Applied Soft Computing Journal 151 (2024) 111173

Contents lists available at ScienceDirect

Applied Soft Computing

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

Using fuzzy transform for sustainable fake news detection

Tayasan Milinda H. Gedara [a,b], Vincenzo Loia [a], Stefania Tomasiello [c,d,*]

[a] Department of Management & Innovation Systems, University of Salerno, Via Giovanni Paolo II, 132, Fisciano, 84084, Salerno, Italy
[b] IMT School for Advanced Studies, Piazza S. Ponziano, 6, 55100, Lucca, Italy
[c] Department of Industrial Engineering, University of Salerno, Via Giovanni Paolo II, 132, 84084, Fisciano, Italy
[d] Institute of Computer Science, University of Tartu, Narva mnt 18, 50090, Tartu, Estonia

ARTICLE INFO

Keywords:
Misinformation
Deep learning
LSTM
F-transform
Energy

ABSTRACT

The proliferation of fake news has raised concerns regarding its detection, posing a significant challenge. Motivated by the ongoing discussion on the sustainability of machine learning algorithms, this paper discusses the usefulness of data reduction for fake news detection. This is accomplished by using the fuzzy transform (or F-transform for short), which has already been proven effective, in the literature, to reduce the training time. A Long Short Term Memory architecture is then employed for classification to determine the authenticity of the news. From the formal perspective, we discuss in general the role of the F-transform in a learning system. Regarding the numerical experiments, we use five publicly available datasets, trying different compression ratios and different types of F-transform, assessing accuracy, F1 score, training time and energy consumption with and without F-transform. Although the F-transform is a lossy compression technique, the results show a negligible variation in accuracy and F1-score when comparing results with and without F-transform (i.e. 1%–3% in most cases and around 10% in one case). This seems to be congruent with the theoretical achievement. Furthermore, the approach yields substantial training time and energy savings, with over 50% reduction in energy consumption.

1. Introduction

In the digital era, information drives the globe, and individuals share news and comments on digital platforms like Facebook and Twitter. As a result of the enormous expansion and development of social media services, there has been a vast diffusion of digital information. Over 4.89 billion people used social media globally in 2023, with a projected growth of 5.85 billion in 2027 [1]. Social media has become a straightforward way for people to consume online information on a regular basis due to its intrinsic characteristics of fast transfer, easy access, and ease of use. The widespread availability of smart devices and mobile equipment has enabled people to read news, discuss thoughts, and submit comments at any time and from any location. As a result, the growing popularity of social media platforms has altered how people interact with each other, changing how information moves on the Internet, and having far-reaching consequences in people's daily lives. The rapid growth of social media services has created a risk in cyberspace related to the spread of fake news and misinformation. This can cause panic, negative impact, and, in the worst-case scenario, some violence among the community, posing a serious threat to cybersecurity as well as national security [2]. For example, during the 2016 presidential election in the United States, fake news about the two candidates was shared more than 37 million times on Facebook [3]. Furthermore, during the COVID-19 outbreak, there was a lot of bogus news about this pandemic on social media, influencing people's activities [4].

In the literature, the term "fake news" is sometimes described as "misinformation", "disinformation", "hoax", and "rumour", all of which are variations of misleading information. There are numerous research projects, methods, and applications for fact-checking and fake news identification [5], the majority of which tackle the problem as a veracity classification. Several classifiers, including Naive Bayes, Decision Trees, Support Vector Machine, have been tried, but not leading to high accuracy [6,7]. Deep learning (DL) models, such as Long Short-Term Memory (LSTM) have been widely used to improve accuracy (e.g. see [6,8,9]). Anyhow, DL algorithms are attracting criticism since they are costly to train both from the computational and energy standpoint. It has been shown that training and tuning a deep architecture may produce the same amount of carbon dioxide as five cars during their lifespan [10].

Armed with such a vision, we propose to use F-transform to reduce the dataset size, saving training time and energy. The F-transform has been successfully used for data compression [11] and to speed up machine learning (ML) algorithms (e.g. see [12]). We formally

* Corresponding author at: Institute of Computer Science, University of Tartu, Narva mnt 18, 50090, Tartu, Estonia.
E-mail addresses: stomasiello@unisa.it, stefania.tomasiello@ut.ee (S. Tomasiello).

https://doi.org/10.1016/j.asoc.2023.111173
Received 7 August 2023; Received in revised form 2 November 2023; Accepted 15 December 2023
Available online 23 December 2023

discuss its role in a learning system, i.e. how the F-transform-based compression may affect the error. We apply the F-transform to reduce the cardinality of the data presented to an LSTM. LSTM is probably the most popular deep architecture for fake news detection (e.g. see [6, 13]). To the best of our knowledge, this is the first time the F-transform has been tried jointly with LSTM for fake news detection. The numerical experiments were performed on five publicly available datasets, namely GossipCop, ISOT, Fake News Inference Dataset (FNID), CoVID-19-FNIR, and PolitiFact. The results showed the effectiveness of the proposed approach.

The article is structured as follows. In the next section, we briefly discuss the related works. In Section 3, the methodology is presented and some properties are discussed. Section 4 is devoted to the numerical experiments. In the last section, some conclusions are drawn.

2. Related works

Various ML methodologies have been proposed in recent years to detect bogus news that travels over social media [6]. The main DL techniques employed to this end fall in the broad class of RNNs (including LSTM) and Convolutional Neural Networks (CNNs), without excluding a combination of the two, although the best results seem to be achieved by the simplest LSTM models [6]. Actually, many papers proposing LSTM and variants of it for fake news detection have been appearing over the last five-year period.

