Concept Drift Adaptation in Text Stream Mining Settings: A Systematic Review

CRISTIANO MESQUITA GARCIA, Instituto Federal de Santa Catarina, Brazil and Programa de Pós-Graduação em Informática (PPGIa), Pontifícia Universidade Católica do Paraná (PUCPR), Brazil
RAMON ABILIO, Instituto Federal de São Paulo, Brazil and Universidade de Campinas, Brazil
ALESSANDRO LAMEIRAS KOERICH, École de Technologie Supérieure, Université du Québec, Canada
ALCEU DE SOUZA BRITTO JR, Programa de Pós-Graduação em Informática (PPGIa), Pontifícia Universidade Católica do Paraná (PUCPR), Brazil and Universidade Estadual de Ponta Grossa, Brazil
JEAN PAUL BARDDAL, Programa de Pós-Graduação em Informática (PPGIa), Pontifícia Universidade Católica do Paraná (PUCPR), Brazil

The society produces textual data online in several ways, *e.g.*, via reviews and social media posts. Therefore, numerous researchers have been working on discovering patterns in textual data that can indicate peoples' opinions, interests, *etc.* Most tasks regarding natural language processing are addressed using traditional machine learning methods and static datasets. This setting can lead to several problems, *e.g.*, outdated datasets and models, which degrade in performance over time. This is particularly true regarding concept drift, in which the data distribution changes over time. Furthermore, text streaming scenarios also exhibit further challenges, such as the high speed at which data arrives over time. Models for stream scenarios must adhere to the aforementioned constraints while learning from the stream, thus storing texts for limited periods and consuming low memory. This study presents a systematic literature review regarding concept drift adaptation in text stream scenarios. Considering well-defined criteria, we selected 48 papers published between 2018 and August 2024 to unravel aspects such as text drift categories, detection types, model update mechanisms, stream mining tasks addressed, and text representation methods and their update mechanisms. Furthermore, we discussed drift visualization and simulation and listed real-world datasets used in the selected papers. Finally, we brought forward a discussion on existing works in the area, also highlighting open challenges and future research directions for the community.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Computing methodologies → Artificial intelligence**;

Additional Key Words and Phrases: Concept drift, text stream mining, semantic shift, representation shift, drift detection

Authors' Contact Information: Cristiano Mesquita Garcia, cristiano.garcia@ifsc.edu.br, Instituto Federal de Santa Catarina, Caçador, Brazil and Programa de Pós-Graduação em Informática (PPGIa), Pontifícia Universidade Católica do Paraná (PUCPR), Curitiba, Brazil; Ramon Abilio, Instituto Federal de São Paulo, Capivari, Brazil and Universidade de Campinas, Limeira, Brazil, ramon.abilio@ifsp.edu.br; Alessandro Lameiras Koerich, École de Technologie Supérieure, Université du Québec, Montréal, Canada, alessandro.koerich@etsmtl.ca; Alceu de Souza Britto Jr, Programa de Pós-Graduação em Informática (PPGIa), Pontifícia Universidade Católica do Paraná (PUCPR), Curitiba, Brazil and Universidade Estadual de Ponta Grossa, Ponta Grossa, Brazil, alceu@ppgia.pucpr.br; Jean Paul Barddal, Programa de Pós-Graduação em Informática (PPGIa), Pontifícia Universidade Católica do Paraná (PUCPR), Curitiba, Brazil, jean.barddal@ppgia.pucpr.br.

ACM 2157-6912/2024/8-ART111
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Cristiano Mesquita Garcia, Ramon Abilio, Alessandro Lameiras Koerich, Alceu de Souza Britto Jr, and Jean Paul Barddal. 2024. Concept Drift Adaptation in Text Stream Mining Settings: A Systematic Review. *ACM Trans. Intell. Syst. Technol.* 37, 4, Article 111 (August 2024), 69 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Intelligent systems (IS) based on machine learning (ML) have been increasingly researched as processing power has increased and storage capacity has been cheapened. The development of frameworks and libraries, such as Weka [74] and Scikit-Learn [156], has enabled the rapid development and deployment of ML models and their applications. Moreover, Tensorflow [1], Keras [34], PyTorch [154] and HuggingFace [208] are more contemporary enablers that are related to deep learning models and generally rely on graphic processing units (GPUs) to expedite the training process. Therefore, there has been an increase in the development of ML applications, such as credit scoring [15], emotion recognition [44], and cryptocurrency pricing prediction [63].

Software, sensors, processes, and humans generate data, the primary raw resource for developing ML models. Humans, in particular, produce a considerable amount of unstructured data on the Internet, especially on social media, where users upload pictures and post opinions regarding anything, including products, artists, and politicians. Therefore, social networks have been considered a low-cost, rapid source of information, with the collected data utilized for election prediction [30, 49, 199], stance analysis [27], event detection [192], *etc.*

Texts are unstructured data. Most ML approaches expect numbers as input parameters, so texts cannot be directly used as input for ML methods. To overcome the aforementioned limitation, text must be processed, cleaned, sometimes standardized, and converted to fixed-size numerical vector representations. The conversion from unstructured to structured data is also known as feature extraction [5, 197]. Recent advances in natural language processing (NLP) advances have simplified text-based real-life applications. It is worth mentioning Word2Vec [135], which is a neural network-based approach for generating word embeddings (vector representation), and BERT [46], a bidirectional transformer-based modeling architecture, that can be applied in tasks such as sentiment analysis, and spam detection. One advantage of the aforementioned methods is their reuse capability. Several pre-trained models are available on the Internet in specialized hubs such as HuggingFace[1]. A pre-trained model can aid in extracting features from text and use them as input for a classifier, *e.g.*, a sentiment classifier. The time necessary to develop the final ML model can be drastically reduced if tailoring a representation-learning model from scratch is not required. For instance, using pre-trained models is a common approach when the target application is aligned with the context in which the pre-trained model was built. Additionally, in the case of using the pre-trained model in a transfer learning fashion, it has been shown possible to fine-tune the ML model, the representation model, or both, depending on the computational resources available and the expected outcomes of the intelligent system.

Although the aforementioned approaches were initially designed for batch learning, it is possible to use pre-trained models to extract features in data stream scenarios. Data streams are considered a collection of sequential data that comes consecutively, or in small batches, in a timely order [21]. Thus, for ML models in data streams, there are challenges such as learning from the data the instant it arrives, adapting the model in case of pattern change, and keeping it concise. *Text streams* represent a continuous flow of textual data, such as social media updates, news articles, customer reviews, or online discussions. Several social networks and news agencies provide application programming interfaces (API) that function as a text stream. X [2] (former Twitter) is an example of a

[1]https://huggingface.co/models
[2]https://x.com/

social media platform that offers API access to its data. Conversely, Massive Online Analysis (MOA) [22] and RiverML [138] have been enablers of experimentation and development of methods for stream mining, despite not targeting textual data specifically.

A data pattern change is commonly referred to as *concept drift* in the scientific machine learning literature. *Concept drift* is a phenomenon that occurs in data subject to non-stationary processes [21, 62]. In real life, for example, changes may occur in temperature or customer purchasing patterns across given analyzed periods. Concept drift imposes several difficulties for ML models, *e.g.*, if concept drifts are not captured and managed by the model, its performance will degrade over time. It can be even more challenging for ML approaches that require the processing of text streams due to the constraints inherent to streaming learning settings, such as the speed of the stream. In text streams, concept drift occurs when the underlying patterns and relationships within the textual data shift, making previously learned models or approaches ineffective. Concept drift in text streams arises from the dynamic and evolving nature of language and its data sources, where trends, contexts, and sentiments change over time. Therefore, understanding and addressing concept drift is crucial for maintaining the accuracy, relevance, and ethical integrity of ML models for text stream processing.

