.1 (0.5)	0.4 (0.3)	0.04 (0.03)
DS-3	100	Accuracy	**1** (0.89)	0.48 (0.55)	**1** (0.82)	**1** (0.91)	**1** (0.91)	**1** (0.86)	**1** (0.89)	**1** (0.89)
		Running time (s)	0.2 (0.3)	0.01 (0.01)	0.07 (0.07)	0.4 (0.3)	0.8 (0.9)	0.01 (0.02)	0.3 (0.2)	0.01 (0.02)
	200	Accuracy	**1** (0.92)	0.68 (0.54)	**1** (0.56)	**1** (0.57)	**1** (0.91)	**1** (0.92)	**1** (0.92)	**1** (0.92)
		Running time (s)	0.2 (0.3)	0.01 (0.02)	0.09 (0.08)	0.3 (0.4)	0.8 (3.3)	0.02 (0.05)	0.2 (0.4)	0.01 (0.02)
	400	Accuracy	**0.98** (0.84)	0.69 (0.61)	**0.98** (0.57)	**0.98** (0.56)	**0.98** (0.83)	**0.98** (0.62)	0.96 (0.82)	**0.98** (0.84)
		Running time (s)	0.2 (0.3)	0.01 (0.02)	0.07 (0.1)	0.3 (0.7)	1.5 (12.5)	0.06 (0.3)	0.2 (0.3)	0.02 (0.03)
	600	Accuracy	0.98 (0.80)	0.76 (0.52)	0.96 (0.50)	0.98 (0.51)	0.98 (0.78)	**0.99** (0.79)	0.90 (0.79)	0.97 (0.81)
		Running time (s)	0.2 (0.3)	0.01 (0.04)	0.07 (0.3)	0.3 (1.1)	1.7 (21.5)	0.1 (1.1)	0.3 (0.2)	0.02 (0.04)
DS-4	100	Accuracy	**0.98** (0.92)	**0.98** (0.95)	0.95 (0.95)	0.95 (0.95)	0.95 (0.95)	0.95 (0.95)	0.92 (0.95)	0.95 (0.95)
		Running time (s)	0.2 (0.2)	0.01 (0.01)	0.07 (0.06)	0.2 (0.2)	0.8 (0.7)	0.02 (0.01)	0.3 (0.2)	0.01 (0.02)
	200	Accuracy	0.96 (**0.98**)	0.94 (0.91)	0.94 (0.96)	0.95 (0.96)	0.96 (0.96)	0.96 (90.6)	0.94 (0.96)	0.96 (0.96)
		Running time (s)	0.3 (0.2)	0.01 (0.02)	0.09 (0.07)	0.3 (0.4)	1.6 (1.9)	0.03 (0.02)	0.4 (0.2)	0.02 (0.01)
	400	Accuracy	**0.98** (0.97)	0.96 (0.93)	0.96 (0.96)	**0.98** (0.96)	**0.98** (**0.98**)	**0.98** (**0.98**)	0.96 (0.95)	**0.98** (**0.98**)
		Running time (s)	0.3 (0.2)	0.02 (0.01)	0.2 (0.07)	0.4 (0.3)	2.5 (2.6)	0.2 (0.04)	0.3 (0.2)	0.03 (0.02)
	600	Accuracy	**0.96** (**0.96**)	0.95 (0.93)	0.95 (0.93)	**0.96** (0.95)	0.95 (**0.96**)	0.95 (0.95)	0.95 (0.94)	0.95 (**0.96**)
		Running time (s)	0.2 (0.3)	0.01 (0.02)	0.08 (0.07)	0.4 (0.3)	3.2 (2.8)	0.2 (0.07)	0.2 (0.3)	0.02 (0.03)
DS-5	100	Accuracy	0.92 (**0.95**)	0.82 (0.90)	0.88 (**0.95**)	0.89 (**0.95**)	0.90 (**0.95**)	0.90 (**0.95**)	0.85 (**0.95**)	0.90 (**0.95**)
		Running time (s)	0.1 (0.2)	0.02 (0.01)	0.1 (0.07)	0.4 (0.3)	1.2 (0.6)	0.03 (0.01)	0.3 (0.2)	0.03 (0.01)
	200	Accuracy	0.88 (**0.98**)	0.81 (0.91)	0.90 (0.96)	0.91 (0.96)	0.84 (0.96)	0.89 (0.96)	0.88 (0.96)	0.79 (0.96)
		Running time (s)	0.1 (0.2)	0.02 (0.01)	0.09 (0.06)	0.4 (0.2)	1.2 (1.1)	0.07 (0.02)	0.3 (0.2)	0.02 (0.01)
	400	Accuracy	0.88 (0.92)	0.77 (0.62)	0.87 (0.91)	0.88 (0.91)	0.88 (**0.95**)	0.87 (0.94)	0.88 (0.93)	0.89 (**0.95**)
		Running time (s)	0.3 (0.2)	0.03 (0.01)	0.09 (0.07)	0.4 (0.3)	2.4 (1.6)	0.1 (0.2)	0.2 (0.3)	0.02 (0.02)
	600	Accuracy	0.87 (0.92)	0.71 (0.67)	0.86 (0.91)	0.89 (0.90)	0.90 (0.94)	0.92 (**0.95**)	0.89 (0.93)	0.92 (0.94)
		Running time (s)	0.2 (0.3)	0.02 (0.01)	0.1 (0.09)	0.5 (0.6)	2.5 (2.4)	0.2 (0.1)	0.2 (0.2)	0.03 (0.02)

(To be continued)

Table 8 Classifiers accuracy and running time for binary classification on frequent sequential patterns discovered by TKS (CM-SPAM).