Cui et al. [14] suggested an explainable fake news detection system (dEFEND) to determine and detect false news using a BiLSTM model. The two different datasets employed by the authors were GossipCop and PolitiFact. The experimental findings showed that the detection performance was excellent, although only two datasets were used. Sadeghi et al. [15] proposed natural language processing and DL to perform experiments with the FNID dataset [16]. Here the biLSTM model achieved the highest accuracy among the test cases. Zhang et al. [13] focused on DL-based fast fake news detection in Chinese short texts by comparing the training time of each model. LSTM-based models turned out to have higher accuracy and reasonable training time with respect to some other models. A combination of biLSTM and CNN was proposed in [17], where the numerical experiments were successfully performed on the LIAR dataset. Mohamed et al. [8] introduced fuzzy logic combined with LSTM to detect fake news in Covid19 data. In their approach, fuzzy sentiment scores were used to encode tokenized text data and convert it into training data for the LSTM model. A better performance of the LSTM model was observed, although there was not a significant improvement in accuracy using fuzzy sentiment (0.914 against 0.902 without it). Covid19 data was also used in [18] along with an LSTM-based approach. Dixit et al. [19] also proposed an LSTM-based model but with a variant of Principal Component Analysis (PCA), called Probability Principal Component Analysis (PPCA), for data reduction. The authors performed the numerical experiments with the BuzzFeed, GossipCop, PolitiFact, and ISOT datasets. Chen et al. [9] proposed another approach for feature extraction using fuzzy logic, adopting the LSTM model and Covid19 dataset. In this approach, the fuzzy C-means clustering method was employed for feature extraction before training the model. The fuzzy clustering method helped to increase the accuracy and, additionally, to reduce the training time. According to the best results reported in the article, the training time was reduced by 30%, but at the same time, accuracy dropped by 13%–15%. In [20], the LSTM was employed again, this time with a feature selection technique based on a metaheuristic called successive position-based barnacles mating optimization. The experiments were performed on only one publicly available dataset.

Lately, the application of Graph Neural Networks (GNNs) to fake news detection has attracted a certain interest [21]. In this regard, among the papers listed in the survey [21], it is worth mentioning [22], where the experiments were performed on the same datasets used in the already mentioned [14], i.e. PolitiFact and GossipCop, adopting the same number of training and testing samples as in [14]. The authors found as best accuracy 0.811 and 0.853 for the two above-mentioned datasets, respectively.

Regarding other approaches, Shu et al. [23] suggested an unsupervised technique based on a hierarchical propagation network for detecting false information. The performance measures, using the GossipCop and PolitiFact datasets, were accuracy, precision, recall, and F1-score. The experiments showed that the proposed approach was robust enough, with an accuracy of 0.843 for PolitiFact and 0.861 for GossipCop. Sharma et al. [24] proposed a novel ensemble strategy based on text embedding, with fuzzy evolutionary logic at the classification layer. This approach was based on Word2Vec, GloVe, and BERT models. It was validated using the Headlines, Self-Annotated Reddit Corpus (SARC), and Twitter datasets.

By looking at the prior research works in the fake news context, it is possible to observe that

- they mostly focused on the model accuracy, without minding the training time;
- data reduction was seldom applied, and, in any case, without a proper discussion, either formal or numerical on training time and energy consumption.

3. Methodology

In this section, we present the proposed approach. The pipeline is shown in Fig. 1. The first stage is preprocessing data, typically removing punctuation and misplaced symbols. Afterwards, word encoding is used to convert the documents into sequences of numeric indices using word vectors [25]. This stage produces a numerical matrix whose higher dimension is reduced by the F-transform, which is applied only in one direction (typically to reduce the number of rows). Then a conventional LSTM model is trained for the binary classification problem of fake news detection.

3.1. Data preprocessing

Data preprocessing is a particular operation to clean the data and to rearrange it in a more convenient way to be processed. It can be regarded as the most essential phase employed to enhance the efficiency of the results [26]. It consists of several stages, as described in the following.

- Cleaning. Stop words are eliminated. Stop words are used for concluding sentences appropriately and linking different phrases. They are usually irrelevant, and lacking informative content.
- Lower-case letter conversion. During the process, numerical values are substituted with 0 characters and uppercase letters are represented as lower-case letters.
- Tokenization. It is the process of breaking down the original text into smaller parts called tokens while removing punctuation. Numerical values are also removed, as well as words with fewer characters [27].

3.2. Fuzzy transform: data compression and learnability

We start off this section by recalling some definitions. Let $I = [x_1, x_n]$ be a closed interval and $x_1 < x_2 < \cdots < x_n$ n points of I (nodes), with $n > 3$. A fuzzy partition of I is the sequence $\{A_j\}_{j=1,\dots,n}$ of normal continuous fuzzy sets $A_i : I \to [0,1]$, satisfying the Ruspini's condition $\sum_{i=1}^{n} A_i(x) = 1, \quad \forall x \epsilon I$. These fuzzy sets are also called basic functions, and they form a uniform fuzzy partition when the nodes are equidistant.

Common basic functions are:

- hat functions

$$A_j(x) = \begin{cases} (x_{j+1} - x)/(x_{j+1} - x_j), & x \epsilon [x_j, x_{j+1}] \\ (x - x_{j-1})/(x_j - x_{j-1}), & x \epsilon [x_{j-1}, x_j] \\ 0, & otherwise, \end{cases} \quad (1)$$

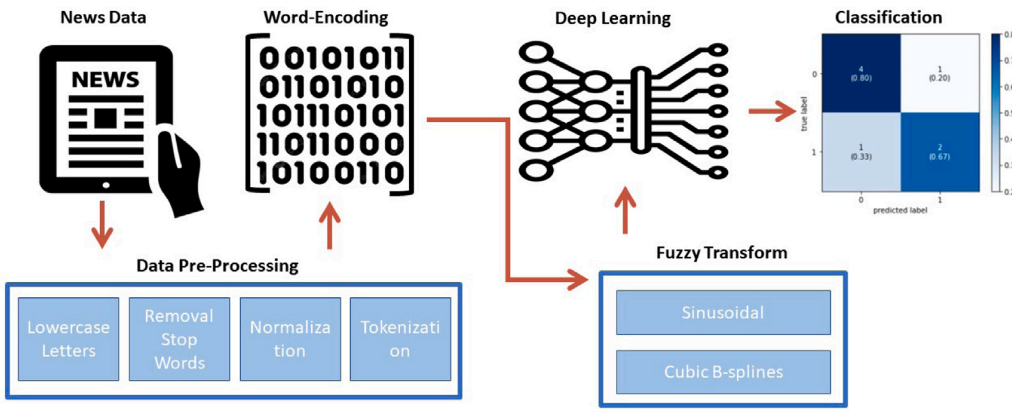

Fig. 1. Stages of the proposed approach.

