big data and cognitive computing

MDPI

Article

Topological Data Analysis Helps to Improve Accuracy of Deep Learning Models for Fake News Detection Trained on Very Small Training Sets

Ran Deng † and Fedor Duzhin *

School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371, Singapore; deng0079@ntu.edu.sg

* Correspondence: fduzhin@ntu.edu.sg; Tel.: +65-9423-5027

† The first author contributed 80% of this work.

Abstract: Topological data analysis has recently found applications in various areas of science, such as computer vision and understanding of protein folding. However, applications of topological data analysis to natural language processing remain under-researched. This study applies topological data analysis to a particular natural language processing task: fake news detection. We have found that deep learning models are more accurate in this task than topological data analysis. However, assembling a deep learning model with topological data analysis significantly improves the model's accuracy if the available training set is very small.

Keywords: topological data analysis; persistent homology; recurrent neural network; LSTM; transformer; BERT; natural language processing; text classification

Citation: Deng, R.; Duzhin, F. Topological Data Analysis Helps to Improve Accuracy of Deep Learning Models for Fake News Detection Trained on Very Small Training Sets. *Big Data Cogn. Comput.* **2022**, *6*, 74. https://doi.org/10.3390/bdcc6030074

Academic Editors: Andrea Prati, Luis Javier García Villalba and Vincent A. Cicirello

Received: 16 April 2022
Accepted: 20 June 2022
Published: 5 July 2022

1. Introduction

Topological data analysis is a collection of algorithms aimed at extracting structural information from a dataset [1–3]. Topological data analysis is concerned with a very crude structure, e.g., it can distinguish between a set of data points in the plane uniformly spread over a certain area, such as a disk, from a set of points in the plane scattered around a circle, a figure-eight shape (wedge sum of two circles), two disjoint circles, etc., as shown in Figure 1. Structures detected by topological data analysis are robust in the sense that they remain stable under homeomorphisms (continuous bijections that may significantly distort directions and distances). Topological data analysis can provide qualitative and quantitative insights of the data [2] using low dimensional representations [3]. This has allowed topological data analysis use to grow in many fields, such as image analysis [4,5], time-series analysis [6], biology [7,8], neuroscience [9], and medicine [10].

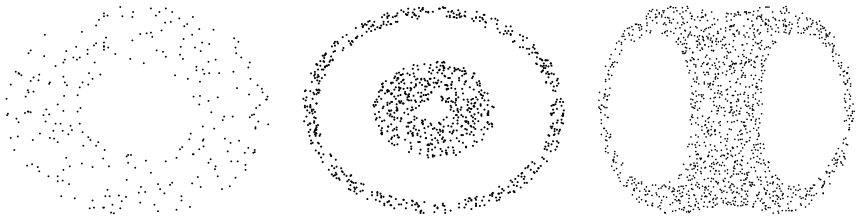


Figure 1. Datasets scattered around a circle, around two disjoint circles, and around a figure-eight shape.

Despite promising results in other areas of machine learning, applications of topological data analysis to natural language processing remain under-researched. Our paper aims to at least partially fill this gap. We explore how topological data analysis can complement deep learning algorithms for text classification. In particular, we show that assembling deep learning models with topological data analysis can significantly improve the accuracy of fake news detection if very few training data are available.

2. Related Work

Algorithms of topological data analysis are unsupervised, but they are often used to generate features that can then be fed into supervised machine learning algorithms. We will be concerned with the application of this paradigm to text classification. Below is a brief literature review of existing works in the field.

Zhu developed an algorithm in [11] that converts a text to a set of data points and identifies "1-dimensional holes" (to be clarified later) and showed that adolescent essays had more of these holes than child essays. Zhu did not apply his results to text classification, even though it was possible and could be promising. Instead, he stated the potential usefulness of topological information in enhancing text representation techniques such as discourse structure modelling or parsing.

Doshi and Zadrozny, in [12], used features generated by topological data analysis to predict a movie genre based on its plot summary. They reported an improvement of 5.7% on the Jaccard index as compared to a prior study [13] that used gated recurrent units [14].

In [15], Gholizadeth et al. came up with an algorithm that uses topological data analysis methods to generate a vector of features representing a text and applied it to classify preprints in quantitative finance according to five major categories. Gholizadeth et al. then showed that extreme gradient boosting (a popular supervised machine learning algorithm) [16] trained on such topological features outperforms convolutional neural networks [17] trained on features generated by popular vector embeddings—fastText [18], GloVe [19], and Numberbatch [20]. The same algorithm was also used in [21] to predict movie genres from movie plots, and the authors reported an improvement in accuracy as compared to a bidirectional long short-term memory (BiLSTM) artificial neural network [22]. Finally, in [23], Gholizadeh et al. applied topological data analysis to understand character networks in novels by some prominent 19th century authors.

Elyasi and Moghadam, in [24], investigated the difference in the topological structure of poems composed by two Iranian poets, Ferdowsi and Hafez. They did not try to predict the author. Instead, they explored how Mapper [25], one of the topological data analysis algorithms, can be used to visualize topological structure in texts. They found that Mapper can identify clusters of poems with high concentrations of those authored by Ferdowsi or those authored by Hafez.

In most of these studies, models trained on features generated by topological data analysis consistently outperformed models trained on other features. However, authors of these studies spent considerably more effort on fine-tuning topological data analysis than they did on conventional machine learning algorithms. For instance, to the best of our knowledge, no one has compared topological data analysis to BERT [26] (state-of-the-art deep learning model for natural language processing).

Our research question is comparing topological data analysis to state-of-the-art deep learning models for text classification. The particular scenario we are looking at is fake news detection in a situation when very few training data are available. We show that features generated by topological data analysis improve performance of deep learning algorithms trained on very small datasets (between 50 and 400 observations). The workflow of our analysis is shown in Figure 2.