(Continued)

Dataset	FreqP	Result	BNB	GNB	DT	RF	MLP	SVM	kNN	LR
DS-6	100	Accuracy	0.90 (**0.98**)	0.57 (0.62)	0.90 (0.88)	0.88 (0.95)	0.95 (**0.98**)	0.92 (0.95)	0.90 (0.90)	0.95 (**0.98**)
		Running time (s)	0.3 (0.2)	0.01 (0.009)	0.07 (0.07)	0.2 (0.3)	0.7 (0.5)	0.02 (0.01)	0.2 (0.3)	0.01 (0.02)
	200	Accuracy	**0.96** (0.91)	0.71 (0.68)	**0.96** (0.85)	**0.96** (0.84)	0.91 (0.89)	0.91 (0.85)	0.92 (0.82)	0.92 (0.88)
		Running time (s)	0.3 (0.2)	0.01 (0.01)	0.08 (0.07)	0.4 (0.3)	1.5 (1.8)	0.02 (0.03)	0.3 (0.2)	0.03 (0.02)
	400	Accuracy	**0.91** (0.88)	0.67 (0.71)	0.89 (0.84)	0.89 (0.85)	**0.91** (0.86)	0.89 (0.86)	0.88 (0.83)	0.90 (0.86)
		Running time (s)	0.2 (0.3)	0.02 (0.01)	0.07 (0.08)	0.4 (0.3)	2.8 (4.1)	0.08 (0.09)	0.2 (0.5)	0.02 (0.02)
	600	Accuracy	0.90 (**0.91**)	0.75 (0.76)	0.88 (0.86)	0.88 (0.87)	0.90 (0.89)	0.89 (0.90)	0.87 (0.87)	0.89 (0.90)
		Running time (s)	0.3 (0.2)	0.02 (0.01)	0.08 (0.07)	0.5 (0.4)	5.01 (5.6)	0.2 (0.2)	0.3 (0.2)	0.03 (0.02)
WDS	100	Accuracy	0.78 (0.82)	0.72 (0.75)	0.76 (0.82)	0.78 (**0.83**)	**0.83** (0.82)	0.77 (**0.83**)	0.70 (0.78)	**0.83** (0.82)
		Running time (s)	0.7 (0.7)	0.03 (0.03)	0.1 (0.1)	0.7 (0.7)	11.3 (11.1)	1.4 (0.8)	0.2 (0.2)	0.03 (0.06)
	200	Accuracy	0.72 (0.85)	0.66 (0.71)	0.76 (0.76)	0.77 (0.78)	0.78 (**0.88**)	0.78 (**0.88**)	0.65 (0.80)	0.79 (0.87)
		Running time (s)	0.9 (0.7)	0.07 (0.08)	0.2 (0.2)	1.5 (1.3)	23.8 (17.1)	3.2 (2.9)	0.2 (0.2)	0.08 (0.09)
	400	Accuracy	0.85 (0.84)	0.81 (0.75)	0.87 (0.80)	0.89 (0.82)	**0.91** (0.86)	0.90 (0.86)	0.80 (0.79)	0.88 (0.86)
		Running time (s)	0.8 (0.7)	0.2 (0.2)	0.9 (0.7)	3.4 (3.6)	59.8 (30.5)	13.5 (14.8)	0.2 (0.2)	0.2 (0.2)
	600	Accuracy	0.79 (0.87)	0.64 (0.75)	0.80 (0.87)	0.83 (0.84)	0.84 (0.87)	0.86 (**0.89**)	0.70 (0.81)	0.84 (0.88)
		Running time (s)	1.5 (1.6)	0.5 (0.8)	2.3 (2.8)	8.3 (8.1)	78.6 (71.8)	55.2 (52.6)	0.3 (0.3)	0.5 (0.6)
Average	100	Accuracy	0.891 (88)	0.767 (0.778)	0.905 (0.854)	0.90 (0.879)	0.925 (0.875)	0.916 (0.862)	0.88 (0.845)	**0.929** (0.869)
		Running time (s)	0.28 (0.24)	0.015 (0.017)	0.086 (0.088)	0.39 (0.43)	3.1 (4.8)	0.26 (0.2)	0.2 (0.2)	0.01 (0.02)
	200	Accuracy	0.90 (0.90)	0.79 (0.77)	**0.931** (0.831)	0.92 (0.817)	0.911 (0.901)	0.919 (0.904)	0.89 (0.864)	0.91 (0.911)
		Running time (s)	0.31 (0.3)	0.02 (0.03)	0.09 (0.08)	0.5 (0.4)	4.5 (4.3)	0.04 (0.04)	0.2 (0.2)	0.02 (0.02)
	400	Accuracy	0.925 (0.892)	0.827 (0.782)	0.922 (0.832)	0.929 (0.838)	**0.938** (0.903)	0.936 (0.873)	0.899 (0.859)	**0.938** (0.901)
		Running time (s)	0.3 (0.3)	0.04 (0.04)	0.09 (0.08)	0.4 (0.5)	10.6 (8.4)	2.03 (2.2)	0.2 (0.3)	0.03 (0.02)
	600	Accuracy	0.919 (0.904)	0.822 (0.786)	0.903 (0.836)	0.924 (0.848)	0.925 (0.905)	**0.935** (0.911)	0.88 (0.87)	0.928 (0.912)
		Running time (s)	0.3 (0.3)	0.02 (0.03)	0.3 (0.5)	1.5 (1.5)	14.05 (16.7)	7.9 (7.8)	0.3 (0.2)	0.03 (0.03)
Average		Accuracy	0.908 (0.894)	0.802 (0.781)	0.915 (0.838)	0.918 (0.845)	0.924 (0.896)	**0.926** (0.887)	0.887 (0.859)	**0.926** (0.898)
		Running time (s)	0.3 (0.3)	0.02 (0.02)	0.1 (0.2)	0.6 (0.7)	8.06 (8.5)	2.5 (2.5)	0.2 (0.2)	0.02 (0.023)

Except Ref. [12], no previous study used the MCC metric to evaluate models.