- sinusoidal-shaped basic functions

$$A_j(x) = \begin{cases} \frac{1}{2}\left(\cos(\pi\frac{x-x_j}{x_{j+1}-x_j})+1\right), & x\epsilon[x_j,x_{j+1}] \\ \frac{1}{2}\left(\cos(\pi\frac{x-x_j}{x_j-x_{j-1}})+1\right), & x\epsilon[x_{j-1},x_j] \\ 0, & otherwise. \end{cases} \tag{2}$$

A fuzzy partition with small support has the additional property that there exists $r \geq 1$ such that $supp(A_i) = cl\{x\epsilon I\ :\ A_i(x) > 0\} \subseteq [x_i, x_{i+r}]$, where cl stands for closure. A good choice for such kind of fuzzy partition is represented by cubic B-splines:

$$A_j(x) = \frac{1}{h^3}\begin{cases} (x-x_{j-2})^3, & x\epsilon[x_{j-2},x_{j-1}) \\ (x-x_{j-2})^3 - 4(x-x_{j-1})^3, & x\epsilon[x_{j-1},x_j) \\ (x_{j+2}-x)^3 - 4(x_{j+1}-x)^3, & x\epsilon[x_j,x_{j+1}) \\ (x_{j+2}-x)^3, & x\epsilon[x_{j+1},x_{j+2}) \\ 0, & otherwise, \end{cases} \tag{3}$$

where $h = max|x_{j+1} - x_j|$ is the norm of the partition, and $j = 0,\dots,n$. It must be mentioned that for cubic B-splines two auxiliary points, both on the left and on the right of the considered interval, are required.

Let us consider the $N \times M$ data matrix $\mathbf{D}$ and the fuzzy partitions $\{A_1,\dots,A_n\}$ and $\{B_1,\dots,B_m\}$ of the intervals $[1,N]$ and $[1,M]$ respectively, with $n < N$ and $m < M$. Let $\mathbf{A}$ and $\mathbf{B}$ be the matrices with entries $A_k(i)$ and $B_l(j)$, respectively.

The discrete F-transform matrix of $\mathbf{D}$ with respect to the fuzzy partitions $\{A_1,\dots,A_n\}$ and $\{B_1,\dots,B_m\}$ is given by

$$\mathbf{F} = \mathbf{P}\circ\mathbf{Q}, \tag{4}$$

where $\circ$ represents the Hadamard product, $\mathbf{P} = \mathbf{A}^T\mathbf{D}\mathbf{B}$ and $\mathbf{Q}$ is the matrix whose entries are the inverse of the entries of the matrix $\overline{\mathbf{Q}} = \mathbf{A}^T\mathbf{I}_{NM}\mathbf{B}$, with $\mathbf{I}_{NM}$ being the $N \times M$ matrix with all unit entries.

This is a particular case of the so-called discrete F^1-transform, also denoted as F^0-transform. The F^1-transform of the $N \times M$ data matrix $\mathbf{D}$ is the $n \times m$ matrix, whose entries are computed as follows [11]:

$$F^1_{kl}(i,j) = c^{00}_{kl} + c^{10}_{kl}(i-k) + c^{01}_{kl}(j-l), \tag{5}$$

where

$$c^{00}_{kl} = F^0_{kl} = \frac{\sum_{j=1}^M\sum_{i=1}^N D_{ij}A_k(i)B_l(j)}{\sum_{j=1}^M\sum_{i=1}^N A_k(i)B_l(j)}, \tag{6}$$

$$c^{10}_{kl} = \frac{\sum_{j=1}^M\sum_{i=1}^N D_{ij}A_k(i)(i-k)B_l(j)}{\sum_{j=1}^M\sum_{i=1}^N A_k(i)(i-k)^2B_l(j)}, \tag{7}$$

$$c^{01}_{kl} = \frac{\sum_{j=1}^M\sum_{i=1}^N D_{ij}A_k(i)B_l(j)(j-l)}{\sum_{j=1}^M\sum_{i=1}^N A_k(i)B_l(j)(j-l)^2}. \tag{8}$$

The inverse discrete F^1-transform is

$$D^{F,1}_{ij} = \sum_{k=1}^m\sum_{l=1}^n F^1_{kl}A_k(i)B_l(j). \tag{9}$$

To support a further discussion, it is useful to recall that the F-transform of an N-sized vector $\mathbf{v}$ is the n-sized row vector (with $n < N$)

$$\mathbf{f}^T = \mathbf{v}^T\mathbf{A}\mathbf{S}^{-1}, \tag{10}$$

where $\mathbf{S}$ is the diagonal matrix with entries $S_{jj} = \sum_{i=1}^N A_{ji}$.

It must be mentioned that, because of the basic functions, the matrices $\mathbf{A}$ and $\mathbf{B}$ turn out to be quasi-band matrices, i.e. rectangular matrices exhibiting a bank-like structure.

In [28], a block approach involving the F-transform was proposed. In this approach, the $N \times M$ data matrix $\mathbf{D}$ is divided into submatrices $\mathbf{D}_T$ with dimension $N(T) \times M(T)$, each one compressed to a block $\mathbf{F}_T$ of size $n(T) \times m(T)$ by means of the discrete F-transform.

Some studies tackled the problem of compressed data with a generic learning system [29,30]. A learning system offers a map from a training set to a set of hypotheses, from which the computed output is the one which minimizes a loss function. Hence, the outcome is the approximation $y = f(\mathbf{w}^T\mathbf{d})$, given a dataset $\{y_i,\mathbf{d}_i\}_{i=1,\dots,N}$, with $\mathbf{d}_i \in \mathbb{R}^M$. The parameters collected in the M-sized vector $\mathbf{w}$ are usually called weights. Using compressed data with a learning system means replacing the original training dataset with one which has lower cardinality.