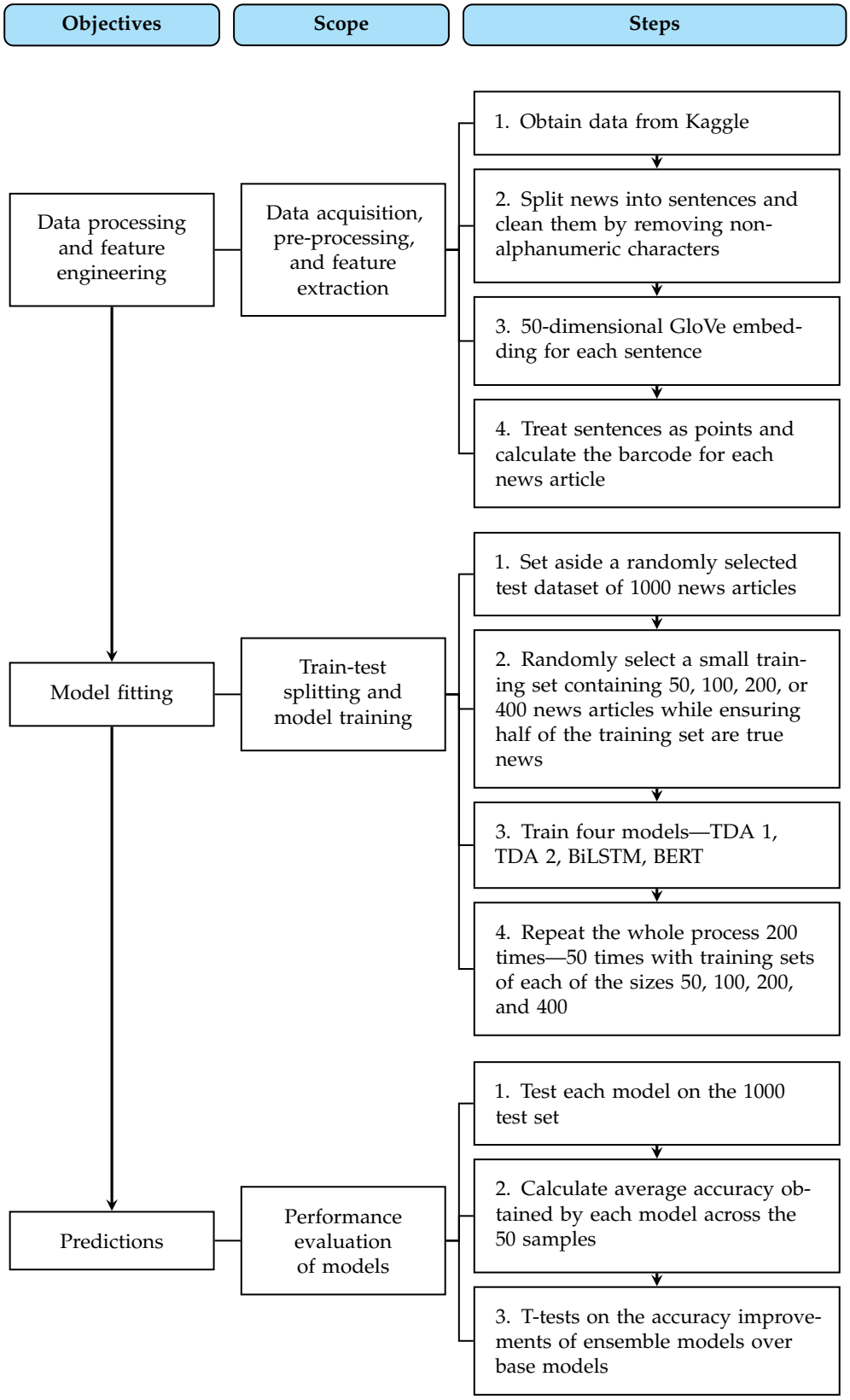
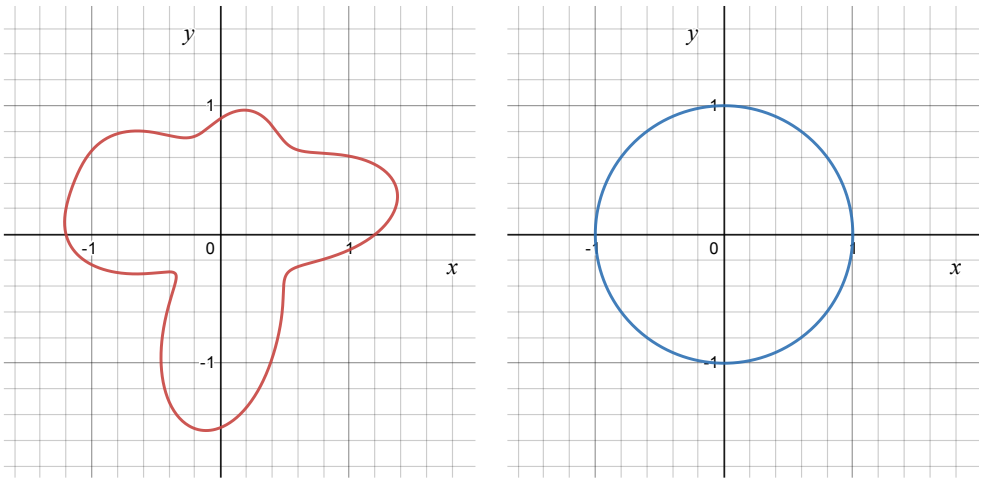


Figure 2. Workflow of our analysis.

3. Topological Data Analysis

3.1. Betti Numbers

In classical algebraic topology, algebraic structures (such as groups and vector spaces) are assigned to topological spaces. Intuitively, a topological space can be thought of as a geometric structure stripped of any metric information. From this perspective, any simple closed curve is not distinguishable from a circle (the proper term is that they are *homeomorphic*)—see Figure 3.



Some simple closed curve A circle

Figure 3. Any simple closed curve is homeomorphic to a circle, i.e., these two figures are essentially the same as a topological space.

Intricate details of algebraic topology are beyond the scope of our paper. Instead of diving into complicated algebra, we will try to explain some basic intuition behind *Betti numbers*, one of the most popular tools in algebraic topology. Betti numbers of a topological space X form an infinite sequence