In addition to concept drift, a specialized type of drift can emerge in texts: *semantic shift*. Semantic shift also referred to as *semantic change* [43], regards changes in word meanings over time [25]. These changes can affect not only the words themselves but also their entire context, which can influence the performance in downstream tasks such as classification, for example. Another interesting aspect regarding text is that they cannot be treated in its raw form, thus requiring processing to be represented in a numeric format so that the drifts/semantic shifts are to be detected. Even though some authors argue the existence of different types of drifts/semantic shifts in real-world datasets, *e.g.*, Heusinger et al. [84], these drifts are difficult to label. This is supported by one of the findings reported in this paper, in which only one dataset had drifts labeled [65] and corresponded to sentiment drift events identified during a soccer match.

Processing text and learning in stream scenarios is challenging due to the requirements for ML models to function effectively in such scenarios. The requirements include: (i) learning from the data as it arrives; (ii) discarding the data after learning from it; (iii) performing all operations in a single-pass fashion [21, 61]. In addition, NLP-related activities can be challenging in stream scenarios, such as maintaining an updated and concise vocabulary and updating representations when possible. Therefore, text stream scenarios are even more restrictive since the NLP-related activities must also be designed to ideally perform one-pass operations.

Motivated by the challenges and constraints of text streams, the existence of concept drift, and the characteristic of intelligent systems to learn incrementally in these scenarios, this study offers a systematic review regarding concept drift adaptation in text streams. Fig. 1 provides our scope for this review, in which we target the intersection of text streams, concept drift detection/adaptation, and works that introduce novel incremental and adaptive learning methods for such scenarios. In other words, this systematic review unravels the most common approaches to managing concept drift, updating the model to recover from concept drifts, text representation methods, datasets, and applications in challenging scenarios such as text streams. This work is organized as follows: Section 2 introduces data stream mining and presents the aspects of concept drift, semantic shift, and concept drift detectors. Section 3 details the protocol for this systematic review. Section 4 presents and discusses the results. Section 5 lists and describes the available real-world datasets. Section 6 discusses concept drift visualization and drift simulation settings. Section 7 concludes the study and emphasizes open challenges and future directions. To facilitate the reader to follow the acronyms, we added Section A to list and explain the acronyms present in this manuscript, functioning as a glossary.

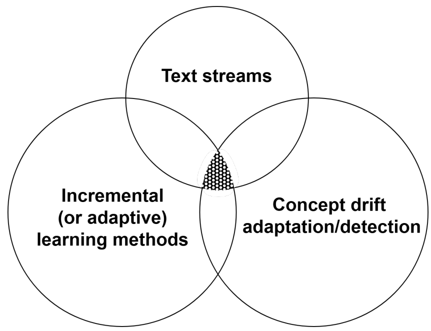

Fig. 1. Intersection of subjects of interest in this review. We are mainly interested in the papers on the intersection (hatched area) of these three subjects.

2 BACKGROUND

According to Bifet et al. [21], "data streams are an algorithmic abstraction to support real-time analytics". Data streams are data items arriving continuously and are temporally ordered. In traditional data mining, it is compulsory to have data collection so that the ML model can learn patterns from it and perform the desired task. However, there are several constraints in Data Stream Mining (DSM). Because the data arrives continuously and streams are potentially infinite, storing the data to posteriorly learn from can become unfeasible.

Thus, the ML model must learn from the data and discard it within a short period [21]. In addition, Bifet et al. [21] mentioned that there are two main challenges for ML models when handling data streams: (a) learning from the data the instant it arrives and (b) being able to adapt in case the data evolves. Since these challenges must be addressed quickly and consume minimal processing, the outcome is an approximate model rather than a precise model. Furthermore, the same authors highlighted that since data streams are continuously arriving rapidly and can be infinite, the data generation process may undergo significant changes over time, reflecting the data distribution. These changes, namely *concept drift*, increase the challenges of managing data and text streams.

In this paper, we define a text $\boldsymbol{X}$ as a sequence of arbitrary length composed of tokens $\boldsymbol{t}$. These tokens can include words (lexical units), punctuation marks, subwords, and other elements. Thus, we represent a text as $\boldsymbol{X} = (\langle \boldsymbol{t}_i \rangle \mid i = 1, ..., n)$, where n denotes the total number of tokens. Typically, these tokens are organized in a specific order that adheres to the rules of natural language, allowing them to convey meaningful information. Initially, texts were primarily used for communication between humans. More recently, they have also served as logs for communication from systems to humans. Furthermore, in the last developments, text facilitates interactions from humans to systems, exemplified by chatbots and large language models like ChatGPT.

Concept drift in text streams can be formally defined as follows. Let a text stream $\boldsymbol{T} = (\langle \boldsymbol{X}_j \rangle \mid j = 1, ...)$ represent a potentially infinite sequence of input texts $\boldsymbol{X}_j$, where j denotes the text index. In the context of a classification task involving textual data streams, each text may be associated with a label y, resulting in a sequence of pairs $(\boldsymbol{X}, y)$, or more formally, $\boldsymbol{T} = (\langle \boldsymbol{X}_j,\ y_j \rangle \mid j = 1, ...)$. According to Gama et al. [62] concept drift is said to occur if

$$\exists \boldsymbol{X} : p_{t_0}(\boldsymbol{X}, y) \neq p_{t_1}(\boldsymbol{X}, y), \tag{1}$$

where $p_{t_0}(\boldsymbol{X}, y)$ represents the joint distribution of $\boldsymbol{X}$ and the label y at time t_0. It is important to note that $\boldsymbol{X}$ can be represented numerically as a dense vector or through word frequencies and co-occurrences over time. Such numerical representations facilitate the extraction of statistics that are essential for detecting concept drift and semantic shifts.

According to Gama et al. [62], "data is expected to evolve". Thus, the data distribution can change as time passes. These changes are referred to as *concept drift*. The machine learning literature highlights two primary types of drifts in data distribution: (i) *Real concept drift*, where the relationship between X (input data) and y (class) changes, and (ii) *Virtual concept drift*, where the data distribution in X changes, but $p(y|X)$ does not change, meaning that the boundaries are unchanged. Real concept drift can occur even if the data distribution in X does not change. Across scientific and industry communities, virtual drifts may also be referred to as *covariate shift* [54], or *data drift* [168]. Another type of drift is the *label shift*, which corresponds to changes in label distribution, compared to reference data [215]. Fig. 2 shows the aforementioned types of drifts.

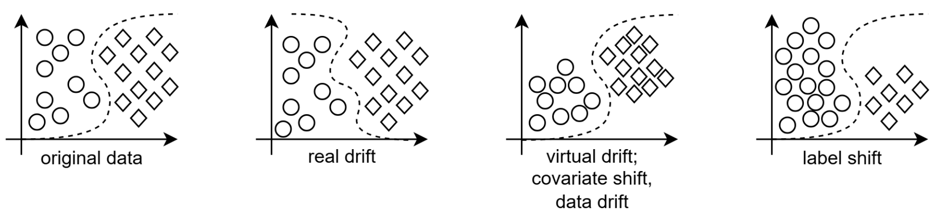

Fig. 2. Types of concept drift. Adapted from Gama et al. [62]. Each marker, *i.e.*, circle, and diamond, represents an arbitrary class/label. Dashed lines correspond to the border between regions of classes.