Interestingly, in studies that used multiple datasets, the highest accuracy was achieved on the PolitiFact dataset which contains few real and fake sequences. Here, the highest accuracy for binary classification is achieved on the GossipCop dataset which is much larger in size than PolitiFact. For binary classification, some studies[36, 43, 46, 58, 59] achieved the highest accuracy of 0.99, followed by Refs. [34, 45, 48] with an accuracy of 0.98. Because LR outperformed other classifiers in both types of classification (binary and MC), we have included LR findings for comparison with other classifiers from the literature. LR outperformed the multimodal approaches[13, 17, 60–66] for fake news detection. Some studies such as Refs. [58, 59] used their models for binary classification in the ISOT dataset and MC classification on the LIAR

Table 9 LR binary classification results on TKS's patterns.

Dataset	FreqP	ACC	*R*	*P*	F1	MCC	AUC	AUPRC
DS-1	100	0.88	0.869	0.804	0.878	0.765	0.94	0.93
	200	0.91	0.925	0.908	0.897	0.84	0.96	0.96
	400	0.94	0.919	0.973	0.94	0.879	0.97	0.97
	600	0.95	0.949	0.950	0.948	0.899	0.97	0.98
DS-2	100	0.95	0.949	0.954	0.949	0.899	0.97	0.97
	200	1	1	1	1	1	1	1
	400	1	1	1	1	1	1	1
	600	1	1	1	1	1	1	1
DS-3	100	1	1	1	1	1	1	1
	200	1	1	1	1	1	1	1
	400	0.98	0.969	0.973	0.972	0.969	1	1
	600	0.99	0.991	0.983	0.987	0.975	1	1
DS-4	100	0.95	0.938	0.968	0.932	0.832	0.96	0.97
	200	0.96	0.988	0.933	0.954	0.912	0.98	0.97
	400	0.98	0.967	0.993	0.977	0.950	1	1
	600	0.95	0.947	0.949	0.954	0.901	0.98	0.97
DS-5	100	0.90	0.918	0.89	0.898	0.824	0.95	0.96
	200	0.79	0.788	0.813	0.793	0.635	0.88	0.92
	400	0.89	0.908	0.892	0.888	0.781	0.97	0.97
	600	0.92	0.919	0.918	0.915	0.846	0.98	0.98
DS-6	100	0.95	0.94	0.955	0.949	0.89	0.97	0.97
	200	0.92	0.917	0.919	0.911	0.855	0.98	0.98
	400	0.90	0.896	0.902	0.901	0.793	0.92	0.98
	600	0.89	0.888	0.908	0.893	0.778	0.92	0.96

Table 10 MC classification results (Case 1).

Dataset	Result	BNB	GNB	DT	RF	MLP	SVM	kNN	LR
WDS	Accuracy	0.70	0.56	0.70	0.76	0.74	**0.77**	0.72	**0.77**
	Running time (s)	32.6	129.9	181.4	197.5	1117.2	17 894.1	5.1	213.5

Table 11 MC classification results (Case 2) discovered by TKS (CM-SPAM).

Dataset	FreqP	Result	BNB	GNB	DT	RF	MLP	SVM	kNN	LR
WDS	100	Accuracy	0.68 (0.71)	0.53 (0.58)	0.56 (0.69)	0.62 (**0.74**)	0.68 (0.73)	0.70 (0.72)	0.62 (0.72)	0.67 (0.73)
		Running time (s)	0.9 (1.3)	0.1 (0.1)	0.2 (0.4)	0.8 (1)	20.6 (23.7)	0.8 (1)	0.2 (0.4)	0.1 (0.2)
	200	Accuracy	0.68 (0.750)	0.55 (0.56)	0.68 (0.60)	0.64 (0.67)	0.72 (**0.76**)	0.70 (**0.76**)	0.63 (0.66)	0.71 (**0.76**)
		Running time (s)	0.7 (0.8)	0.3 (0.4)	0.2 (0.2)	1.4 (1.8)	49.9 (56)	5.5 (4.1)	0.2 (0.2)	0.5 (0.4)
	400	Accuracy	0.72 (0.75)	0.62 (0.60)	0.66 (0.66)	0.72 (0.70)	0.74 (0.77)	0.74 (0.74)	0.67 (0.66)	0.72 (**0.78**)
		Running time (s)	0.9 (0.9)	1.03 (1.5)	0.8 (0.6)	3.8 (4.1)	69.9 (74.7)	17.9 (24.8)	0.2 (0.2)	1.3 (1.3)
	600	Accuracy	0.82 (0.77)	0.80 (0.62)	0.84 (0.74)	**0.85** (0.78)	**0.85** (0.78)	**0.85** (0.79)	0.73 (0.64)	**0.85** (0.80)
		Running time (s)	0.9 (0.9)	1.9 (2.1)	1.4 (1.4)	7.2 (7.8)	147.5 (129.7)	63.2 (75.7)	0.2 (0.3)	3.4 (4.04)
Average		Accuracy	0.725 (0.745)	0.63 (0.59)	0.685 (0.672)	0.707 (0.722)	0.747 (0.76)	0.747 (0.757)	0.662 (0.67)	0.737 (**0.767**)
		Running time (s)	0.85 (0.97)	1.1 (1.02)	0.65 (0.65)	3.37 (3.67)	71.9 (71)	21.85 (26.4)	0.2 (0.27)	1.32 (1.48)

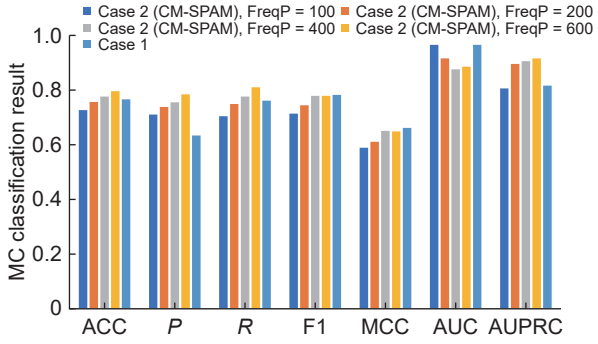


Fig. 4 MC classification results for LR.

dataset. The proposed framework outperformed all the recent approaches[12, 42–45, 56–59, 61–63] (published in the last three years) for binary and MC fake news detection.