The main result in the literature, valid for any compression scheme, is reported in the following. Let e_c be the error coming from compressed data with cardinality k in a learning system, and $K > k$ the cardinality of the original data. Moreover, let P denote an arbitrary though fixed probability distribution over the data domain.

Theorem 1 ([29,30]). *For any $\varepsilon > 0$ and any compression scheme, the following holds:*

$$P(e_c > \varepsilon) < \binom{K}{k}(1-\varepsilon)^{K-k}. \tag{11}$$

In the following, we formally discuss the order of the error due to the F-transform-based compression with a learning system.

For the remainder of this work, $\|.\|$ will denote the induced 2-norm. It is useful to recall that for any $N \times n$ matrix $\mathbf{M}$, the max norm is defined as $\|\mathbf{M}\|_{max} = \max_{ij}|m_{ij}|$, while the infinity norm is $\|\mathbf{M}\|_\infty = \max_i\sum_{j=1}^n|m_{ij}|$. Let $e_F = \|\overline{\mathbf{y}}^T - \mathbf{y}_F^T\| = \|\overline{\mathbf{y}}^T - f(\mathbf{w}_F^T\mathbf{F}^T)\|$ and $e_D = \|\mathbf{y}^T - \mathbf{y}_D^T\| = \|\mathbf{y}^T - f(\mathbf{w}_D^T\mathbf{D}^T)\|$ be the distance between the targets and the computed outputs by using the F-transform compressed data and the original data $\mathbf{D} \in \mathbb{R}^{N\times M}$, respectively. In what follows, we consider the compression over the rows, i.e. $\mathbf{F} \in \mathbb{R}^{n\times M}$, implying $\mathbf{B} = \mathbf{I}$.

Theorem 2. *Suppose $\|\mathbf{S}^{-1}\| \leq \frac{1}{\sqrt{N}}$ and $\|\mathbf{w}_F\| \leq \frac{\|\mathbf{w}_D\|}{\sqrt{nM}}$. If the map $f(.)$ is Lipschitz continuous and $f(0) = 0$, then*

$$e_F = O(e_D). \tag{12}$$

Proof. First, let us note that

$\|\mathbf{A}\| \leq \sqrt{N}\|\mathbf{A}\|_{\infty}$,

with $\|\mathbf{A}\|_{\infty} = 1$ because of the Ruspini's condition, and besides,

$\|\mathbf{Q}\| \leq \sqrt{nM}\|\mathbf{Q}\|_{max}$,

$\|\mathbf{Q}\|_{max} = \|\mathbf{S}^{-1}\| = \sigma_{max}(\mathbf{S}^{-1}) = \max_j \frac{1}{\sum_{i=1}^{N} A_{ji}}$.

All this considered, it is possible to write

$$e_F \leq \|\overline{\mathbf{y}}^T - f(\mathbf{0}^T)\| + \|f(\mathbf{0}^T) - f(\mathbf{w}_F^T\mathbf{F}^T)\| \leq \|\mathbf{y}^T\mathbf{A}\mathbf{S}^{-1}\| + L\|\mathbf{w}_F^T\mathbf{D}^T\mathbf{A}\circ\mathbf{Q}^T\| \leq$$
$$\leq \|\mathbf{y}^T\|\|\mathbf{A}\|\|\mathbf{S}^{-1}\| + L\|\mathbf{w}_F^T\|\|\mathbf{D}^T\|\|\mathbf{A}\|\|\mathbf{Q}^T\| \leq$$
$$\leq \sqrt{N}\|\mathbf{S}^{-1}\|(\|\mathbf{y}^T\| + L\sqrt{nM}\|\mathbf{w}_F\|\|\mathbf{D}^T\|).$$

Because of the hypotheses, then

$$e_F \leq \|\mathbf{y}^T\| + L\|\mathbf{w}_D\|\|\mathbf{D}^T\|$$

and since the same upper bound holds for e_D, i.e. $e_F/e_D \leq 1$, then the conclusion follows. □

Theorem 2 formally states that, under some conditions, the order of the error using compressed data is the same as obtained with original data. The above-mentioned conditions are not inconsistent with practical application. In fact, $\|\mathbf{S}^{-1}\| \leq \frac{1}{\sqrt{N}}$ means $\min_j \sum_{i=1}^{N} A_{ji} \geq \sqrt{N}$, and, recalling that $A_{ij} \leq 1$ for any i, j, this poses a condition on the band size of the matrix $\mathbf{A}$.

A typical function which is Lipschitz continuous with $f(0) = 0$ is $tanh$. In the case of Lipschitz continuous functions with $f(0) \neq 0$ (e.g. the sigmoid), then $e_D - e_F \leq (\sqrt{N} - \sqrt{n})f(0)$.

3.3. LSTM model

LSTM is a deep-in-time recurrent architecture. It is a well-known model, hence we recall it very briefly. The LSTM strengthens long-term dependencies to improve the memory function of classical Recurrent Neural Networks (RNNs) [31]. To this end, the model includes forget, input, and output gates. More in detail

- the input gate, determines whether the current input has to be added to the model's long-term memory;
- the output gate, determines whether the current input should be used as the feature training model's output;
- the forget gate, evaluates the correlation between the current input and the model's long-term memory and determines whether to add the current input to the long-term memory or remove it.

The model uses the cell state to determine if the long-term memory should be updated. The forward propagation of the given input $\mathbf{x}_t$ at the time step t in the LSTM can be formalized as follows:

- forget gate activation

  $\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{V}_f\mathbf{y}_{t-1} + \mathbf{b}_f)$,
- update gate activation

  $\mathbf{u}_t = \sigma(\mathbf{W}_u\mathbf{x}_t + \mathbf{V}_u\mathbf{y}_{t-1} + \mathbf{b}_u)$,
- candidate cell state

  $\overline{\mathbf{c}}_t = \overline{\sigma}(\mathbf{W}_c\mathbf{x}_t + \mathbf{V}_c\mathbf{y}_{t-1} + \mathbf{b}_c)$,
- output gate activation

  $\mathbf{z}_t = \sigma(\mathbf{W}_z\mathbf{x}_t + \mathbf{V}_z\mathbf{y}_{t-1} + \mathbf{b}_z)$,
- new cell state

  $\mathbf{c}_t = \mathbf{f}_t\circ\mathbf{c}_{t-1} + \mathbf{u}_t\circ\overline{\mathbf{c}}_t$,
- final output

  $\mathbf{y}_t = \mathbf{z}_t\circ\overline{\sigma}(\mathbf{c}_t)$,

where σ and $\overline{\sigma}$ are the sigmoid and tanh activation functions respectively, $\mathbf{W}_*$, $\mathbf{V}_*$ are weight matrices and $\mathbf{b}_*$ bias vectors, $* \in \{f, u, c, z\}$.