In addition, Gama et al. [62] highlighted four different types of concept drift dynamics over time. The four categories are as follows: (a) *abrupt*, where the data distribution changes from t_i to t_{i+1}; (b) *incremental*, where the data distribution changes from t_i to $t_{i+\Delta}$, where $\Delta > 1$; (c) *gradual*, where the data distribution switches between different means until remaining in the last distribution; and lastly (d) *reoccurring*, where the data distribution changes and later, switches back to the first data distribution observed. Fig. 3 depicts the concept drift types concerning the dynamics over time.

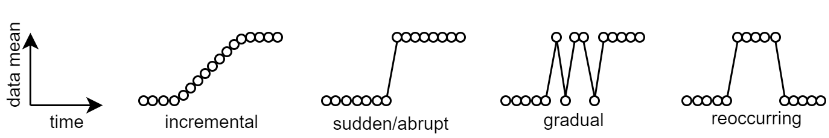

Fig. 3. Dynamics of concept drift over time. Adapted from Gama et al. [62].

When it comes to text, different aspects of drifts may emerge, such as a word gaining or losing meanings over time, known as *semantic shift* [25], sometimes referred to as *semantic change* [43]. In this paper, we use only the terminology *semantic shift* for the sake of simplicity. According to Kutuzov et al. [112], *semantic shift* constitutes "the evolution of word meaning over time". Fig. 4 depicts examples of semantic shifts that occurred across decades and centuries [77]. Fig. 4 was generated using Word2Vec representations [135] and t-SNE [201] for dimensionality reduction, according to Hamilton et al. [77]. In the 1850s, *awful* had a positive connotation, as depicted in Fig. 4 (c). The surrounding words, *e.g.*, *majestic* and *solemn*, corroborated the previous statement. However, in the 1900s, the word *awful* shifted to a negative connotation due to its proximity to the words *terrible* and *horrible*. More precisely, *semantic shift* has been studied across the years. de Sá et al. [43] overviewed the subject and characterized semantic changes considering the aspects of *dimension*, *relation*, and *orientation*. In the case of dimension, de Sá et al. [43] considered broadening, *i.e.*, gaining new meanings, and narrowing, *i.e.*, becoming more specific or losing previous meanings. Considering the relation, de Sá et al. [43] mentioned metaphorization and metonymization, which occurs, according to the authors, "when a word takes on a new meaning that, to some extent, inherits qualities from its original meaning through a figurative relationship the speaker aims to convey". Finally, changes in orientation regard the connotation of a new meaning, *i.e.*, towards positive (amelioration) or negative (pejoration).

Several works have been proposed to measure the evolution of a word's meaning over time [17, 47, 180]. Some papers provide semantic shift detection methods that measure the cosine distance between word embeddings in a period and the word embeddings from the same words in a previous period [7]. If the distance exceeds a certain threshold, it is deemed a semantic shift to have occurred. Other approaches may use embedding alignment across time slices, such as orthogonal Procrustes [76] and compass alignment [17]. However, traditional ML methods mostly address semantic shift detection, *i.e.*, outside of the streaming context. It means that for most of the approaches, there are no constraints on processing and storage.

Approaches capable of handling text streaming become relevant in a world where enormous quantities of data are generated each second. Therefore, this review focuses exclusively on approaches applied to text stream scenarios. In addition, despite works that depict semantic shifts over long periods, works such as Garcia et al. [65], Stewart et al. [189] demonstrated that semantic shifts may occur not only in decades or centuries but also in a shorter period, *e.g.*, weeks or even a few minutes/hours.

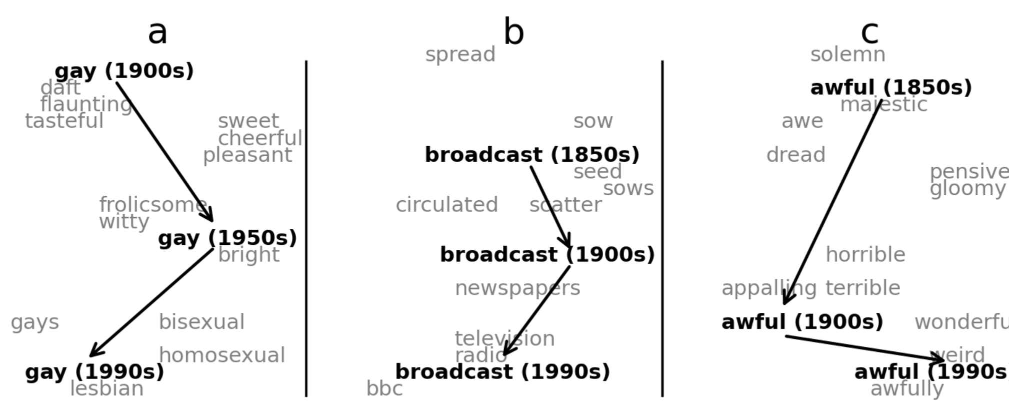

Fig. 4. Semantic shift across several decades or centuries. Adapted from [77].

Concept drift detectors are methods used for detecting changes in data distribution, and they can be beneficial in performing both concept drift and semantic shift detection. These types of detectors were initially developed in statistics. However, there is no guarantee that such methods would work specifically in streaming scenarios because some may not work in a one-pass fashion [21]. Gama et al. [62] categorized concept drift detection methods into four classes: (i) *sequential analysis*; (ii) *control charts*; (iii) *monitoring two distributions*; and (iv) *context-based methods*, which are also called *heuristic methods*. *Sequential analysis* corresponds to a scenario in which two subsets of data are generated sequentially by processes bound to different unknown distributions, *e.g.*, P_0 and P_1. According to Gama et al. [62], "when the underlying distribution changes from P_0 to P_1 at point w, the probability of observing certain subsequences under P_1 is expected to be *significantly* higher than that under P_0". It signifies that a statistical test, for example, can be used to detect this change. Two primary representatives of this category are the cumulative sum (CUSUM) test [153] and the Page-Hinkley test [153], which is a variant of the CUSUM test [21, 62].

The second category proposed by Gama et al. [62] is *control charts*, also known as *statistical process control* (SPC). Control charts correspond to "standard statistical techniques to monitor and control the quality of a product during continuous manufacturing" [62]. In this case, the data are received over time and are input to the model, and the model's error is used to determine the states of the system. The system states are as follows: (i) *in-control*, which indicates that the system is stable; (ii) *drift detection*, which signifies the error increased significantly, compared to the historical error; and (iii) *warning*, which indicates the error increased but was insufficient to raise a detection. Drift and warning are generally associated with a statistical confidence of 99% and 95%, respectively. An example of this category is the exponentially weighted moving average (EWMA) [178]. The third category regards *monitoring two distributions*. Methods in this category, according to Gama et al. [62], "typically use a fixed reference window that summarizes the past information and a sliding detection window over the most recent examples". In this scenario, a drift is considered to have occurred if the distributions of the windows are statistically different. An example of a method that embeds a concept drift detector from this category is the Very Fast Decision Tree (VFDT) [60]. An actual concept drift detector that fits in this category is the Adaptive Windowing (ADWIN) [20]. ADWIN is a distribution-free concept drift detector suited for detecting drifts in real-valued or bits streams [21]. It maintains a window with the most recent items, from which subwindows are compared. If these subwindows exhibit different means above a threshold based on Hoeffding's bounds, a drift is flagged [62]. ADWIN is computationally more expensive in time and memory than *sequential analysis* detectors; however, it is simpler to use because the user does not need to specify a cutoff parameter [21, 62]. In addition, ADWIN provides more precise change points [62].