RF results (highlighted in bold) for both binary and MC classification of FNACSPM outperformed other classifiers. For MC classification, the majority of the previous studies used the LIAR dataset that has 6 labels. The whole dataset used in this work for MC classification has 12 labels. Interestingly, MC results

Table 12 Comparison of FNACSPM with recent studies for fake news identification.

Classification	Reference	Dataset used	Best learning model	ACC	*P*	*R*	F1	MCC	AUC	AUPRC
Binary	[5]	PHEME	LSTM-RNN	–	0.83	0.84	0.83	–	–	–
	[10]	BuzzFeed, **PolitiFact**	Linear SVM	–	–	–	0.82	–	–	–
	[11]	BuzzFeed, **PolitiFact**	RF	0.89	0.87	0.90	0.89	–	–	–
	[12]	George M., **Kaggle**, GossipCop, PolitiFact	BERT+CNN	0.97	0.96	0.98	0.97	0.94	–	–
	[13]	BuzzFeed, **PolitiFact**	SAFE (Multimodal)	0.87	0.88	0.90	0.89	–	–	–
	[14]	BuzzFeed, **PolitoFact**	DT	0.92	–	–	0.93	–	–	–
	[17]	Twitter, **Weibo**	EANN (Multimodal)	0.82	0.84	0.81	0.82	–	–	–
	[18]	FakeNewsAMT, **Celebirty**	Linear SVM	0.76	–	–	–	–	–	–
	[30]	Combine 5 datasets	HDSF	0.82	–	–	–	–	–	–
	[33]	Buzzfeed	XGB	–	–	–	0.81	–	0.86	–
	[34]	LIAR, **George M.,** self made	RoBERTa	0.98	0.98	0.98	0.98	–	–	–
	[35]	BuzzFeed, **PolitiFact**	Linguistic+SVM	0.84	–	–	–	–	–	–
	[36]	**ISOT Fake News**, 2 Kaggle, George M.	LIWC+RF	0.99	0.99	1	0.99	–	–	–
	[37]	BuzzFeed, Random Political News, **ISOT Fake News**	TF-IDF+DT	0.96	0.96	0.97	0.96	–	–	–
	[38]	GossipCop, **PolitiFact**	SAF	0.69	0.63	0.78	0.70	–	–	–
	[39]	Weibo, self made	TCNN-URG	0.89	–	–	–	–	–	–
	[40]	News Headlines from CNN, Daily Mall	BERT+WCE	–	–	–	0.74	–	–	–
	[41]	GossipCop, **PolitiFact**	Co-attention network	0.90	0.90	0.95	0.92	–	–	–
	[42]	GossipCop, **PolitiFact**	Co-attention network	0.93	0.93	0.97	0.95	–	–	–
	[43]	**ISOT Fake News**, 2 Kaggle	GloVe+BiLSTM	0.99	0.99	0.99	0.99	–	–	–
	[44]	GossipCop, **PolitiFact**	BERT+LSTM	0.88	0.91	0.90	0.90	–	–	–
	[45]	Kaggle	BERT+CNN	0.98	–	–	–	–	–	–
	[46]	Fake and Real News Dataset	BERT-based ensemble	0.99	0.98	0.99	0.99	–	–	–
	[47]	George M.,	Word2Vec+LSTM	0.91	0.89	0.94	0.91	–	–	–
	[48]	Kaggle	GloVe+CNN	0.98	0.99	0.96	0.98	–	–	–
	[58]	ISOT Fake News	CNN-ML	0.99	–	–		–	–	–
	[59]	ISOT Fake News	Static+Capsule neural net	0.99	–	–	–	–	–	–
	[60]	Buzzfeed, **Politifact**	TriFN (MultiModal)	0.87	0.86	0.89	0.88	–	–	–
	[61]	GossipCop, **Weibo**	TRIMOON (MultiModal)	0.91	0.92	0.88	0.90	–	–	–
	[62]	Twitter, Weibo A, Weibo B, **Weibo C**	MCN+CARN (MultiModal)	0.92	0.92	0.92	0.92	–	–	–
	[63]	GossipCop, **PolitiFact**	BERT+CapsNet (MultiModal)	0.93	0.92	0.91	0.92	–	–	–
	[64]	GossipCop, **PolitiFact**	SceneFND (Multimodal)	0.83	0.84	0.84	0.83	–	–	–
	[65]	**Twitter**, Weibo	MPFN (MultiModal)	0.88	0.82	0.82	0.81	–	–	–
	[66]	**Twitter15**, Twitter16	GCAN (MultiModal)	0.86	0.79	0.79	0.79	–	–	–
	FNACSPM (LR)	George M., **GossipCop**, PolitiFact, Buzzeed, Fake News Classification, Fake and Real News classification, all combined	LR	**1**	**1**	**1**	**1**	**1**	**1**	**1**

(To be continued)

Table 12 Comparison of FNACSPM with recent studies for fake news identification.

(Continued)

Classification	Reference	Dataset used	Best learning model	ACC	*P*	*R*	F1	MCC	AUC	AUPRC
MC	[5]	PHEME	LSTM-RNN	–	–	–	0.79	–	–	–
	[49]	LIAR, 6 classes	Hybrid CNN	0.27	–	–	–	–	–	–
	[50]	LIAR, 6 classes	MMDF	0.34	–	–	–	–	–	–
	[51]	PolitiFact, 6 classes	LIWC+LSTM	–	–	–	0.22	–	–	–
	[52]	LIAR, 6 classes	DT	0.39	–	–	–	–	–	–
	[53]	CT-FAN-21, 4 classes	RoBERTa	0.47	0.36	0.34	0.29	–	–	–
	[54]	LIAR, 6 classes	BiLSTM	0.41	–	–	–	–	–	–
	[55]	LIAR, 6 classes	BERT	0.41	–	–	–	–	–	–
	[56]	LIAR, 6 classes	BERT+CNN-BiLSTM	0.47	–	–	–	–	–	–
	[57]	LIAR, 6 classes	AC-BiLSTM	0.33	–	–	0.36	–	–	–
	[58]	LIAR, 6 classes	Static CNN-ML	0.41	–	–	–	–	–	–
	[59]	LIAR, 6 classes	Non-static+capsule neural net	0.40	–	–	–	–	–	–
	FNACSPM (LR, Case 1)	Combination of 6 datasets, 12 classes	LR	**0.77**	**0.76**	**0.63**	**0.78**	**0.66**	**0.97**	**0.82**
	FNACSPM (LR, Case 2)	Combination of 6 datasets, 12 classes	LR	**0.80**	**0.81**	**0.78**	**0.78**	**0.65**	**0.89**	**0.92**
	FNACSPM (BNB)	LIAR, 6 classes	BNB	**0.49**	**0.49**	**0.47**	**0.48**	**0.42**	**0.78**	**0.61**

that we obtained with classifiers in Case 1, when all the words in the pre-processed data are considered, are better than the approaches listed in Table 12, except for Ref. [5] that achieved the highest F1.