Table 1
LSTM parameters.

Feature/Parameter	Values
Batch Size	16/64/128
Optimiser	Adam
Activation Function	Sigmoid
Epochs	20
GradientThreshold	1
InitialLearnRate	0.005
LearnRateDropFactor	0.2

4. Numerical experiments

This section is devoted to presenting the experimental stage of the proposed approach for detecting fake news. Various experiments were performed using five different datasets, namely, GossipCop, ISOT, FNID, CoVID19-FNIR, and PolitiFact. Finally, a comparative analysis was conducted in order to check the method's effectiveness. The numerical experiments for the proposed approach were carried out using MATLAB 2022a, with a 12th gen Intel(R) Core(TM) i7-1260P 2.10 GHz CPU. To simulate the energy consumption, we used Intel Power Gadget 3.6. The LSTM model features and parameters are shown in Table 1. The parameters were fixed by trial and error, starting with the common settings in the relevant literature. The batch size was adjusted according to the accuracy.

4.1. Performance measures

The proposed approach's performance has been assessed using accuracy and F1 score:

$$Accuracy = \frac{|T_N| + |T_P|}{|T_N| + |T_P| + |F_N| + |F_P|} \tag{13}$$

$$F1 = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \tag{14}$$

where

$$Precision = \frac{|T_P|}{|T_P| + |F_P|} \tag{15}$$

$$Recall = \frac{|T_P|}{|T_P| + |F_N|} \tag{16}$$

and

- $|T_P|$ stands for true positive, meaning that the predicted news is really fake, i.e. counterfeit;
- $|T_N|$ denotes the true negative, implying that the predicted news is practically authentic;
- $|F_P|$ represents false positive, i.e. the predicted fake news is instead authentic;
- $|F_N|$ is false negative, i.e. the predicted real news is instead fake.

To compute the time saving using the F-transform, we use the following

$$TimeSaving = \frac{Time - Time_F}{Time} * 100, \tag{17}$$

where $Time$ and $Time_F$ denote the training time without and with F-transform. A similar formula can be adopted to calculate the energy saving, by replacing the time with the energy consumption.

Table 2
Training and testing data.

Datasets	Total data	Training data	Validation data	Testing data
ISOT	44,897	35,919	6,286	2,693
PolitiFact	7,819	6,254	1,095	468
GossipCop	32,785	26,229	4,590	1,967
FNID	32,535	26,029	4,555	1,952
Covid-19	9,075	7261	1,271	544

4.2. Datasets and preprocessing

This section describes the adopted datasets, i.e. GossipCop, ISOT, FNID, CoVID-19-FNIR, and PolitiFact. Table 2 contains details about the adopted training and testing datasets.

The GossipCop dataset includes news content discussed by professional journalists, experts in gathering temporal information and social content. There are 15,969 authentic news stories and 16,817 bogus news stories. The ISOT dataset was generated by the University of Victoria's Information Security and Object Technology (ISOT) research laboratory [32]. Essentially, it is a mashup of numerous existing datasets (malicious and non-malicious). The collection contains 44,900 data points, of which 21,578 are real and the remaining 23,322 are fake news. The PolitiFact dataset [23] contains news discussed by professional journalists, 4656 of which are fake news and 3161 of which are factual news.

FNID [16] contains labelled samples, of which 16,178 labelled as 'false' and 16,358 as 'real' from 2007 to 2020. The dataset was created processing samples of the PoliticFact.com website through a specific API. The above-mentioned website is a reliable source of information, with a team of fact-checking experts assessing political news from numerous sources (CNN, BBC, Facebook).

The Covid-19-FNIR dataset [33] comprises fact-checked news reports about the Covid-19 pandemic. This dataset consists of fact-checked news collected from Poynter and Twitter from February to June 2020. There are 4386 false news stories and 3669 true news stories in this dataset.

4.3. Numerical results

This section illustrates the performance assessment of the proposed approach for detecting fake news. Accuracy and F1-score are discussed with regard to the five different datasets described in the previous section, i.e. GossipCop, ISOT, Politifact, FNID, and Covid-19-FNIR.

We compare the results obtained with and without F-transform data reduction. We performed several experiments, using different types of F-transform and different compression ratios. For the sake of brevity, we report only the best results. Only for the FNID and GossipCop datasets, the F-transform arranged as the block approach provided the best results. All the details about the basic functions are provided in the following. As illustrated in Fig. 2(a), the effect of data compression using F-transform does not significantly affect the prediction accuracy. For instance, for the ISOT dataset, the accuracy without compression is 99% and 97% with compression using the F-transform with sinusoidal basic function (2). For the PolitiFact and GossipCop datasets, the accuracy without compression is 87% and 92% respectively. Using the F1-transform with the sinusoidal basic function, the achieved accuracy for the Politicalfact dataset is 81%. For the GossipCop dataset, it is possible to reach 90% using the F-transform with cubic B-splines (3). The accuracy obtained for the Covid-19 dataset without compression is 99%, while using F-transform with the sinusoidal basic function is 98%. These results show that there is not a significant difference in the accuracy, i.e. falling in the range 1%–3%. The only exception is represented by the FNID dataset. The accuracy by F-transform with cubic B-splines is 85%, while without compression, the accuracy is 97%, showing a 12% decrease.