The last category, *i.e.*, *context-based*, regards specific approaches that use characteristics intrinsic to ML methods to perform drift detection or adaptation. For instance, Garcia et al. [64], Leite et al. [116], Soares et al. [187] proposed a method that balances incremental learning and forgetting using fuzzy granular computation. Whenever a new instance is inputted, the existing granules, *i.e.*, groups that share similar properties, have their (either complete or partial, whenever there are missing attributes in the new instance) similarity with the newly seen instance calculated. The new instance is assigned to the chosen granule if the similarity exceeds a certain threshold. However, if no granule can match the newly seen instance, *i.e.*, a drift occurs, and a new granule is created to accommodate the new instance. In addition, a pre-defined parameter controls the periods of verifying stale granules, which can be deleted to maintain the model's conciseness.

The common metrics used to evaluate and compare concept drift detection methods, according to Bifet et al. [21], are as follows: (i) mean time between false alarms (*MTFA*), which assesses the frequency with which a method raises false alarms; (ii) false alarms rate (*FAR*), which is the inverse of *MTFA*; (iii) mean time to detection (*MTD*), which assesses how quickly the method detects and responds to drift once it occurs; (iv) missing detection rate (*MDR*), which determines how frequently the method fails to warn when drift occurs; and (v) average run length (*ARL*), which is the time it takes to raise the alarm once a drift occurs [21]. *ARL* integrates *MTD* and *MTFA* [21]. Additional metrics, such as Mean Time Rate (*MTR*) [19, 207], may emerge in the literature; however, the primary focus is on missing drifts, hits, time/iterations until detecting an actual drift, or a combination of such factors. *MTR*, for instance, is analogous to *ARL* [207].

Typically, concept drift detectors are coupled to traditional or online ML systems by receiving the hits and errors of prediction. These concept drift detectors have two levels of alarms: *warning* and *drift*. The most straightforward use is when a warning alarm is issued. Either the input data are buffered, or a new ML model is trained such that when the drift alert occurs, a new model (trained using data from the buffer) replaces the outdated one. This learning strategy is called *background learning* [71]. Thus, the idea is to maintain an updated model based on the most recent/frequent data.

3 SYSTEMATIC REVIEW PROTOCOL

This review followed the guideline proposed by Kitchenham and Charters [103], which comprises three steps: (i) planning the review, (ii) conducting the review, and (iii) reporting the review. *Planning the review* includes identifying the need for the review and formulating the research questions. In *conducting the review*, we select primary studies and perform data extraction and synthesis. Finally, in *reporting the review*, it is expected to disclose the results and findings. In this work, we used five sources of studies: IEEEXplore[3], Science Direct[4], ACM Digital Library[5], Springer Link[6], and Scopus[7]. We devised a series of four questions to guide our research. The primary question, *RQ*1, takes precedence, while the remaining questions are derived from *RQ*1. Table 1 displays our research questions for reference.

Table 1. Research questions used in this work.

ID	Research Questions
RQ1	"How to handle concept drift using ML approaches having as source text streams?"
RQ2	"Which type of application is addressed?"
RQ3	"Which type of token/word/sentence representation is used in the study?"
RQ4	"Which datasets were used to evaluate the proposed approach(es)?"

The search query was developed considering *RQ*1. We also used a few synonyms to aid in developing a broadening query. The reader can discover additional information on the terms and synonyms in Table 2. *RQ*2 focuses on the applications the papers addressed when handling concept drift in textual streams. This question is crucial because it can illustrate various scenarios, the potential, and increased interest in specific problems. Besides the application, we wanted to know which ML methods are employed and how these models are updated, *e.g.*, incrementally or regularly retrained. With *RQ*3, we intended to uncover the most common approaches to representing texts (or smaller parts, such as tokens, words, and sentences). Finally, *RQ*4 pursues insights into the existence of consolidated datasets for the field and their aspects, such as the level of labeling in the dataset, *e.g.*, instance or token, the data mining task employed, *e.g.*, clustering, classification, whether the dataset contains real-world data or it is synthesized, metrics used in those data mining tasks, and whether drifts are labeled in the dataset.

We developed the query presented below using Table 2. The terminologies *semantic shift* and *representation drift* are closely related to *concept drift*, especially in the textual context. Semantic shift (or semantic change), according to Bloomfield [25], refers to "innovations which change the lexical meaning rather than the grammatical function of a form". However, according to Fu et al. [59], the representation shift in NLP relates to changes in the vector representation, when using semantic vectors as representations for word meaning. We included *social network streams* because they are the notable source of text streams produced directly by humans nowadays.

[3]https://ieeexplore.ieee.org/
[4]https://www.sciencedirect.com/
[5]https://dl.acm.org/
[6]https://link.springer.com/
[7]https://www.scopus.com/

Table 2. Table containing keywords and respective synonyms.

Keyword	Synonyms
concept drift	semantic shift, representation shift, semantic change
machine learning	-
text streams	textual streams, social network streams, Twitter streams, diachronic, text streaming
detection	-

We also used the terminology *Twitter streams*, because Twitter, *e.g.*, currently named *X*[8], is a microblog (one of the most popular) and generated around 500 million tweets (posts) per day, in 2022[9]. Furthermore, we included the term *diachronic*. When serving as an adjective for a dataset, *diachronic* refers to a dataset that contains data produced over time. The term *machine learning* was withdrawn because concept drift is mostly addressed by or in processes that use ML techniques. The query used in the search is: `(“concept drift” OR “semantic shift” OR “representation shift” OR “semantic change”) AND (“text streams” OR “textual streams” OR “textual streaming” OR “social network streams” OR “twitter streams” OR “diachronic”) AND (“detection”)`. Each source has its parameters, but we prioritized full-text search in all of them.

3.1 Inclusion and Exclusion Criteria

The inclusion and exclusion criteria used in this review are described below. It is crucial to note that we limited this review to papers published after 2018 because other previous secondary studies tackle similar problems [112, 155, 160, 196]. Kutuzov et al. [112] evaluated several papers regarding diachronic word embeddings and semantic shifts. The authors approached several aspects, such as diachronic semantic relations and the sources of diachronic data for training and testing. Tahmasebia et al. [196] developed a survey on computational approaches for lexical semantic change detection. They approached aspects such as the semantic change types and computational modeling of diachronic semantics. Patil et al. [155] also developed a survey on concept drift detection for social media. The authors provided information on datasets and the evolution of techniques over time. Periti and Montanelli [160] presented a survey on modeling lexical semantic change through modern, deep language models, including large language models, regarding aspects such as time awareness, learning scheme, language model, training language, and corpus language. The surveys/reviews from Kutuzov et al. [112], Periti and Montanelli [160], Tahmasebia et al. [196] evaluated semantic shift and diachronic aspects without concerning specifically streams and methods that respect the streaming processing constraints. Patil et al. [155] approached a similar aspect as ours; however, we provided deeper analysis on several characteristics, such as model update scheme, text representation methods, and their update schemes when available, datasets, and so on. Thus, a substantial difference between our systematic review and the aforementioned works is that we focus on papers that approach the problem of concept drift/semantic shift using text streams as a data source. Using streams as data sources requires specific approaches to overcome the stream processing constraints, as seen in Section 2. Therefore, according to Table 3, we considered the following inclusion and exclusion criteria. It is also essential to note that this review protocol was last executed on August 20, 2024.