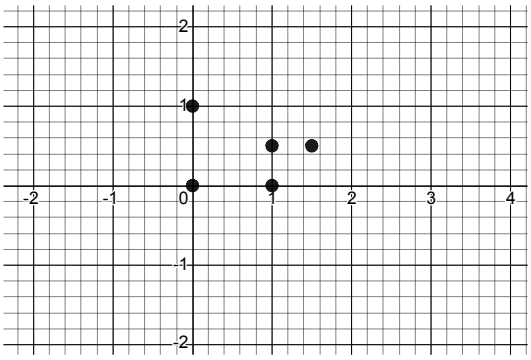
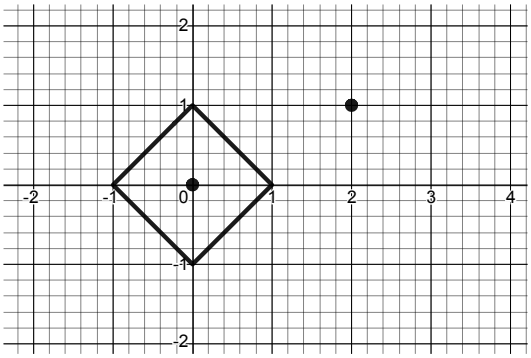
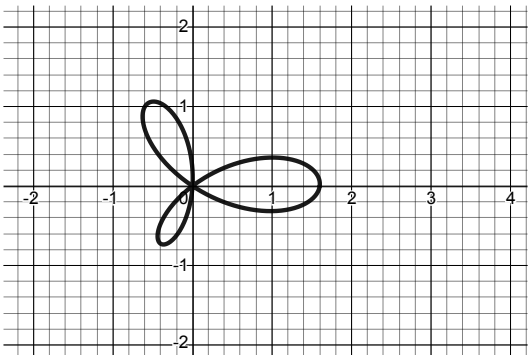
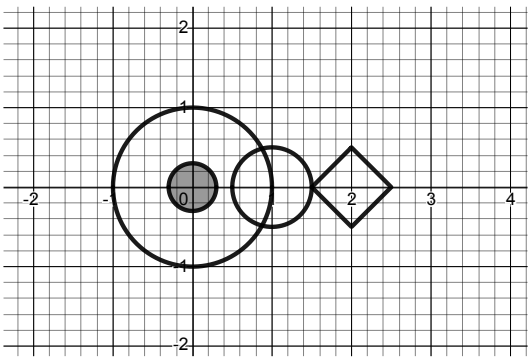
$$b_0(X), b_1(X), b_2(X), \cdots$$

of nonnegative integers that captures certain information about the topological space. While the sequence is infinite, all its elements are zero starting at some position—namely, $b_i(X) = 0$ whenever $i > \dim X$.

The first two Betti numbers, b_0 and b_1, are easier to understand than b_i with $i \geq 2$. We will only explain these two, as we will not need the rest of Betti numbers. The Betti number $b_0(X)$ is simply the number of path-connected components of X, and $b_1(X)$ can be thought of as the number of simple closed curves in X that do not enclose an area that is entirely contained in X. Informally, $b_1(X)$ is the number of one-dimensional holes in X.

A few examples of topological spaces and their Betti numbers are shown in Table 1. Note that the topological spaces shown in Table 1 are subspaces of the plane $\mathbb{R}^2$. Such two-dimensional spaces are easy to visualize directly, and the value of topological data analysis may not seem apparent if one thinks just of two-dimensional datasets. However, the true power of topological data analysis is in working with high-dimensional datasets, those that cannot readily be plotted or even imagined.

Table 1. Some topological spaces and their Betti numbers.

Topological Space X	Description	$b_0(X)$	$b_1(X)$
	Five points	5	0
	Square and two points	3	1
	Three closed curves	1	3
	Two circles, square and a disk. The two circles and the square form four closed loops.	2	4

3.2. From a Dataset to a Topological Space

A numeric dataset containing N observations of p independent variables can be thought of as a point cloud, i.e., a finite subset of $\mathbb{R}^p$. For topological data analysis, the values of independent variables are not essential. What is important is that distances between every pair of data points are known.

Let X be a finite metric space, i.e., a finite set with known distances between every two points. Further, let $\delta > 0$ be a threshold. We will construct a topological space X_δ

called the *Vietoris–Rips complex* by connecting any two points whose distance is less than δ with an interval, any three points whose all pairwise distances are less than δ with a 2-simplex (solid triangle), any four points whose all pairwise distances are less than δ with a 3-simplex (tetrahedron), etc. For different values of δ, we will see different topological spaces as shown in Figure 4.

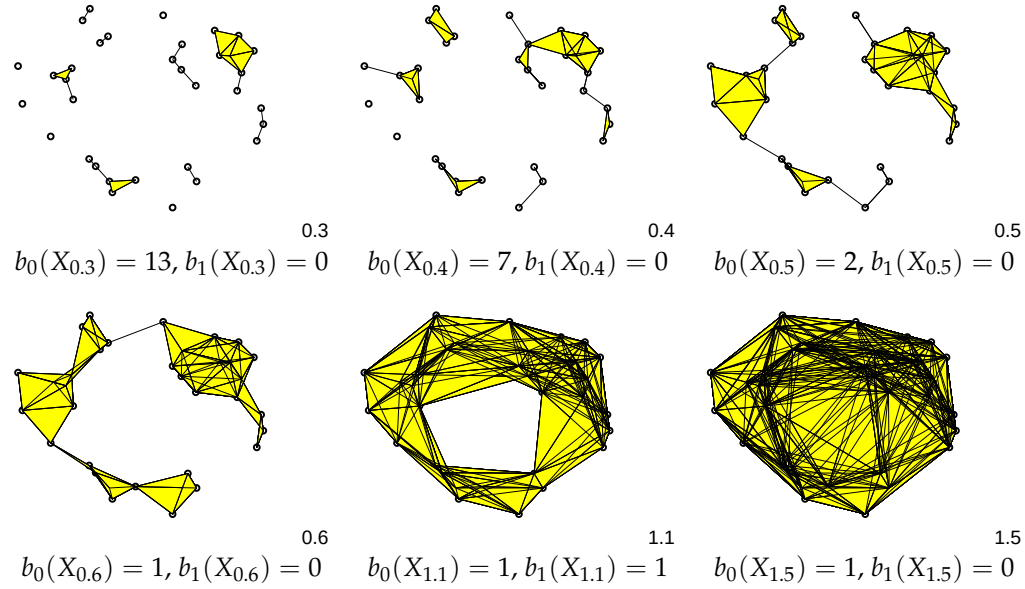


Figure 4. Vietoris–Rips complexes constructed out of a plane dataset for different values of the threshold δ and their Betti numbers.

This construction produces a one-parametric family of topological spaces $\{X_\delta\}_{\delta \geq 0}$ with the property $X_{\delta_1} \subset X_{\delta_2}$ whenever $\delta_1 < \delta_2$.

3.3. Persistent Homology

Let X be a finite metric space/a point cloud/a numeric dataset. As shown in Figure 4, Betti numbers of Vietoris–Rips complexes X_δ change as δ is varied. It is clear that $b_0(X_\delta)$ is a decreasing function of δ, for b_0 is the number of connected components and, as δ increases, connected components merge but never fall apart. On the other hand, $b_1(X_\delta)$, which can be thought of as the number of circular holes, will be 0 for small and large values of δ, but it may be positive for some intermediate value of δ. In Figure 4, the Vietoris–Rips complex $X_{1.1}$ has a hole and therefore a positive Betti number b_1.

Persistent homology, one of main tools in topological data analysis, is an algorithm that puts all information about Betti numbers of X_δ for different values of δ into one diagram called a *barcode*. To construct such a barcode, one chooses δ_{min} and δ_{max} and traces how X_δ changes when δ is varied between δ_{min} and δ_{max}. Whenever Betti numbers of X_δ change, we either start (if the Betti number increases) or end (if the Betti number decreases) a horizontal line.