The F1-score with and without compression is almost the same for any of the considered datasets, as shown in Fig. 2(b). The F1-score for the ISOT dataset is almost 0.99 in both cases. For the GossipCop dataset, this score is 0.91 after data reduction, while without compression, it is 0.92. Similar results for the Covid-19 dataset, with F1-score equal to 0.97 and 0.99 with and without reduction, respectively. As for the accuracy, when it comes to FNID, there is a relatively bigger difference between the cases with and without compression, i.e. 0.93 against 0.97 respectively. Regarding the PolitiFact dataset, both accuracy and F1-score are relatively low. The F1-score of the PolitiFact dataset is 0.81 and 0.80 with and without compression, respectively. A reason for low accuracy and F1-score is the imbalanced classes, as shown in Table 2.

Fig. 3(a) shows the importance of data reduction in terms of training time. The training time is significantly reduced by the proposed F-transform-based method.

The graph shown in Fig. 3(b) is generated using (17). According to this graph, ISOT, FNID and GossipCop show higher time-saving percentages (72%, 72%, and 70%) than PolitiFact and Covid-19 (52% and 64% respectively). It must be pointed out that both PolitiFact and Covid-19 have fewer training samples than the other three datasets, as shown in Table 2. This makes it evident that the F-transform offers better time-savings for larger datasets, also showing the efficiency of the proposed data reduction technique.

The energy consumption for training the model over all the datasets is monitored using the Intel Power Gadget [34]. As shown in Fig. 4(a), the energy consumption with F-transform is significantly lower. Fig. 4(b) clearly shows the energy saving. For the ISOT, FNID, and GossipCop datasets, the energy-saving percentages (73%, 70%, and 64%) are greater than those referred to Covid-19 and PolitiFact datasets (59% and 54% respectively). This is somewhat related to the size of the datasets: ISOT, FNID, and GossipCop datasets are bigger than Covid-19 and PolitiFact datasets. In any case, we can say that the F-transform-based data reduction may yield more than 50% reduction in energy consumption during the training stage. Figs. 3b and 4b look identical at first sight, although the values are not identical but very similar (less than 5% difference on average). Recalling that the energy measured in joules (J) is the total power (watts) consumed during an interval of time (s), the above-mentioned high similarity is due to the almost constant power consumed during the time interval (i.e. no big difference between the minimum and maximum values).

Fig. 5(a) illustrates the compression ratio of each dataset. As one can observe, when the data size increases the compression ratio also increases. ISOT, GossipCop, and FNID datasets have more than 25,000 data samples (Table 2) and their compression ratios are 0.7, 0.55, 0.6. Instead, Covid-19 and PolitiFact datasets have less than 10,000 samples and the compression ratios are 0.5 and 0.4, respectively. The compression ratio and energy saving by the proposed method have a strong linear correlation, as shown in Fig. 5(b). This confirms the intuitive fact that the proposed F-transform-based data compression is more effective when applied to bigger datasets.

For the sake of completeness, we report in Table 3 the achieved accuracy by other methods using data reduction techniques, i.e. the approaches described in [9,19], even though such results are not directly comparable because the authors of [9,19] used a different number of samples for training and testing. In order to recall this fact, we introduce the ratios $R_{tr} = N_{tr,F}/N_{tr}$ and $R_{ts} = N_{ts,F}/N_{ts}$, where N_{tr} and N_{ts} are the number of instances used for training and testing in the other approaches, respectively, while the letter F denotes our approach.

As one can see, when $R_{tr} > 1$ and $R_{ts} > 1$ then the accuracy achieved by the F-transform approach is worse. This is a situation when both the number of training and testing samples used for the F-transform approach is greater. Instead, when $R_{tr} < 1$, then the accuracy of the F-transform approach is better or at least equal. This is a situation when there is a smaller number of training samples.

Finally, for the sake of completeness again, it is worth mentioning that regarding the results reported in [22,23], mentioned in Section 2, the accuracy by the F-transform approach is better, confirming this behaviour when $R_{tr} < 1$.

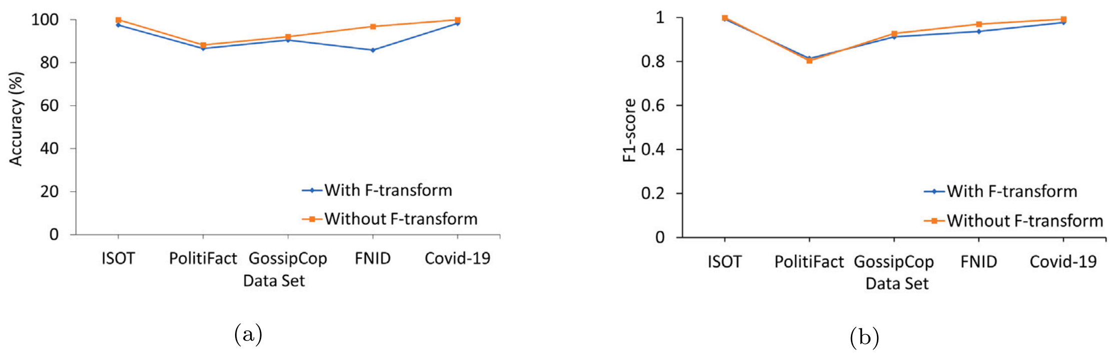

Fig. 2. Performance analysis. 2(a) Accuracy, 2(b) F1-score.

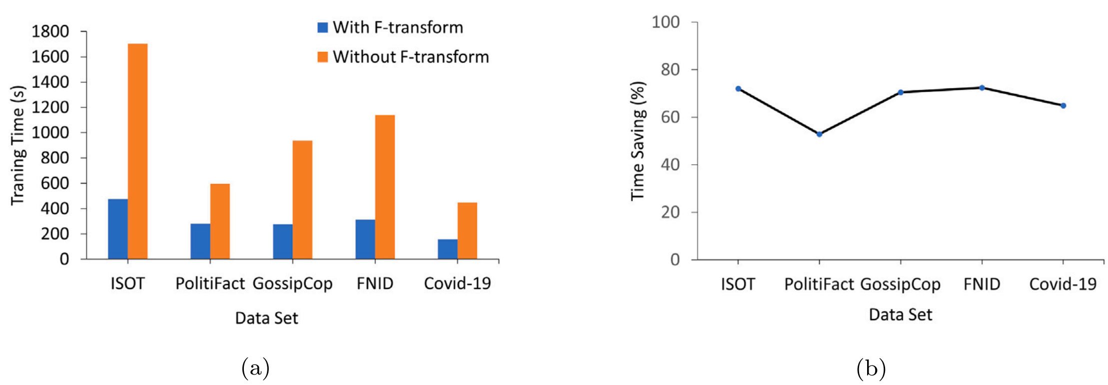

Fig. 3. Comparative graphs: 3(a) training time with and without F-transform, 3(b) time saving with F-transform.