After gathering the returned papers, each researcher screened their abstracts to flag the inclusion or exclusion of each study. Concerning divergences, the researchers agreed to read the divergent

[8] https://x.com/
[9] https://www.dsayce.com/social-media/tweets-day/

Table 3. Inclusion and Exclusion criteria used in this study.

Ref	Inclusion criteria
IC1	The study is published in journals or conference proceedings
IC2	The study is published from 2018 (inclusive)
IC3	The study presents a method for handling concept drift
IC4	The study uses text streams as data source

Ref	Exclusion criteria
EC1	The study is not primary
EC2	The study is not written in English
EC3	The study is incomplete
EC4	The study is not an article
EC5	The study is duplicated
EC6	The study does not meet the inclusion criteria

papers carefully to have confidence in their decision. We used Cohen's Kappa coefficient [131] to measure the agreement level between the researchers.

4 RESULTS AND DISCUSSION

Fig. 5 overviews the paper selection process. We collected 870 papers, considering the research query. The final calculated Cohen's Kappa coefficient reached 84.61%, which indicates a high level of agreement between the researchers. In addition, the divergences were discussed after a thorough reading of the divergent papers, and a decision was reached on their inclusion or exclusion. After removing duplicates (n=178), non-article studies (n=5), non-primary studies (n=46), and unrelated studies (n=562+31=593), we retained 48 articles for a full reading and analysis.

Considering the process depicted in Fig. 5, the reader's attention may be drawn by the high number of unrelated studies after screening the abstract. It occurred due to the query term *diachronic*, which relates to something that evolves, especially concerning language. Most approaches that handle language evolution cannot work in streaming environments (about 60% of the papers in our initial identification using the query). Therefore, we excluded those studies from our paper selection. In addition, we highlight that we are interested in approaches that handle *text streams* as data sources. It means that to be considered for our selection, the approaches must process the datasets seeking to respect the text stream constraints (see Section 2). This characteristic filtered out several papers from our selection. Furthermore, around 16% of the papers did not handle/mention drift, although the terminology was included in the keywords, as shown in Table 2.

Based on the information extracted from the selected papers using the research questions, we categorized the approaches for handling text drifts presented according to the following characteristics: (DC) text drift categories; (DD) drift detection types; (MU) model update; (TR) text representation; and (TRUS) text representation update scheme. Our proposed taxonomy is depicted in Fig. 6. In addition, Table 4 shows the selected papers according to our proposed taxonomy. Subsection 4.1 describes the main statistics of the selected papers.

The selected papers were studied in detail considering the taxonomy presented in Fig. 6. Section 4.2 describes and categorizes the types of concept drift handled in the selected papers, *i.e.*, *Drift categories*. Section 4.3 analyzes how the text-related concept drift detection is performed, *i.e.*, in a model-adaptive way or explicitly, regarding the *Drift detection* in our proposed taxonomy. Section 4.4 describes how the ML models used in the papers are updated when handling a text stream, *i.e.*, *Model update* in the taxonomy. We categorized the approaches according to the related *Stream mining tasks*, in addition to the applications and related metrics. Section 4.5 expands the information on the stream mining tasks presented in the papers. In *Text representation*, we uncovered the text representation methods used in the papers, considering embeddings, frequency-based methods, and words directly. Section 4.6 describes the text representation methods used in the papers. For *Text representation update mechanism*, we analyzed whether and how the text representations are

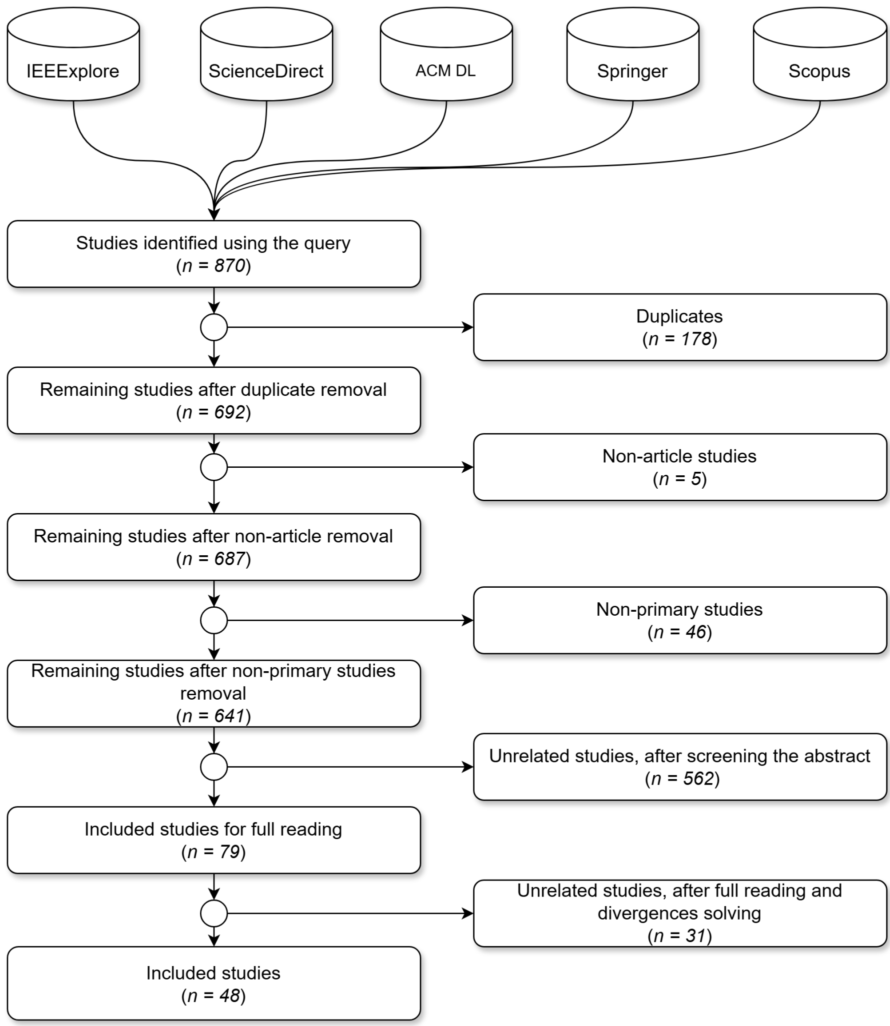

Fig. 5. Process of papers selection. Each rounded-corner rectangle on the right side corresponds to an exclusion criterion. The numbers of remaining studies after each elimination are presented on the left side.

updated over time. Section 4.7 explores the update scheme of the text representation methods. All selected methods were studied under the taxonomy's second level, *i.e.*, text drift categories, text drift detection, model update, stream mining task, text representation, and text representation update mechanism. In addition, the methods can fit more than one characteristic below the second level. Recalling the Research Questions presented in Section 3.1, RQ1, *i.e.*, *"How to handle concept drift using ML approaches having as source text streams?"*, is addressed in Sections 4.2, 4.3, and 4.4; RQ2, *i.e.*, *"Which type of application is addressed?"* is addressed in Section 4.5; RQ3, *i.e.*, *"Which type of token/word/sentence representation is used in the study?"* is approached in Section 4.6; and finally, RQ4, *i.e.*, *"Which datasets were used to evaluate the proposed approach(es)?"*, is conveyed in Section 5.