We also accessed the proposed framework robustness and scalability on LIAR dataset[49] that contains 12 800 short statements, labeled manually, in various contexts from PolitiFact. From this dataset, we took relevant attributes including "statement", "subject", "speaker", "speaker's job", "state", "party affiliation", and "context (venue)". For the LIAR dataset, BNB achieved the highest accuracy of 0.49 on patterns discovered by using TKS. This result for the LIAR dataset shows the superior performance of the proposed framework by outperforming other approaches[49, 50, 52, 54–59] that also used the LIAR dataset for MC classification.

6 Conclusion

A novel SPM-based framework (called FNACSPM) is presented to analyze and classify fake news. Six diverse datasets, and their combination, were used to investigate the effectiveness and generalization ability of FNACSPM. The datasets were first abstracted and algorithms for SPM were then applied on them to discover frequent words, their frequent sequential patterns, and sequential rules. Discovered frequent patterns were then used in the classification process. Eight classifiers were applied and their performance was accessed and compared by using seven metrics. The results suggest that LR performed better than others for binary and MC classification. It was also observed that (1) using all the words (or news) not only provided less accurate results, compared to using frequent patterns, but also took more time and memory, and (2) limited (or short) sequences of news that contain only frequent patterns of words can be used for reliable prediction and classification rather than entire news. Moreover, FNACSPM outperformed the previous fake news classification/approaches. The proposed framework can handle both binary and MC classification tasks, showcasing its versatility and efficacy in distinguishing between fake and genuine news articles across different complexity levels. Additionally, the research has shed light on the linguistic and semantic structures underlying fake news articles through the utilization of frequent patterns.

This study has various limitations: (1) A drawback of using SPM for fake news classification is that it may exclude crucial words that serve as significant differentiators between fake and real news. This occurs when these words have low frequency and are not considered frequent patterns. As a result, this approach may overlook valuable discriminatory features, which

may affect the classification accuracy. (2) The credibility of online datasets used for training and testing may not be reliable and bias in information collection may not be completely eliminated. (3) The interpretability of the extracted frequent sequential patterns and their relationship to the classification decisions may be limited. Understanding the underlying reasons behind the classification results and explaining them to users or stakeholders may be challenging. (4) Patterns and rules discovered by SPM algorithms require validation and verification from experts. The study focused on extracting patterns from static and retrospective datasets, which do not capture the dynamic nature of fake news propagation in real-time. Real-time analysis and detection of emerging fake news may require additional considerations and techniques beyond pattern mining. Moreover, emerging or contrast pattern mining[78] can be used on the datasets to find contrasting frequent patterns of words and using these patterns for analysis and classification.

References

[1] X. Zhou and R. Zafarani, A survey of fake news: Fundamental theories, detection methods, and opportunities, *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–40, 2020.

[2] G. Ruffo, A. Semeraro, A. Giachanou, and P. Rosso, Studying fake news spreading, polarisation dynamics, and manipulation by bots: A tale of networks and language, *Comput. Sci. Rev.*, vol. 47, p. 100531, 2023.

[3] X. Zhang and A. A. Ghorbani, An overview of online fake news: Characterization, detection, and discussion, *Inf. Process. Manag.*, vol. 57, p. 102025, 2020.

[4] C. Kong, G. Luo, L. Tian, and X. Cao, Disseminating authorized content via data analysis in opportunistic social networks, *Big Data Mining and Analytics*, vol. 2, no. 1, pp. 12–24, 2019.

[5] S. A. Alkhodair, S. H. H. Ding, B. C. M. Fung, and J. Liu, Detecting breaking news rumors of emerging topics in social media, *Inf. Process. Manag.*, vol. 57, p. 102018, 2020.

[6] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, Fake news detection on social media: A data mining perspective, arXiv preprint arXiv: 1708.01967, 2017.

[7] T. Buchanan, Why do people spread false information online? The effects of message and viewer characteristics on self-reported likelihood of sharing social media disinformation, *PLoS One*, vol. 15, no. 10, p. e0239666, 2020.

[8] C. Boididou, S. Papadopoulos, Y. Kompatsiaris, S. Schifferes, and N. Newman, Challenges of computational verification in social multimedia, in *Proc. 23rd Int. Conf. World Wide Web*, Seoul, Republic of Korea, 2014, pp. 743–748.

[9] C. Boididou, S. Papadopoulos, M. Zampoglou, L. Apostolidis, O. Papadopoulou, and Y. Kompatsiaris, Detection and visualization of misleading content on Twitter, *Int. J. Multimed. Inf. Retr.*, vol. 7, no. 1, pp. 71–86, 2018.

[10] N. Sitaula, C. K. Mohan, J. Grygiel, X. Zhou, and R. Zafarani, Credibility-based fake news detection, in *Disinformation, Misinformation, and Fake News in Social Media*, K. Shu, S. Wang, D. Lee, and H. Liu, eds. Cham, Switzerland: Springer, 2020, pp. 163–182.

[11] X. Zhou, A. Jain, V. V. Phoha, and R. Zafarani, Fake news early detection: A theory-driven model, *Digit. Threats Res. Pract.*, vol. 1, no. 2, p. 12, 2020.