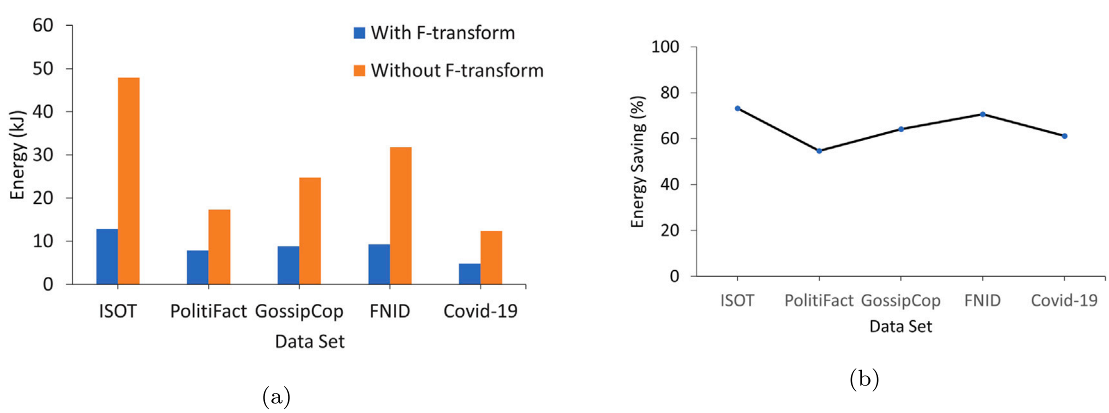

Fig. 4. Comparative graphs 4(a) energy consumption with and without F-transform, 4(b) energy saving.

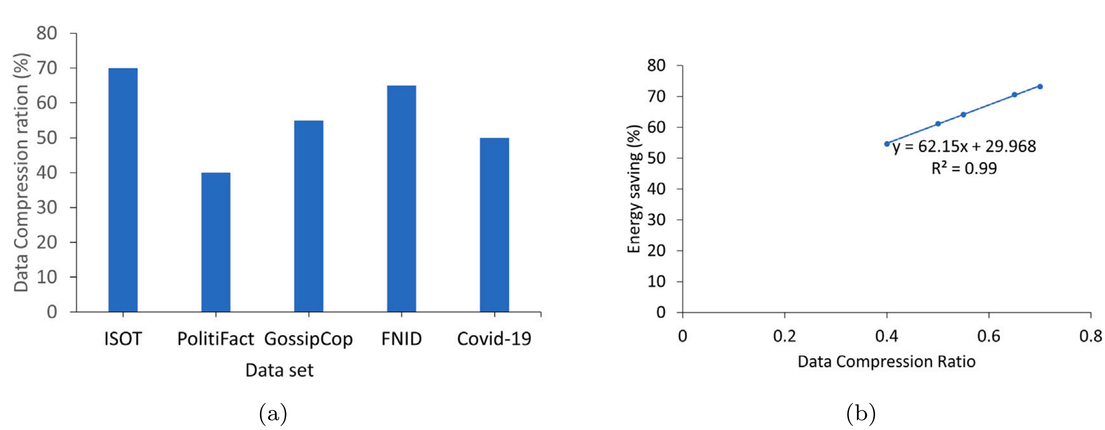

Fig. 5. F-transform based compression: 5(a) data compression ratio w.r.t datasets, 5(b) compression ratio and energy saving relationship.

Table 3
Summary of achieved accuracy in the relevant literature.

Dataset	Accuracy			R_{tr}	R_{ts}
	PPCA [19]	Fuzzy clustering [9]	F-Transform		
GossipCop	96%	NA	91%	15.41	13.11
ISOT	94%	NA	97%	0.82	9.98
Politifact	93%	NA	87%	8.93	7.825
Covid-19	NA	99%	99%	0.84	0.85
FNID	NA	NA	86%	NA	NA

5. Conclusions

The automatic detection of misleading information is attracting growing interest. DL techniques finalized for this task have been discussed in the literature. On the other hand, DL techniques have been criticized because of their energy consumption, implying low sustainability. This has motivated us to explore the potential of a data reduction technique such as the F-transform, which has been proven in the literature to be effective in reducing training time without significantly affecting the accuracy. In this paper, it has been used to reduce the cardinality of the encoded training data to be passed to an LSTM model for the classification of news. From a formal perspective, we discussed the role of the F-transform in a learning system and the conditions under which the error would have the same order with or without data reduction. The numerical experiments were performed on five publicly available datasets. The results showed the effectiveness of the approach, with significant training time and energy savings without losing accuracy in most cases. It has also been possible to observe that the proposed approach is more effective for bigger datasets than smaller ones.

CRediT authorship contribution statement

Tayasan Milinda H. Gedara: Software, Data curation, Investigation, Validation, Writing – original draft. **Vincenzo Loia:** Conceptualization, Supervision. **Stefania Tomasiello:** Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing, Supervision.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgement

Stefania Tomasiello acknowledges partial support from the Estonian Research Council, grant PRG1604.