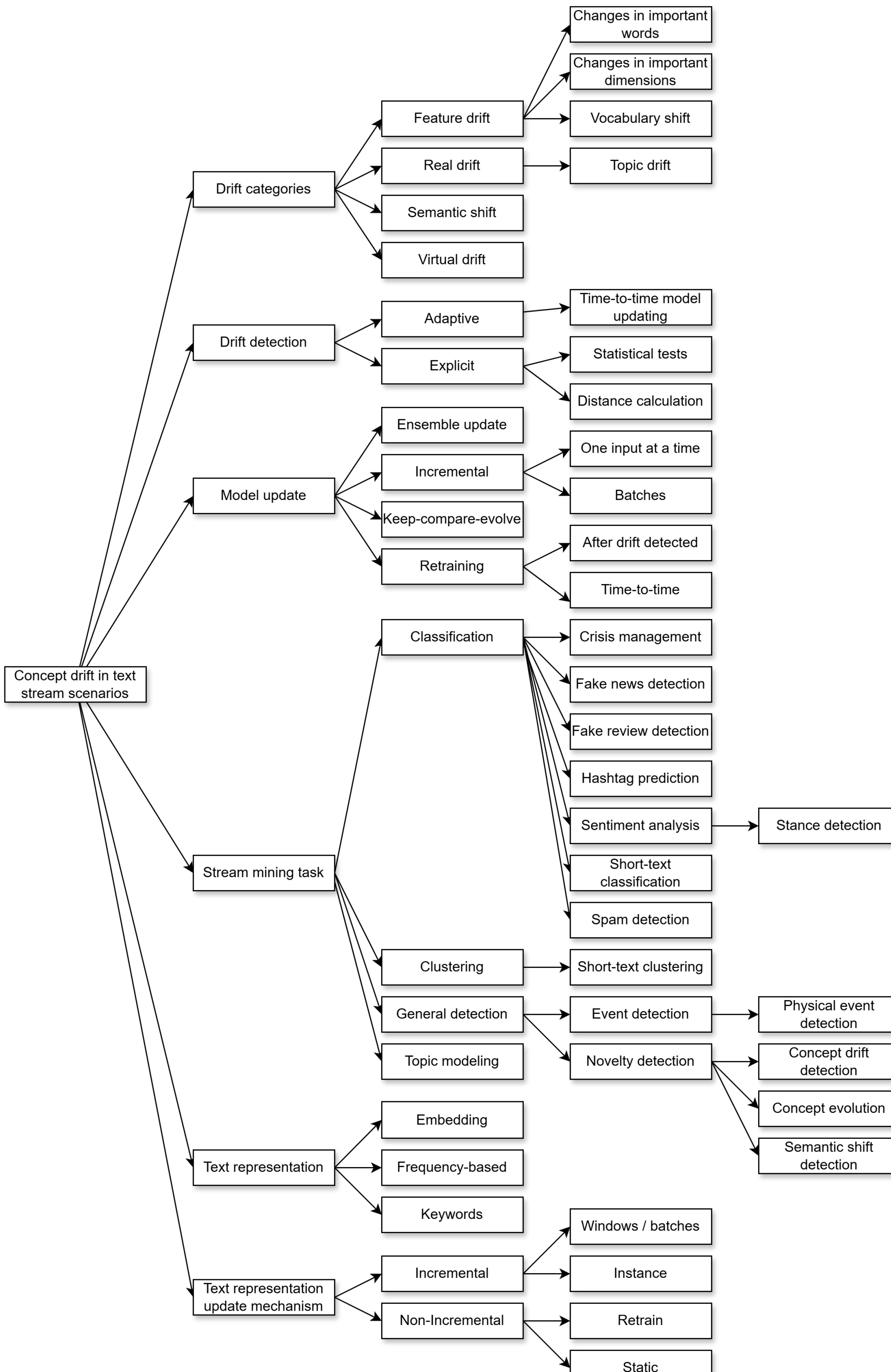

Fig. 6. A taxonomy regarding concept drift in text stream scenarios.

Table 4. Selected papers ordered by year. Acronyms are explained in the legend.

Method	(DC)	(DD)	(MU)	(SMT)	(TR)	(TRUS)
AWILDA [141]	r > td	e > st	i > o	tm	kw	n > s
OBAL [164]	r	a	i > b	class > cm	fb	n > s
CRQA [41]	r	e > st	-	class > sa, nd > cdd	-	n > s
AIS-Clus [2]	r, fd > ciw	a	i > b, i > o	clust, class, gd > ed, nd > ce	kw	n > s
- [118]	r > td	e > dc	i > b	class > stclass	fb	n > s
- [133]	fd > ciw	a, e > st[1]	i > b	class > s, class > sa	fb	n > s
MStream [211]	r > td	a	i > b	clust > stclust	fb	n > s
OurE.Drift [90]	r > td	e > dc	eu	class > stclass	fb	n > s
- [78]	r > td	e > st	i > o	class > tc	kw	n > s
- [81]	r	a	i > b	class	e	n > s
AIS-Clus [3]	r, fd > ciw	a	i > b, i > o	clust, class, gd > ed, nd > ce	kw	n > s
LITMUS-ASSED [193]	r	a	i > b	gd > ed > ped	e	n > s
LITMUS [192]	r	a	i > b	gd > ed > ped	e	n > s
DCFS [33]	fd > cid	e > st	r > ad	class > s, nd > ce	fb	n > s
LITMUS [194]	r, v	e > dc	eu	gd > ed > ped	e	n > s
ESACOD [206]	r	e > st	r > ad	class, nd > ce	e	n > s
- [39]	r	a	r > t	class > sa > sd	fb, e	n > r
- [137]	r	e > st	i > o	class > frd	fb	n > r
- [84]	r	a	i > o	class > ht	fb, e	n > s
OFSER [42]	r	a	i > o	class > s	fb	n > s
- [16]	r	a	i > b	class > sa > sd	fb, e	n > r
- [191]	r	e > dc	eu	class > stclass	fb	n > s
- [7]	r, s, fd > v	a	kce	class > ht	e	i > b
EStream [173]	r > td	a	i > o	clust > stclust	fb, e	n > s
EWNStream+ [210]	r > td	a	i > b	clust > stclust	fb	n > s
GCTM [202]	r > td	a	i > b	tm	e	n > s
BSP [144]	r > td	a	i > b	tm	e	n > s
- [85]	r	e > st	i > b	class > ht	e, fb	n > s
DDAW [170]	r, v	e > dc	eu	class > sa	-	-
GOWSeqStream [203]	r > td	a	i > b	clust > stclust	e	n > s
GDWE [127]	r, s	a	i > b	class	e	i > b
- [29]	r	a	i > o	class > sa	fb	i > inst
- [27]	r	a	i > b, r > t	class > stclass	fb	n > r
SMAFED [105]	r	a	i > b	class, clust, gd > ed	e	n > s
WIDID [159]	s	e > dc	i > b	nd > ssd	e	n > s
- [119]	r > td	e > dc	r > td	class > stclass	e	n > s
FFCA index [55]	r	e > dc	-	class > fnd	fb	n > s
TSDA-BERT [195]	r	e > dc	r > ad	class > sa	e	n > r
DDAW [169]	r, v	e > dc	eu	class > sa	f	-
textClust [10]	r	a	i > b, i > o	clust	fb	i > b
- [65]	r	e	-	stclass	kw, fb	i > inst
- [66]	r > td	dc	i	hd	kw	n > s
OSMTS [110]	r	dc	i	class > ml	kw, e	n > s
TCR-M [204]	r > td	dc	r > t	class	fb	n > s
- [188]	r, fd > ciw	dc	r > t	clust	e	i > inst
- [48]	r	a	i, r	class	fb	n > r
AE [168]	v	e	-	cdd	a	n > s
AdaNEN [69]	r	a	i	class	e	n > s