[12] M. Choudhary, S. S. Chouhan, E. S. Pilli, and S. K. Vipparthi, BerConvoNet: A deep learning framework for fake news classification, *Appl. Soft Comput.*, vol. 110, p. 107614, 2021.

[13] X. Zhou, J. Wu, and R. Zafarani, SAFE: Similarity-aware multi-modal fake news detection, in *Proc. 24th Pacific-Asia Conference, PAKDD 2020*, Singapore, 2020, pp. 354–367.

[14] X. Zhou and R. Zafarani, Network-based fake news detection: A pattern-driven approach, arXiv preprint arXiv: 1906.04210, 2019.

[15] B. Shi and T. Weninger, Discriminative predicate path mining for fact checking in knowledge graphs, *Knowl. Based Syst.*, vol. 104, no. C, pp. 123–133, 2016.

[16] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini, Computational fact checking from knowledge networks, *PLoS One*, vol. 10, no. 6, p. e0128193, 2015.

[17] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, and J. Gao, EANN: Event adversarial neural networks for multi-modal fake news detection, in *Proc. 24th ACM SIGKDD Conf. Knowledge Discovery & Data Mining*, London, UK, 2018, pp. 849–857.

[18] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, Automatic detection of fake news, arXiv preprint arXiv: 1708.07104, 2017.

[19] P. Fournier-Viger, J. C. W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas, A survey of sequential pattern mining, *Data Science and Pattern Recognition*, vol. 1, no. 1, pp. 54–77, 2017.

[20] M. Cheng, X. Jin, Y. Wang, X. Wang, and J. Chen, A sequential pattern mining approach to tourist movement: The case of a mega event, *J. Travel. Res.*, vol. 62, no. 6, pp. 1237–1256, 2023.

[21] M. S. Nawaz, P. Fournier-Viger, M. Aslam, W. Li, Y. He, and X. Niu, Using alignment-free and pattern mining methods for SARS-CoV-2 genome analysis, *Appl. Intell.*, vol. 53, no. 19, pp. 21920–21943, 2023.

[22] M. S. Nawaz, P. Fournier-Viger, Y. He, and Q. Zhang, PSAC-PDB: Analysis and classification of protein structures, *Comput. Biol. Med.*, vol. 158, p. 106814, 2023.

[23] L. Ni, W. Luo, N. Lu, and W. Zhu, Mining the local dependency itemset in a products network, *ACM Trans. Manage. Inf. Syst.*, vol. 11, no. 1, pp. 1–31, 2020.

[24] R. U. Mustafa, M. S. Nawaz, J. Ferzund, M. I. U. Lali, B. Shahzad, and P. Fournier-Viger, Early detection of controversial Urdu speeches from social media, *Data Science and Pattern Recognition*, vol. 1, no. 2, pp. 26–42, 2017.

[25] D. Schweizer, M. Zehnder, H. Wache, H. F. Witschel, D. Zanatta, and M. Rodriguez, Using consumer behavior data to reduce energy consumption in smart homes: Applying

machine learning to save energy without lowering comfort of inhabitants, in *Proc. IEEE 14th Int. Conf. Machine Learning and Applications* (*ICMLA*), Miami, FL, USA, 2015, pp. 1123–1129.

[26] M. S. Nawaz, P. Fournier-Viger, M. Z. Nawaz, G. Chen, and Y. Wu, MalSPM: Metamorphic malware behavior analysis and classification using sequential pattern mining, *Comput. Secur.*, vol. 118, p. 102741, 2022.

[27] M. S. Nawaz, M. Sun, and P. Fournier-Viger, Proof guidance in PVS with sequential pattern mining, in *Proc. FSEN 2019*, Tehran, Iran, 2019, pp. 45–60.

[28] P. Fournier-Viger, T. Gueniche, and V. S. Tseng, Using partially-ordered sequential rules to generate more accurate sequence prediction, in *Proc. 8th Int. Conf. Advanced Data Mining and Applications, ADMA 2012*, Nanjing, China, 2012, pp. 431–442.

[29] S. Feng, R. Banerjee, and Y. Choi, Syntactic stylometry for deception detection, in *Proc. 50th Annual Meeting of the Association for Computational Linguistics, ACL 2012*, Jeju Island, Republic of Korea, 2012, pp. 171–175.

[30] H. Karimi and J. Tang, Learning hierarchical discourse-level structure for fake news detection, in *Proc. 2019 Conf. the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, Minneapolis, MN, USA, 2019, pp. 3432–3442.

[31] V. L. Rubin and T. Lukoianova, Truth and deception at the rhetorical structure level, *J. Assoc. Inf. Sci. Technol.*, vol. 66, no. 5, pp. 905–917, 2015.

[32] B. Horne and S. Adali, This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news, *Proc. Int. AAAI Conf. Web Soc. Medium.*, vol. 11, no. 1, pp. 759–766, 2017.

[33] J. C. S. Reis, A. Correia, F. Murai, A. Veloso, and F. Benevenuto, Supervised learning for fake news detection, *IEEE Intell. Syst.*, vol. 34, no. 2, pp. 76–81, 2019.

[34] J. Y. Khan, M. T. I. Khondaker, S. Afroz, G. Uddin, and A. Iqbal, A benchmark study of machine learning models for online fake news detection, *Mach. Learn. Appl.*, vol. 4, p. 100032, 2021.

[35] G. Gravanis, A. Vakali, K. Diamantaras, and P. Karadais, Behind the cues: A benchmarking study for fake news detection, *Expert Syst. Appl.*, vol. 128, no. C, pp. 201–213, 2019.

[36] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, Fake news detection using machine learning ensemble methods, *Complexity*, vol. 2020, p. 8885861, 2020.

[37] F. A. Ozbay and B. Alatas, Fake news detection within online social media using supervised artificial intelligence algorithms, *Phys. A*: *Stat. Mech. Appl.*, vol. 540, p. 123174, 2020.

[38] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media, *Big Data*, vol. 8, no. 3, pp. 171–188, 2020.