The central idea in persistent homology is that long lines in the barcode represent true topological features of the data, while short lines are just random artefacts. For example, Figure 5 shows a dataset scattered around the wedge sum of three circles. The barcode has three long lines in dimension 1 and one line in dimension 0 that is much longer than the rest. At the same time, all shorter lines in dimension 0 disappeared before the first long line in dimension 1 appeared. This observation tells us that the Betti numbers of the data are $b_0(X) = 1$ and $b_1(X) = 3$.

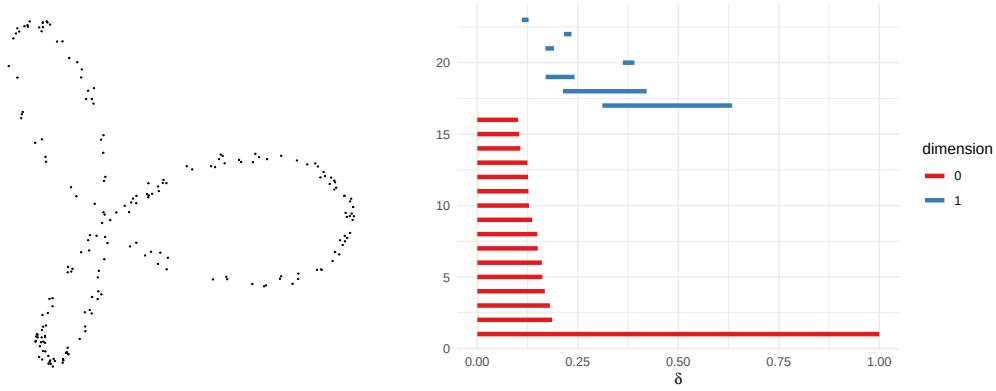


Figure 5. A dataset scattered around the wedge sum of three circles and its barcode for $0.15 \leq \delta \leq 1$. The three longest lines in dimension 1 correspond to the three circles in the dataset. The longest line in dimension 1 shows that the biggest hole in X_δ exists for $0.31 \leq \delta \leq 0.63$.

3.4. Persistent Image

Barcodes, such as the one shown in Figure 5 on the right, describe certain topological properties of datasets. To use these properties as features for predictive modelling (i.e., for text classification), one needs to create a vector representation of a barcode. There are three main difficulties here. The first is that the number of lines in a barcode is not fixed, but the number of features for predictive modelling is. The second difficulty is that the vector representation should not change much if one adds a short line to a barcode, as such a short line is thought to be a random artefact rather than a true feature of the dataset. The third difficulty is that the representation should not change if we re-arrange the order of the lines.

A *persistent image* [27] is one such vector representation of a barcode. To create a persistent image from a barcode, we first fix a dimension (in the current study, we will only use dimensions 0 and 1) and then write a complete list

$$(b_1, l_1), (b_2, l_2), \ldots, (b_d, l_d)$$

of lines in the barcode in that dimension. Here, (b_i, l_i) represents the line that starts at b_i and ends at $b_i + l_i$. The next thing is to define the special function

$$\rho(x, y) = \sum_{i=1}^{d} w(l_i) \times \frac{1}{2\pi h^2} e^{-\frac{(x-b_i)^2+(y-l_i)^2}{2h^2}}, \tag{1}$$

where $w(u)$ is a piece-wise linear weighting function of one variable given by

$$w(u) = \begin{cases} 0 & \text{if } u \leq 0, \\ \frac{u}{L_{max}} & \text{if } 0 < u < L_{max}, \\ 1 & \text{if } u \geq L_{max}. \end{cases}$$

Here, L_{max} is the length of the longest line in the barcode, and $h > 0$ is a hyper-parameter of the algorithm.

The persistent image is then created by integrating the special function (1) over the rectangle $[\delta_{min}, \delta_{max}] \times [0, \delta_{max} - \delta_{min}]$. Specifically, let us split it into p^2 smaller rectangular regions. Then, the intensity of the (i, j)-pixel of the persistent image for $i, j = 1, 2, \ldots, p$ is given by

$$p_{ij} = \int_{\delta_{min}+(i-1)\frac{\delta_{max}-\delta_{min}}{p}}^{\delta_{min}+i\frac{\delta_{max}-\delta_{min}}{p}} \int_{(j-1)\frac{\delta_{max}-\delta_{min}}{p}}^{j\frac{\delta_{max}-\delta_{min}}{p}} \rho(x, y)\, dy\, dx.$$

Here, p is the second hyper-parameter of the algorithm.

An example of some persistent images is shown in Figure 6. Note that while a persistent image can be visualized as an image, as the name suggests, it is still, under the hood, just a vector. Thus, we can use a persistent image as a vector of features for supervised machine learning.

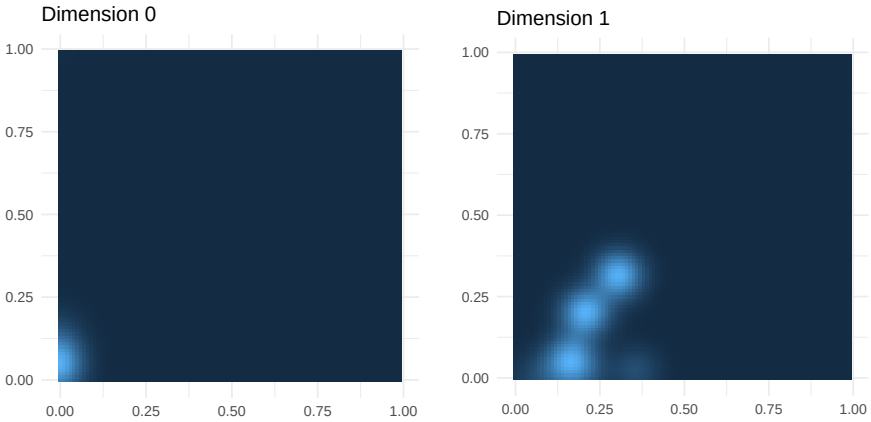


Figure 6. Persistent images of the barcode shown in Figure 5 in dimension 0 (**left**) and dimension 1 (**right**). Here, $\delta_{min} = 0$, $\delta_{max} = 1$, $h = 0.05$, and $p = 100$.

4. Data

The dataset was obtained from the public data platform Kaggle, uploaded by Clément Bisaillon with acknowledgements to [28,29]. It contains 21,417 texts of real news articles collected primarily from Reuters and 23,481 texts of fake news collected from various sources.