Legends: > : a level down in the taxonomy. **(DC) Drift category** → *r*: real drift; *td*: topic drift; *fd*: feature drift; *cid*: changes in important dimensions; *ciw*: changes in important words; *v*: vocabulary shift; *vd*: virtual drift; *s*: semantic shift. **(DD) Drift detection method** → *a*: adaptive; *e*: explicit; *st*: statistical tests; *dc*: distance calculation. **(MU) Model update** → *eu*: ensemble update; *i*: incremental; *o*: one input at a time; *b*: batches; *kce*: keep-compare-evolve; *r*: retraining; *ad*: after drift detected; *t*: time-to-time. **(SMT) Stream mining task** → *class*: classification; *cm*: crisis management; *fnd*: fake news detection; *frd*: fake review detection; *ht*: hashtag prediction; *sa*: sentiment analysis; *sd*: stance detection; *s*: spam detection; *stclass*: short-text classification; *clust*: clustering; *stclust*: short-text clustering; *gd*: general detection; *ed*: event detection; *ped*: physical event detection; *nd*: novelty detection; *cdd*: concept drift detection; *ce*: concept evolution; *ssd*: semantic shift detection; *tm*: topic modeling. **(TR) Text representation** → *e*: embedding; *fb*: frequency-based; *kw*: keywords. **(TRUS) Text representation update scheme** → *i*: incremental; *b*: batch; *inst*: instance; *n*: none; *r*: retrain; *s*: static.

[1]: one version uses ADWIN to explicitly detect feature drift.

4.1 Main Statistics

We unraveled statistics on the selected papers regarding (i) the sources, (ii) years of publication, and (iii) venues of publication. Table 5 shows the number of selected papers by source. Scopus provided 37.5% of the selected papers for this work. We noted a steady interest across the years in streaming text applications susceptible to concept drift in its various possibilities. Considering the limited time range in our search, *i.e.*, between 2018 and August 2024, we collected the respective number of papers: (2018) 10 papers; (2019) seven papers; (2020) two papers; (2021) six papers; (2022) 11 papers; (2023) eight papers; and (2024) four papers. Considering the characteristics of the papers across the years, we cannot infer a trend. We hypothesized that this behavior occurred because the research area is still incipient.

Table 5. Number of selected papers according to the source.

Source	Selected papers
ACM Digital Library	5
IEEE Xplore	8
Science Direct	8
Scopus	18
Springer Link	9
Total	48

Table 6 shows the venues that contributed the most to our search. The journal Expert Systems with Applications published four papers, followed by IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS) with three papers, ACM SIGKDD, Evolving Systems, International Joint Conference on Artificial Intelligence (IJCAI), International Joint Conference on Neural Networks (IJCNN), and Neurocomputing, each with two papers.

4.2 Drift Categories

Considering the categories of concept drift in text stream settings, we arranged them into (i) *Feature drift*; (ii) *Real drif*t; (iii) *Semantic shift*; and (iv) *Virtual drift*. Fig. 7 depicts the arrangement.

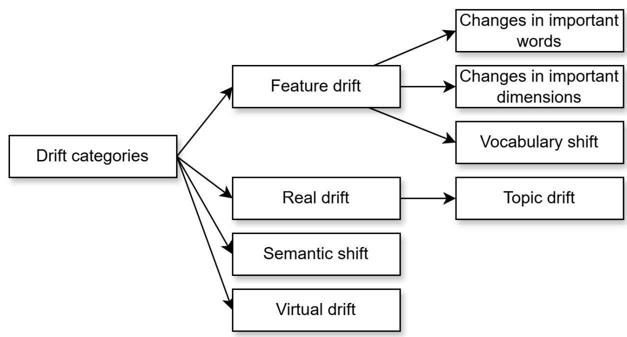

Fig. 7. Drift categories.

Table 6. Venues where the selected papers were published.

Venues	Appearances
Expert Systems with Applications	4
IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS)	3
ACM SIGKDD International Conference on Knowledge Discovery and Data Mining	2
Evolving Systems	2
International Joint Conference on Artificial Intelligence (IJCAI)	2
International Joint Conference on Neural Networks (IJCNN)	2
Neurocomputing	2
ACM International Conference on Distributed and Event-Based Systems	1
ACM Symposium on Document Engineering	1
ACM Transactions on Knowledge Discovery from Data	1
Annual Meeting of the Association for Computational Linguistics: Industry Track	1
Applied Intelligence	1
Asian Conference on Intelligent Information and Database Systems	1
Brazilian Conference on Intelligent Systems (BRACIS)	1
Chaos: An Interdisciplinary Journal of Nonlinear Science	1
Cognitive Computation	1
Computer Systems Science and Engineering	1
Computers, Materials and Continua	1
IEEE Access	1
IEEE Transactions on Big Data	1
IEEE Transactions on Cybernetics	1
IEEE Transactions on Systems, Man, and Cybernetics: Systems	1
International Conference of Reliable Information and Communication Technology	1
International Conference on Collaboration and Internet Computing (CIC)	1
International Conference on Computational Collective Intelligence	1
International Conference on Knowledge-Based and Intelligent Information & Engineering Systems	1
International Conference on Information and Knowledge Management	1
International Conference on Machine Learning and Applications (ICMLA)	1
International Journal of Computer Science (IAENG)	1
International Journal of Information Technology and Decision Making	1
International Workshop on Computational Approaches to Historical Language Change	1
Journal of Big Data	1
Knowledge and Information Systems	1
Neural Computing and Applications	1
Pattern Recognition Letters	1
Technological Forecasting and Social Change	1
Vietnam Journal of Computer Science	1
World Congress on Services	1

4.2.1 Feature drift. Feature drift considers the changes in the importance of features, signifying that, over time, a subset of features may become necessary for an ML model, while other subsets may become obsolete [14]. It constitutes a challenge for an ML model because incrementally defining the best feature set over time can be complex. In addition, the dependent ML model can have its performance degraded over time if the selected feature set is inadequate.

Considering the subcategories of *feature drift* depicted in Fig. 7, we describe the *Changes in important words*, *Changes in important dimensions*, and *Vocabulary shift*. Different types of features regarding text-related tasks may be considered in ML approaches. For instance, one approach is to consider the texts split into tokens or use direct techniques such as bag-of-words or TF-IDF. When considering 1-gram, *e.g.*, a single word/token, these techniques resort to counting tokens and measuring their overall importance, respectively. However, these techniques can be used in n-grams, leveraging subsequent words. In the papers studied in this work, two papers leverage 2-grams (bigrams): Rakib et al. [173] and Assenmacher and Trautmann [10].

Other approaches regarded the numerical transformation of texts, such as Word2Vec [134]. Since methods such as bag-of-words and TF-IDF, in 1-gram fashion, can be directly related to specific words (or tokens), and the changes in those words are regarded in this study as *Changes in important words*. However, changes in numerical representations without direct relation to words or tokens are regarded as *Changes in important dimensions*. For instance, in the case of bag-of-words and TF-IDF, each column represents a given word/token. If a word from an arbitrary point in the text stream starts appearing more (or less) than a previous point, this change would be considered *Changes in important words*. On the other hand, if we leverage a Word2Vec representation, we cannot link directly to a word since the representation learning process is based on semantic connections, using surrounding words to predict a target word, *i.e.*, Continuous bag-of-words (CBOW), or vice-versa (Skip-gram). In the case of Word2Vec, any change in dimensions would correspond to *Changes in important dimensions*.