References

[1] Number of worldwide social network users 2027 - statista, 2023, URL www.statista.com/statistics/278414/number-of-worldwide-//social-network-users/.
[2] S.M. Nematollahzadeh, S. Ozgoli, M.S. Haghighi, A. Jolfaei, Learning influential cognitive links in social networks by a new hybrid model for opinion dynamics, IEEE Trans. Comput. Soc. Syst. 8 (5) (2021) 1262–1271.
[3] M. Farajtabar, J. Yang, X. Ye, H. Xu, R. Trivedi, E. Khalil, S. Li, L. Song, H. Zha, Fake news mitigation via point process based intervention, in: Proceedings of the 34th International Conference on Machine Learning, Vol. 70, 2017, pp. 1097–1106.
[4] S. van der Linden, J. Roozenbeek, J. Compton, Inoculating against fake news about Covid-19, Front. Psychol. 11 (2020).
[5] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, R. Procter, Detection and resolution of rumours in social media: A survey, ACM Comput. Surv. 51 (2) (2018) 1–32.
[6] A. Bondielli, F. Marcelloni, A survey on fake news and rumour detection techniques, Inform. Sci. 497 (2019) 38–55.
[7] H. Saleh, A. Alharbi, S.H. Alsamhi, OPCNN-FAKE: Optimized convolutional neural network for fake news detection, IEEE Access 9 (2021) 129471–129489.
[8] B. Mohamed, H. Haytam, F. Abdelhadi, Applying fuzzy logic and neural network in sentiment analysis for fake news detection: Case of Covid-19, Studies. Comput. Intell. 1001 (2022) 387–400.
[9] M.-Y. Chen, Y.-W. Lai, Using fuzzy clustering with deep learning models for detection of COVID-19 disinformation, ACM Trans. Asian Low-Resour. Lang. Inf. Process. (2022) in press.
[10] M. Ferro, et al., Towards a sustainable artificial intelligence: A case study of energy efficiency in decision tree algorithms, Concurr. Comput. Pract. Exp. 35 (17) (2023) e6815.
[11] P. Hurtik, S. Tomasiello, A review on the application of fuzzy transform in data and image compression, Soft Comput. 23 (2019) 12641–12653.
[12] V. Loia, S. Tomasiello, A. Vaccaro, J. Gao, Using local learning with fuzzy transform: application to short term forecasting problems, Fuzzy Optim. Decis. Mak. 19 (2020) 13–32.
[13] Q. Zhang, Z. Guo, Y. Zhu, P. Vijayakumar, A. Castiglione, B.B. Gupta, A deep learning-based fast fake news detection model for cyber-physical social services, Pattern Recognit. Lett. 168 (2023) 31–38.
[14] K. Shu, L. Cui, S. Wang, D. Lee, H. Liu, dEFEND: A system for explainable fake news detection, in: 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2019, 2019, pp. 395–405.
[15] F. Sadeghi, A.J. Bidgoly, H. Amirkhani, Fake news detection on social media using a natural language inference approach, Multimedia Tools Appl. 81 (2022) 33801–33821.
[16] F. Sadeghi, A. Jalaly Bidgoly, H. Amirkhani, FNID: Fake news inference dataset, 2020, http://dx.doi.org/10.21227/fbzd-sw81.
[17] T.E. Trueman, J.A. Kumar, P. Narayanasamy, J. Vidya, Attention-based C-BiLSTM for fake news detection, Appl. Soft Comput. 110 (2021) 107600.
[18] W.S. Paka, R. Bansal, A. Kaushik, S. Sengupta, T. Chakraborty, Cross-SEAN: A cross-stitch semi-supervised neural attention model for COVID-19 fake news detection, Appl. Soft Comput. 107 (2021) 107393.
[19] D.K. Dixit, A. Bhagat, D. Dangi, Automating fake news detection using PPCA and levy flight-based LSTM, Soft Comput. 26 (2022) 12545–12557.
[20] S.H. Nithya, A. Sahayadhas, Automated fake news detection by LSTM enabled with optimal feature selection, J. Inf. Knowl. Manag. 21 (2022) 107393.
[21] H.T. Phan, N.T. Nguyen, D. Hwang, Fake news detection: A survey of graph neural network methods, Appl. Soft Comput. 139 (2023) 110235.
[22] Y. Han, S. Karunasekera, C. Leckie, Graph neural networks with continual learning for fake news detection from social media, 2020, arXiv preprint arXiv:2007.03316.
[23] K. Shu, D. Mahudeswaran, S. Wang, H. Liu, Hierarchical propagation networks for fake news detection: Investigation and exploitation, in: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 14, 2020, pp. 626–637.
[24] D.K. Sharma, B. Singh, S. Agarwal, N. Pachauri, A.A. Alhussan, H.A. Abdallah, Sarcasm detection over social media platforms using hybrid ensemble model with fuzzy logic, Electronics 12 (4) (2023).
[25] G. Bernard, G. Lebboss, Methods for word encoding: A survey, in: 2017 International Conference on Engineering and Technology, ICET, 2017, pp. 1–6.
[26] G. Kaur, K. Malik, A comprehensive overview of sentiment analysis and fake review detection, in: Lecture Notes in Networks and Systems, vol. 140, 2021, pp. 293–304.
[27] J.J. Webster, C. Kit, Tokenization as the initial phase in NLP, in: COLING 1992 Volume 4: The 14th International Conference on Computational Linguistics, 1992.
[28] F. Di Martino, V. Loia, I. Perfilieva, S. Sessa, An image coding/decoding method based on direct and inverse fuzzy transforms, Internat. J. Approx. Reason. 48 (1) (2008) 110–131.
[29] N. Littlestone, M. Warmuth, Relating data compression and learnability, in: Unpublished manuscript, 1986.
[30] S. Floyd, M. Warmuth, Sample compression, learnability, and the vapnik-chervonenkis dimension, Mach. Learn. 21 (1995) 269–304.
[31] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[32] University of Victoria, ISOT research lab, fake news detection datasets, 2023, URL https://onlineacademiccommunity.uvic.ca/isot/2022/11/27/fake-news-detection-datasets/.
[33] J.A. Saenz, S.R. Kalathur Gopal, D. Shukla, Covid-19 Fake News Infodemic Research Dataset (CoVID19-FNIR Dataset), IEEE Dataport, 2021, http://dx.doi.org/10.21227/b5bt-5244.
[34] Intel power gadget, 2019, URL https://www.intel.com/content/developer/articles/tool/power-gadget.html.