Below is a sample of a real news article:

> *A summer spat between President Donald Trump and Senate Majority Leader Mitch McConnell has turned into a warm embrace—and all it took was a sweeping rewrite of the U.S. tax code. For months, McConnell urged the president to lock his cell phone in a drawer and retire his signature tweets that have Washington abuzz on a daily basis. He even chided Trump for having "excessive expectations" of Congress. For his part, Trump scorched McConnell in August for failing to repeal Obamacare, sidestepped reporters' questions over whether the senator should retire and tweeted, "Mitch, get to work." But with Congress' passage of the tax bill this week, giving Trump his first major legislative victory, the president tweeted on Wednesday, "I would like to congratulate @SenateMajLdr on having done a fantastic job."*

Below is a sample of a fake news article:

> *Pope Francis used his annual Christmas Day message to rebuke Donald Trump without even mentioning his name. The Pope delivered his message just days after members of the United Nations condemned Trump s move to recognize Jerusalem as the capital of Israel. The Pontiff prayed on Monday for the peaceful coexistence of two states within mutually agreed and internationally recognized borders.*

We used the R "OpenNLP" package [30] to split each article into sentences and remove all articles containing fewer than 3 or more than 100 sentences. Then, we changed everything to lower case and removed non-alphanumeric characters, English stopwords, and the word "Reuters", as most of the real news articles come from Reuters.

The next step is representing every news article as a finite metric space $X \subset \mathbb{R}^{50}$ so that we can later apply topological data analysis to X. To do it, we downloaded 50-dimensional GloVe word embeddings pre-trained on Wikipedia and Gigaword, [19], and then represented every sentence by the sum of word embeddings for all the words in the sentence. Thus, every sentence of a news article became a point of a finite metric space

$X \subset \mathbb{R}^{50}$ and the distance between sentences is just the Euclidean distance in $\mathbb{R}^{50}$. We computed persistent homology with the R package "TDAstats" [31].

5. Predictive Models

Our research objective is to explore if topological data analysis helps to improve predictive models for detecting fake news trained on very small datasets. To achieve this objective, we trained a number of models.

We started with an l_1-regularized logistic regression (LASSO) with persistent images as features. We call this model "TDA" from "topological data analysis". We actually tried two different versions of TDA. The first version, TDA 1, uses pixels of the persistent image in dimension 0 as features. The second version, TDA 2, uses persistent images in both dimension 0 and dimension 1 as features, i.e., it has twice as many features as TDA 1. The best of TDA 1 and TDA 2 is then simply TDA.

Then we trained two standard deep learning models for natural language processing—BiLSTM (bi-directional long short-term memory recurrent neural network) whose input is a sequence of 50-dimensional GloVe vectors and BERT (bidirectional encoder representations from transformers) that is mostly pre-trained, but we re-trained the output classification layer. The hyperparameter specifications for the three basic models are given in Table 2.

We trained these models, i.e., TDA 1/2 (LASSO with persistent images as features), BiLSTM, and BERT, on randomly selected training sets of sizes 50, 100, 200, and 400 and repeated this experiment 50 times, i.e., the total number of models we trained is

$$4 \times 50 \times 4 = 800.$$

We found that deep learning models BiLSTM and BERT are more accurate than TDA. However, BiLSTM and BERT are not 100% accurate, and we went on to explore if assembling either deep learning model with TDA improves the former.

An ensemble of a deep learning model and TDA is constructed as follows. We begin with picking the value of w, a hyper-parameter controlling the relative weight of the deep learning model in the ensemble. Then, if $p_{\mathrm{DL}}(y = \text{fake})$ is the probability that the article is fake estimated by the deep learning model and $p_{\mathrm{TDA}}(y = \text{fake})$ is the probability that the article is fake estimated by TDA, then

$$p_{ens}(y = \text{fake}) = \frac{w \times p_{\mathrm{DL}}(y = \text{fake}) + p_{\mathrm{TDA}}(y = \text{fake})}{w + 1} \tag{2}$$

is the probability that the article is fake estimated by the ensemble. We have tried the values $w = 1, 2, \ldots, 20$.

Table 2. Model specifications.

Model	Specification
TDA	We used persistent images as features to train the LASSO model with five-fold cross-validation to select the optimal value of the regularization constant. The hyperparameter values to generate persistent images are $p = 20$, $h = 0.1$, as recommended by [27].
BiLSTM	We used pre-trained 50-dimensional GloVe as the embedding layer, with a maximum sequence length of 100. The BiLSTM layer has 64 units and hyperbolic tangent activation. Then we have a dropout layer with the dropout rate of 0.2 and sigmoid output. We trained the model with a learning rate of 0.002 for 10 epochs with batch size 64.
BERT	We used the pre-trained base BERT with 12 encoders, 768-dimensional embeddings, and 12 bidirectional self-attention heads. We then trained the output layer with a sigmoid classifier with a learning rate of 2×10^{-5} as recommended by [26] for 2 epochs with batch size 4.

6. Results and Discussions

6.1. Main Experiment with Real and Fake News Written by Human Authors

The average test accuracy of 32 different models, i.e., TDA 1, TDA 2, BiLSTM, BERT, and ensembles of BERT with TDA 1 and TDA 2 trained on training sets of sizes 50, 100, 200, and 400, is shown in Table 3. Each of the 32 models was trained 50 times on a training set selected randomly. Note that assembling with TDA improves the performance of BERT and BiLSTM for all training set sizes. Sometimes, this improvement is slightly bigger for a TDA model that only uses pixels of persistent image in dimension 0 (TDA 1) and sometimes it is slightly bigger for a TDA model that uses pixels of persistent images in dimensions 0 and 1 together (TDA 2). Still, the advantage of TDA 2 over TDA 1 seems negligible and is probably due to chance.

Table 3. We trained each of 8 models (TDA 1, TDA 2, BiLSTM, BERT, and ensembles of BERT with TDA 1 and TDA 2) 50 times on a randomly chosen training set. We did it for training sets of sizes 50, 100, 200, and 400. Average test accuracy and the standard deviation are reported in this table. The highest test accuracy for each training set size is bolded.

Model/Training Set Size	50	100	200	400
TDA 1	0.728 (0.037)	0.754 (0.034)	0.763 (0.029)	0.774 (0.023)
TDA 2	0.734 (0.034)	0.749 (0.030)	0.774 (0.029)	0.792 (0.024)
BiLSTM	0.890 (0.041)	0.914 (0.024)	0.944 (0.013)	0.969 (0.009)
BERT	0.638 (0.104)	0.833 (0.138)	0.972 (0.038)	0.992 (0.016)
BiLSTM + TDA 1	**0.912** (0.034)	0.932 (0.018)	0.956 (0.010)	0.975 (0.006)
BERT + TDA 1	0.790 (0.053)	0.877 (0.090)	**0.977** (0.034)	**0.993** (0.014)
BiLSTM + TDA 2	0.908 (0.033)	**0.927** (0.021)	0.954 (0.012)	0.974 (0.008)
BERT + TDA 2	0.785 (0.047)	0.884 (0.076)	0.976 (0.035)	**0.993** (0.014)

The same result is shown graphically in Figure 7, where we just included the best of TDA 1 and TDA 2 in each case. Again we see that while the deep learning models BiLSTM and BERT are more accurate than TDA, assembling either of them with TDA improves accuracy.