[39] F. Qian, C. Gong, K. Sharma, and Y. Liu, Neural user response generator: Fake news detection with collective user intelligence, in *Proc. 27th Int. Joint Conf. Artificial Intelligence* (*IJCAI-18*), Stockholm, Sweden, 2018, pp. 3834–3840.

[40] H. Jwa, D. Oh, K. Park, J. Kang, and H. Lim, exBAKE: Automatic fake news detection model based on bidirectional encoder representations from transformers (BERT), *Appl. Sci.*, vol. 9, no. 19, p. 4062, 2019.

[41] K. Shu, L. Cui, S. Wang, D. Lee, and H. Liu, dEFEND: Explainable fake news detection, in *Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, Anchorage, AK, USA, 2019, pp. 395–405.

[42] F. Khan, R. Alturki, G. Srivastava, F. Gazzawe, S. T. U. Shah, and S. Mastorakis, Explainable detection of fake news on social media using pyramidal co-attention network, IEEE Trans. Comput. Soc. Syst., doi: 10.1109/TCSS.2022.3207993.

[43] I. K. Sastrawan, I. P. A. Bayupati, and D. M. S. Arsa, Detection of fake news using deep learning CNN–RNN based methods, *ICT Express*, vol. 8, no. 3, pp. 396–408, 2022.

[44] N. Rai, D. Kumar, N. Kaushik, C. Raj, and A. Ali, Fake news classification using transformer based enhanced LSTM and BERT, *Int. J. Cogn. Comput. Eng.*, vol. 3, pp. 98–105, 2022.

[45] R. K. Kaliyar, A. Goswami, and P. Narang, FakeBERT: Fake news detection in social media with a BERT-based deep learning approach, *Multimed. Tools Appl.*, vol. 80, no. 8, pp. 11765–11788, 2021.

[46] S. Y. Lin, Y. C. Kung, and F. Y. Leu, Predictive intelligence in harmful news identification by BERT-based ensemble learning model with text sentiment analysis, *Inf. Process. Manag.*, vol. 59, no. 2, p. 102872, 2022.

[47] S. Deepak and B. Chitturi, Deep neural approach to fake-news identification, *Procedia Comput. Sci.*, vol. 167, pp. 2236–2243, 2020.

[48] R. K. Kaliyar, A. Goswami, P. Narang, and S. Sinha, FNDNet—A deep convolutional neural network for fake news detection, *Cogn. Syst. Res.*, vol. 61, no. C, pp. 32–44, 2020.

[49] W. Y. Wang, "Liar, liar pants on fire": A new benchmark dataset for fake news detection, arXiv preprint arXiv: 1705.00648, 2017.

[50] H. Karimi, P. C. Roy, S. Saba-Sadiya, and J. Tang, Multi-source multi-class fake news detection, in *Proc. 27th Int. Conf. Computational Linguistics* (*COLING*), Santa Fe, NM, USA, 2018, pp. 1546–1557.

[51] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi, Truth of varying shades: Analyzing language in fake news and political fact-checking, in *Proc. 2017 Conf. Empirical Methods in Natural Language Processing* (*EMNLP*), Copenhagen, Denmark, 2017, pp. 2931–2937.

[52] T. Rasool, W. H. Butt, A. Shaukat, and M. U. Akram, Multi-label fake news detection using multi-layered supervised learning, in *Proc. 2019 11th Int. Conf. Computer and Automation Engineering*, Perth, Australia, 2019, pp. 73–77.

[53] M. Arif, A. L. Tonja, I. Ameer, O. Kolesnikova, A. F. Gelbukh, G. Sidorov, and A. G. M. Meque, CIC at CheckThat! 2022: Multi-class and cross-lingual fake news detection, in *Proc. CEUR Workshop*, Bologna, Italy, 2022, pp. 434–443.

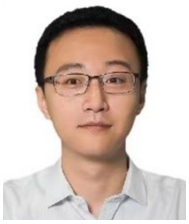
[54] Y. Long, Q. Lu, R. Xiang, M. Li, and C. R. Huang, Fake news detection through multi-perspective speaker profiles, in *Proc. 8th Int. Joint Conf. Natural Language Processing* (*IJCNLP*), Taipei, China, 2017, pp. 252–256.

[55] N. Singh, R. K. Kaliyar, T. Vivekanand, K. Uthkarsh, V. Mishra, and A. Goswami, B-LIAR: A novel model for

handling multiclass fake news data utilizing a transformer encoder stack-based architecture, in *Proc. 1st Int. Conf. Informatics* (*ICI*), Noida, India, 2022, pp. 31–35.
[56] J. Alghamdi, Y. Lin, and S. Luo, Modeling fake news detection using BERT-CNN-BiLSTM architecture, in *Proc. IEEE 5th Int. Conf. Multimedia Information Processing and Retrieval* (*MIPR*), CA, USA, 2022, pp. 354–357.
[57] T. E. Trueman, J. Ashok Kumar, P. Narayanasamy, and J. Vidya, Attention-based C-BiLSTM for fake news detection, *Appl. Soft Comput.*, vol. 110, p. 107600, 2021.
[58] M. H. Goldani, R. Safabakhsh, and S. Momtazi, Convolutional neural network with margin loss for fake news detection, *Inf. Process. Manag.*, vol. 58, no. 1, p. 102418, 2021.
[59] M. H. Goldani, S. Momtazi, and R. Safabakhsh, Detecting fake news with capsule neural networks, *Appl. Soft Comput.*, vol. 101, p. 106991, 2021.
[60] K. Shu, S. Wang, and H. Liu, Beyond news contents: The role of social context for fake news detection, arXiv preprint arXiv: 1712.07709, 2017.
[61] S. Xiong, G. Zhang, V. Batra, L. Xi, L. Shi, and L. Liu, TRIMOON: Two-round inconsistency-based multi-modal fusion network for fake news detection, *Inf. Fusion*, vol. 93, no. C, pp. 150–158, 2023.
[62] C. Song, N. Ning, Y. Zhang, and B. Wu, A multimodal fake news detection model based on crossmodal attention residual and multichannel convolutional neural networks, *Inf. Process. Manag.*, vol. 58, no. 1, p. 102437, 2021.
[63] B. Palani, S. Elango, and V. K. Vignesh, CB-Fake: A multimodal deep learning framework for automatic fake news detection using capsule neural network and BERT, *Multimed. Tools Appl.*, vol. 81, no. 4, pp. 5587–5620, 2022.
[64] G. Zhang, A. Giachanou, and P. Rosso, SceneFND: Multimodal fake news detection by modelling scene context information, *J. Inf. Sci.*, vol. 50, no. 2, pp. 355–367, 2022.
[65] J. Jing, H. Wu, J. Sun, X. Fang, and H. Zhang, Multimodal fake news detection via progressive fusion networks, *Inf. Process. Manag.*, vol. 60, no. 1, p. 103120, 2023.
[66] Y. J. Lu and C. T. Li, GCAN: Graph-aware co-attention networks for explainable fake news detection on social media, in *Proc. 58th Annual Meeting of the Association for Computational Linguistics*, Virtual Event, 2020, pp. 504–514.
[67] G. McIntire, Fake Real News Dataset, https://github.com/GeorgeMcIntire/fake_real_news_dataset, 2024.
[68] Kaggle, BuzzFeed News Analysis and Classification, http://kaggle.com/code/sohamohajeri/buzzfeed-news-analysis-and-classification/, 2024.
[69] Kaggle, Fake News Classification, http://kaggle.com/datasets/saurabhshahane/fake-news-classification, 2024.
[70] Kaggle, Fake and Real News Dataset, http://github.com/MuhammadzohaibNawaz/FakeNewDS6, 2024.
[71] M. S. Nawaz, P. Fournier-Viger, A. Shojaee, and H. Fujita, Using artificial intelligence techniques for COVID-19 genome analysis, *Appl. Intell.*, vol. 51, no. 5, pp. 3086–3103, 2021.
[72] R. Agrawal and R. Srikant, Fast algorithms for mining association rules in large databases, in *Proc. 20th VLDB*, Santiago, Chile, 1994, pp. 487–499.
[73] P. Fournier-Viger, A. Gomariz, T. Gueniche, E. Mwamikazi, and R. Thomas, TKS: Efficient mining of top-*k* sequential patterns, in *Proc. 9th Int. Conf. Advanced Data Mining and Applications* (*ADMA*), Hangzhou, China, 2013, pp. 109–120.
[74] P. Fournier-Viger, A. Gomariz, M. Campos, and R. Thomas, Fast vertical mining of sequential patterns using co-occurrence information, in *Advances in Knowledge Discovery and Data*, V. S. Tseng, T. B. Ho, Z. H. Zhou, A. L. P. Chen, and H. Y. Kao, eds. Cham, Switzerland: Springer, 2014, pp. 40–52.
[75] P. Fournier-Viger, T. Gueniche, S. Zida, and V. S. Tseng, ERMiner: Sequential rule mining using equivalence classes, in *Advances in Intelligent Data Analysis XIII*, H. Blockeel, M. van Leeuwen, and V. Vinciotti, eds. Cham, Switzerland: Springer, 2014, pp. 108–119.
[76] P. Fournier-Viger, J. C. W. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z. Deng, and H. T. Lam, The SPMF open-source data mining library version 2, in *Machine Learning and Knowledge Discovery in Databases*, B. Berendt, B. Bringmann, É. Fromont, G. Garriga, P. Miettinen, N. Tatti, and V. Tresp, eds. Cham, Switzerland: Springer, 2016, pp. 36–40.
[77] O. Kramer, Scikit-learn, in *Machine Learning for Evolution Strategies*, O. Kramer, ed. Cham, Switzerland: Springer, 2016, pp. 45–53.
[78] S. Ventura and J. M. Luna, *Supervised Descriptive Pattern Mining*. Berlin, Germany: Springer, 2018.

M. Saqib Nawaz received the BS degree in computer systems engineering from University of Engineering and Technology, Peshawar, Pakistan in 2011, the MS degree in computer science from University of Sargodha, Pakistan in 2014, and the PhD degree from Peking University, Beijing, China in 2019. He worked as a postdoctoral fellow at Harbin Institute of Technology (Shenzhen), China from September 2019 to January 2022. He is currently working as an associate researcher at Shenzhen University, China. His research interests include bioinformatics, pattern mining, formal methods, and the use of machine learning and data mining in software engineering.

M. Zohaib Nawaz received the bachelor degree from University of Sargodha, Pakistan in 2016 and the master degree from National University of Sciences and Technology (NUST), Pakistan in 2020. He is a lecturer (on leave) at the Department of Computer Science, Faculty of Computing and Information Technology, University of Sargodha, Pakistan since 2018. Currently, he is pursuing the PhD degree in computer science at Shenzhen University, China. His research interests include descriptive pattern mining and formal methods. He is a member of ACM.



Yulin He received the PhD degree from Hebei University, China in 2014. From 2011 to 2014, he served as a research assistant at the Department of Computing, Hong Kong Polytechnic University, Hong Kong, China. From 2014 to 2017, he worked as a postdoctoral fellow at the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He is currently a research fellow at the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China. His main research interests include big data approximate computing technologies, multi-sample statistical analysis theories and methods, and data mining/machine learning algorithms and their applications. He has published over 150+ research papers in ACM, CAAI, IEEE, Elsevier, Springer journals and international conferences. He is an ACM member, CAAI member, CCF member, IEEE member, and the editorial review board members of several international journals.

Philippe Fournier-Viger is a distinguished professor at Shenzhen University, China. He has published more than 330 research papers related to data mining, intelligent systems, and applications, which have received more than 14 000 citations (H-Index 60). He was the associate editor-in-chief of the *Applied Intelligence* journal and editor-in-chief of *Data Science and Pattern Recognition*. He is the founder of the popular SPMF data mining library. He is a co-founder of the UDML, PMDB, and MLiSE series workshop at the ICDM, PKDD, DASFAA, and KDD conferences. His interests are data mining, algorithm design, pattern mining, sequence mining, big data, and applications.