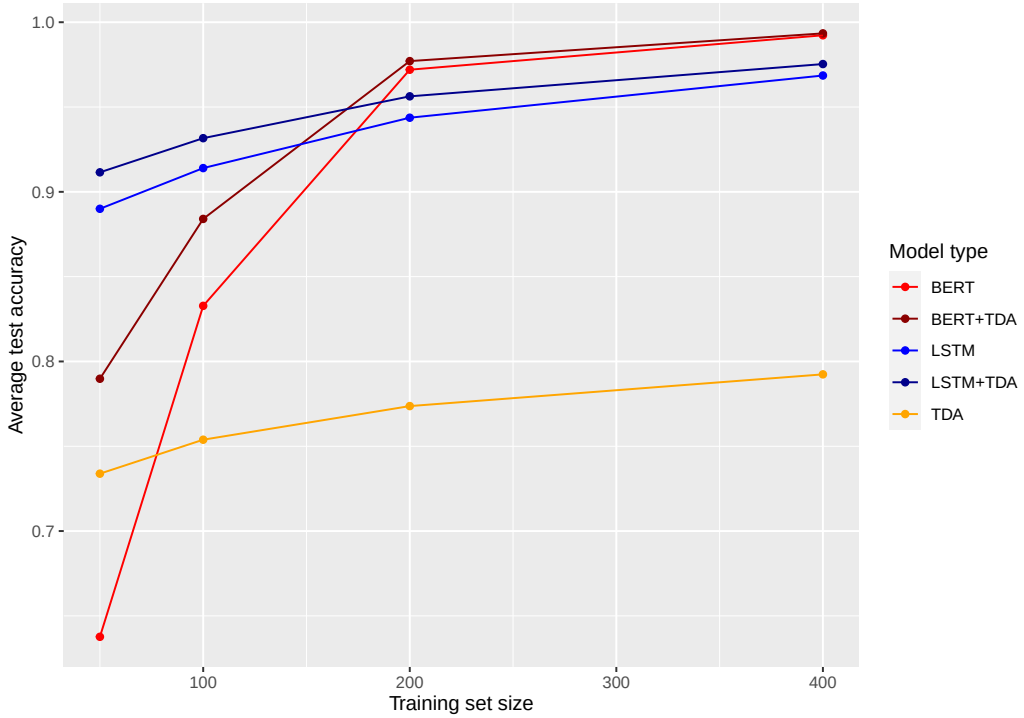


Figure 7. Average test accuracy vs. training set size.

Now, we will shed some light on how we chose the hyper-parameter w that controls the relative weight of the deep learning model in an ensemble of a deep learning model and TDA given by (2). We have tried $w = 1, 2, \ldots, 20$. The average test accuracy of each of the 20 different versions of an ensemble BiLSTM plus TDA is shown in Figure 8 and the average test accuracy of each of the 20 different versions of an ensemble BERT plus TDA is shown in Figure 9. The values of the hyper-parameter w that yield the best test accuracy are reported in Table 4.

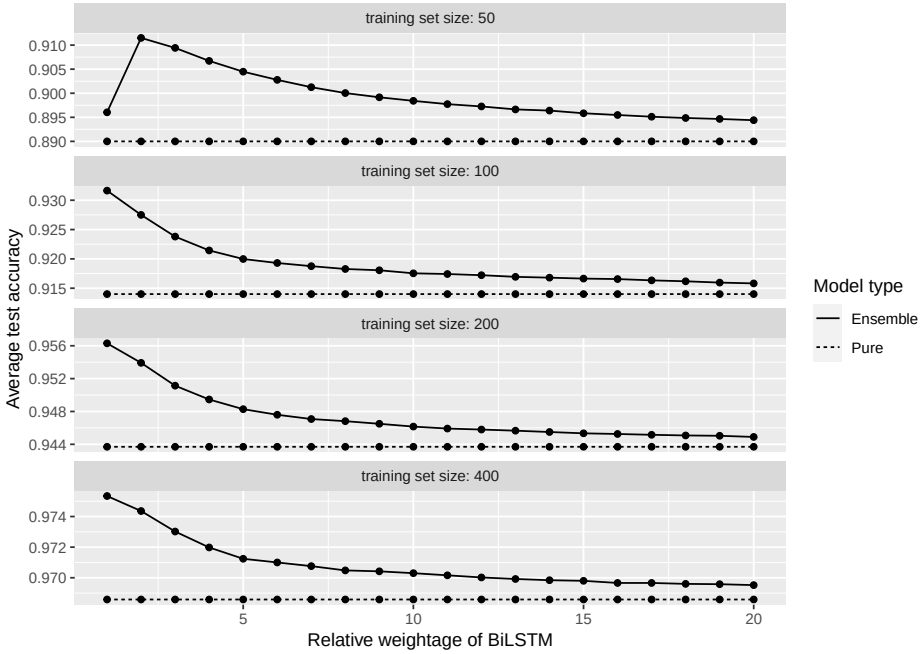


Figure 8. Average test accuracy of BiLSTM and TDA ensembles. The horizontal line is the accuracy of the pure BiLSTM model given as a reference.

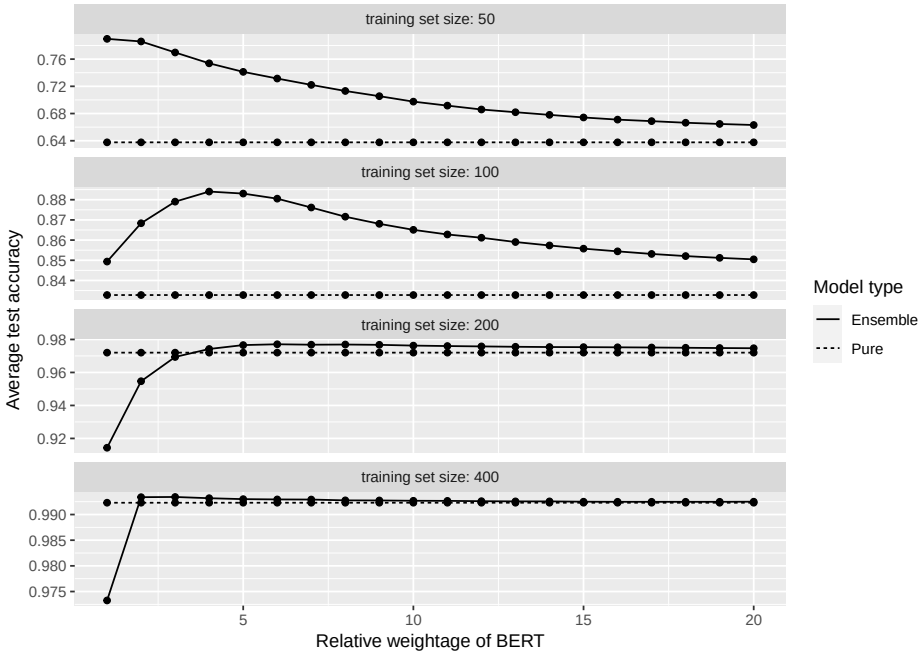


Figure 9. Average test accuracy of BERT and TDA ensembles. The horizontal line is the accuracy of the pure BERT model given as a reference.

Table 4. Accuracy table of all ensembles of a deep learning model with TDA. Every time, we report the average accuracy of models trained on 50 randomly chosen training sets, the optimal value of w (weight of the deep learning model in the ensemble), improvement in average accuracy achieved by ensembling the deep learning model with TDA, and the p-value of the t-test to test the hypothesis that the ensemble has a higher test accuracy than the pure deep learning model.

Ensemble	Training Set Size	Accuracy	Optimal w	Accuracy Improvement	p-Value
BiLSTM + TDA	50	0.912	2	0.022	<0.001
BiLSTM + TDA	100	0.932	1	0.018	<0.001
BiLSTM + TDA	200	0.956	1	0.013	<0.001
BiLSTM + TDA	400	0.975	1	0.007	<0.001
BERT + TDA	50	0.790	1	0.152	<0.001
BERT + TDA	100	0.884	4	0.051	<0.001
BERT + TDA	200	0.977	6	0.005	<0.001
BERT + TDA	400	0.993	3	0.001	0.005

6.2. An Experiment with AI-Generated Fake News

We have also experimented with fake news generated by GPT-3 [32], a state-of-the-art general-purpose AI engine. Specifically, we generated 100 fake news articles with the prompt "generate a piece of fake news article" and fed these fake news articles into all our models. Note that AI-generated fake news are different from human-written fake news that our models were trained on. Here is a typical example of an AI-generated article:

> *The world was shocked today as reports came in that the moon had exploded. Witnesses say that they saw a bright light in the sky, followed by a loud explosion. The moon is now nothing more than a debris field. Scientists are still trying to determine what exactly caused the moon to explode. Some believe that it was a natural phenomenon, while others believe that it was caused by a man-made weapon.*

Typically, the story presented in such articles is ridiculous, but the grammar and vocabulary are perfectly smooth. By contrast, human authors of fake news are not professional journalists. Usually, the story presented in fake news written by human authors is somewhat convincing, but the language is not as smooth.

The accuracy of our models for detecting fake news articles generated by GPT-3 is shown in Figure 10. There are a few insights here. First, topological data analysis does not help much in detecting fake news generated by GPT-3. This means that topological data analysis picks features that have to do more with the flow of text rather than with the story presented. The second interesting thing is that BiLSTM trained on just 50 news articles (25 real and 25 fake, but a different type of fake) is already pretty powerful in detecting AI-generated fake news. The third observation is that accuracy of BiLSTM and TDA does not grow with bigger training set sizes. It means that AI–generated fake news articles are really very different to the fake news generated by human authors that our models were trained on.

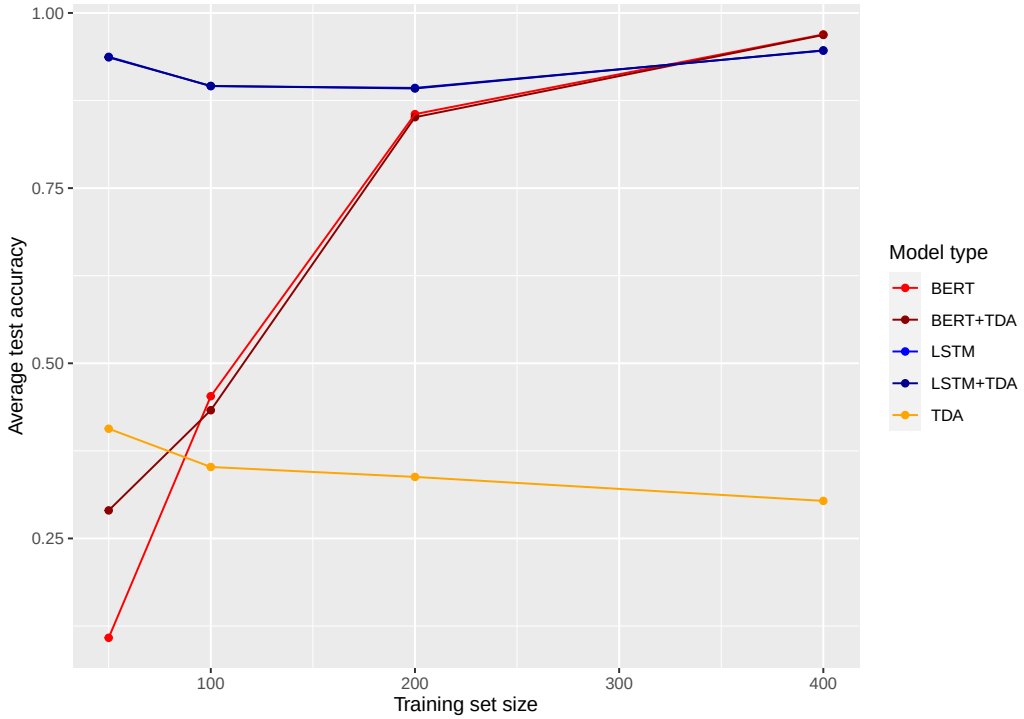


Figure 10. Average accuracy of our models in detecting fake news articles generated by GPT-3 (i.e., not the type of articles the models were trained on). Note that the LSTM accuracy line coincides with the LSTM + TDA accuracy line.

7. Conclusions

Applications of topological data analysis to text classification are still under-researched. In particular, at the time this paper is written, there are no published attempts at using TDA methods for fake news detection, no benchmarking of TDA against state-of-the-art models, such as BERT, no attempts to combine TDA with deep learning models to improve accuracy of prediction, and very little progress in explaining exactly which aspects of natural languages are captured by TDA.

Our results show that state-of-the-art deep learning models are more accurate than TDA for fake news detection. However, ensembling deep learning models with TDA improves the accuracy of the former if available training sets are small. The fewer data are available for training, the greater the improvement in accuracy.

Although we are not able to explain what exactly in natural language is captured by topological features, the results of our experiments with detecting AI–generated fake news show that topological features have to do more with smoothness and complexity of grammar structures rather than with lexical meaning. This is consistent with previous findings in [11,15,24].

Even though our results are promising, they are based on the specific task of fake news identification, which may or may not apply well to other text classification tasks. Hence, it is important to try our approach in the future for other tasks such as genre detection and authorship profiling before making any far-reaching conclusions. Furthermore, while we did some basic data cleaning, we did not thoroughly experiment with data filtering by a codensity threshold, and this may, according to [2], noticeably improve the performance of TDA.

Author Contributions: R.D. is a student who essentially did all the work. F.D. is the supervisor who proposed the research topic and wrote this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data set is publicly available at https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset, accessed on 15 April 2022.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

TDA	Topological data analysis
LASSO	Least absolute shrinkage and selection operator
BiLSTM	Bidirectional long short-term memory
BERT	Bidirectional encoder representations from transformers
GloVe	Global vectors for word representation

References

1. Wasserman, L. Topological Data Analysis. *Annu. Rev. Stat. Appl.* **2018**, *5*, 501–532. [CrossRef]
2. Carlsson, G. Topology and Data. *Bull. Am. Math. Soc.* **2009**, *46*, 255–308. [CrossRef]
3. Ghrist, R. Barcodes: The persistent topology of Data. *Bull. Am. Math. Soc.* **2007**, *45*, 61–76. [CrossRef]
4. Asaad, A.; Jassim, S. Topological Data Analysis for Image Tampering Detection. In *Digital Forensics and Watermarking*; Kraetzer, C., Shi, Y.Q., Dittmann, J., Kim, H.J., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 136–146.
5. Bernstein, A.; Burnaev, E.; Sharaev, M.; Kondrateva, E.; Kachan, O. Topological data analysis in computer vision. In Proceedings of the Twelfth International Conference on Machine Vision (ICMV 2019), Amsterdam, The Netherlands, 16–18 November 2019; Osten, W., Nikolaev, D.P., Eds.; International Society for Optics and Photonics, SPIE: Bellingham, WA, USA, 2020; Volume 11433, pp. 673–679. [CrossRef]
6. Seversky, L.M.; Davis, S.; Berger, M. On Time-Series Topological Data Analysis: New Data and Opportunities. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Las Vegas, NV, USA, 27–30 June 2016.
7. Topaz, C.M.; Ziegelmeier, L.; Halverson, T. Topological Data Analysis of Biological Aggregation Models. *PLoS ONE* **2015**, *10*, 1–26. [CrossRef] [PubMed]
8. Xia, K.; Wei, G.W. Persistent homology analysis of protein structure, flexibility, and folding. *Int. J. Numer. Methods Biomed. Eng.* **2014**, *30*, 814–844. [CrossRef] [PubMed]
9. Sizemore, A.E.; Phillips-Cremins, J.E.; Ghrist, R.; Bassett, D.S. The importance of the whole: Topological Data Analysis for the network neuroscientist. *Netw. Neurosci.* **2019**, *3*, 656–673. [CrossRef]
10. Rucco, M.; Falsetti, L.; Herman, D.; Petrossian, T.; Merelli, E.; Nitti, C.; Salvi, A. Using Topological Data Analysis for diagnosis pulmonary embolism. *arXiv* **2014**, arXiv:1409.5020.
11. Zhu, X. Persistent homology: An introduction and a new text representation for natural language processing. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013.
12. Doshi, P.; Zadrozny, W. Movie Genre Detection Using Topological Data Analysis. In *Statistical Language and Speech Processing*; Dutoit, T., Martín-Vide, C., Pironkov, G., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 117–128.
13. Hoang, Q. Predicting Movie Genres Based on Plot Summaries. *arXiv* **2018**, arXiv:1801.04813.
14. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
15. Gholizadeh, S.; Seyeditabari, A.; Zadrozny, W. A Novel Method of Extracting Topological Features from Word Embeddings. *arXiv* **2020**, arXiv:2003.13074v2.
16. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
17. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent convolutional neural networks for text classification. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
18. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. *arXiv* **2016**, arXiv:1607.01759.
19. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
20. Speer, R.; Chin, J.; Havasi, C. Conceptnet 5.5: An open multilingual graph of general knowledge. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
21. Gholizadeh, S.; Savle, K.; Seyeditabari, A.; Zadrozny, W. Topological Data Analysis in Text Classification: Extracting Features with Additive Information. *arXiv* **2020**, arXiv:2003.13138.
22. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]

23. Gholizadeh, S.; Seyeditabari, A.; Zadrozny, W. Topological signature of 19th century novelists: Persistent homology in text mining. *Big Data Cogn. Comput.* **2018**, *2*, 33. [CrossRef]
24. Elyasi, N.; Moghadam, M.H. An Introduction to a New Text Classification and Visualization for Natural Language Processing Using Topological Data Analysis. *arXiv* **2019**, arXiv:1906.01726.
25. Singh, G.; Mémoli, F.; Carlsson, G.E. Topological methods for the analysis of high dimensional data sets and 3d object recognition. In Proceedings of the Eurographics Symposium on Point-Based Graphics, Prague, Czech Republic, 2–3 September 2007; Volume 2.
26. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
27. Adams, H.; Emerson, T.; Kirby, M.; Neville, R.; Peterson, C.; Shipman, P.; Chepushtanova, S.; Hanson, E.; Motta, F.; Ziegelmeier, L. Persistence images: A stable vector representation of persistent homology. *J. Mach. Learn. Res.* **2017**, *18*, 1–35.
28. Ahmed, H.; Traore, I.; Saad, S. Detecting opinion spams and fake news using text classification. *Secur. Priv.* **2017**, *1*, e9. [CrossRef]
29. Ahmed, H.; Traoré, I.; Saad, S. Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In Proceedings of the International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, Vancouver, BC, Canada, 25–27 October 2017.
30. Hornik, K. openNLP: Apache OpenNLP Tools Interface, R Package Version 0.2-6. 2017. Available online: https://CRAN.R-project.org/package=openNLP (accessed on 15 April 2022).
31. Wadhwa, R.R.; Williamson, D.F.; Dhawan, A.; Scott, J.G. TDAstats: R pipeline for computing persistent homology in topological data analysis. *J. Open Source Softw.* **2018**, *3*, 860.
32. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.