Finally, we considered *vocabulary shift* as one type of feature drift. Vocabulary shift [7] ponders the changes of words in a vocabulary maintained by the approach as a type of text drift. Different from the aforementioned subtypes of feature drift, vocabulary shift considers the changes, *i.e.*, addition or removal of items, in the internal structure that stores the tokens. Amba Hombaiah et al. [7] compared vocabularies in year-timed slices, measuring changes between vocabularies from different years.

Amba Hombaiah et al. [7], Chamby-Diaz et al. [33], and Melidis et al. [133] addressed one of these aforementioned categories of feature drift directly. Melidis et al. [133] proposed an ensemble-based method for predicting feature values in the next time point. Considering this case, the work was categorized as *Changes in important words* because their method used a sketching mechanism to retain essential words in a fixed-size feature space, according to their occurrence count. In one version the authors presented, they utilized ADWIN [20] to evaluate a significant decrease in word usage to decide when to remove it from the sketch.

Chamby-Diaz et al. [33] proposed a feature selection method based on correlation suitable for data streams, categorized as *Changes in important dimensions*. Although the method was not developed specifically for use on text streams, the authors demonstrated its use on a text-related dataset, *i.e.*, a spam dataset. Their method retained a covariance matrix coupled to a concept drift detector. Whenever it received a warning signal, the covariance matrix was incrementally updated. When the concept drift detector triggered a drift signal, a one-pass algorithm computed feature-feature and feature-class correlations. Subsequently, a new Naive Bayes model was trained based on the new feature subset, which was chosen according to the merit of each feature subset from the correlation-based feature selection method (CFS) [75].

Unlike prior works, Amba Hombaiah et al. [7] used *vocabulary shift* to estimate the changes in the usage of tokens across several years, *i.e.*, between 2013 and 2019. The authors proposed sampling methods for updating BERT models [46] to maintain the models' usefulness in text-streaming scenarios. Initially, the authors emphasized that "vocabulary is the foundation of language models". However, vocabularies can contain different types of representation, such as complete words and sub-word segments, *e.g.*, wordpiece [46]. The authors analyzed the vocabulary shift considering the 40,000 most frequent tokens, accounting for hashtags and wordpieces. Regarding hashtags in 2013 and 2019, the vocabulary shift was 78.31%, while for wordpieces in the same period, the shift was 38.47%. The authors argued that these results and their analysis justify the development of such an incremental method proposed by them. Furthermore, the authors stated that although larger vocabularies may lessen the vocabulary shift, they were more computationally costly and, therefore, potentially infeasible for real-world scenarios.

4.2.2 *Real drift.* We considered *real drift* according to the definition in [62], which is changes in $p(y|X)$ that can occur with or without changes in $p(X)$. Considering this case, X regards the input features, while y corresponds to the class, and p is the probability. Real drift in a classification task refers to the change in the classes' boundaries, which may be accompanied by changes in the data distribution in X. In this work, few papers handle different types of real concept drift, *e.g.*, *sentiment drift*. However, because they regarded changes in $p(y|X)$, these papers were categorized as real drift.

This study considered *topic drifts* as an extension of *real drifts*. In the literature, topic drifts are encountered in applications regarding topic modeling, topic labeling, and short-text classification. Thus, a topic could drift by the change of either text labeled as a particular topic, *i.e.*, $p(y|X)$), or by the change of a topic distribution in the stream, *i.e.*, $p(X)$, or both simultaneously. In addition, it is common to use methods based on Latent Dirichlet Allocation (LDA) in short-text-related applications.

A significant number of papers regarded exclusively *real drifts* [2, 7, 10, 16, 27, 29, 39, 41, 42, 48, 55, 65, 66, 69, 81, 84, 85, 105, 110, 137, 164, 170, 188, 191–193, 195, 206]. Most commonly, methods in this category either: (i) used concept drift detectors to detect drift and trigger the model update or (ii) updated the model regularly.

Suprem and Pu [193], Suprem et al. [192], and Suprem and Pu [194] presented from multiple perspectives a system for detecting physical events with emphasis on landslides, *i.e.*, the sudden mass of rock and earth movements downwards steep slopes. They combined data from social media (which is voluminous but not so trustworthy) and governmental reports (scarce but trustworthy) to train a model for landslide detection. The authors argued that the terminology *landslide* can suffer concept drift because of its use in different contexts, such as politics. In their case, the model was updated regularly, using the governmental reports as ground truth. However, Mohawesh et al. [137] and Heusinger et al. [85] utilized concept drift detectors to detect drifts explicitly. Mohawesh et al. [137] used ADWIN [20], DDM [61], EDDM [12], and Page Hinkley [153, 184], while evaluating fake reviews detection. The authors claimed that fake reviews could lead customers to make poor decisions. Also, it is an adversarial problem: once models become better at detecting fake reviews, the unlawful reviewers change patterns over time to overcome the models. The adversarial aspect of this problem results in concept drift, which can cause the models' performance to degrade over time.

Susi and Shanthi [195] proposed a complete system for tweet collection, automated training data generation, and BERT (re)training for sentiment prediction and adaptation to sentiment drift, namely Twitter Sentiment Drift Analysis - BERT (TSDA-BERT). The authors used Apache Kafka[10] to simulate the Twitter stream. A BERT model had a three-layer dense network on top that performed the classification. Since the sentiment drift is verified using the predictions, we categorized this paper in the *real* drift category.

Assenmacher and Trautmann [10] proposed a 2-phase online method for textual clustering, namely textClust. This method leveraged TF-IDF to decide the proximity of incoming text to microclusters. In addition, the authors took advantage of unigram and bigram representations and used cosine similarity to evaluate the most suitable cluster to include the incoming text when possible. Over time, in the offline phase, the method could maintain the model concisely by merging similar clusters and removing outdated ones. To define the outdated clusters, the authors used a fading factor for the cluster weights. The authors mentioned that the fading factor helps the model handle concept drift.

[10] https://kafka.apache.org/

Garcia et al. [65] performed an experiment to detect changes in sentiment in tweets regarding a soccer match using drift detectors and a lexicon-based classifier. The authors collected tweets during a soccer match between a Brazilian and a Chilean team during the Sudamericana Cup. The context comprised two legs: the first leg, Internacional (the Brazilian soccer team) lost the match by 2-0. During the week between the matches, online influencers created an atmosphere to encourage Internacional to reverse the score. However, Internacional conceded a goal for Colo-Colo (the Chilean soccer team). The system could detect the average sentiment regarding the Internacional's supporters. However, during the match, Internacional scored three goals, the average sentiment became positive, and the sentiment changes could also be detected by the system using drift detectors.

Kumar et al. [110] presented an incremental semi-supervised method for multilabel text streams named OSMTS. Their method used the initial part of the stream to create the first micro-cluster structure, and from that, the incremental classification occurred. In addition, the method was capable of keeping itself concise by removing stale micro-clusters using an aging scheme and merging micro-clusters when they are similar enough. The micro-cluster used in this paper stores eight pieces of information, *e.g.*, number of documents, word frequencies, the sum of word frequencies, label, decay weight, last update timestamp, and the timestamp of arriving words. An interesting part of the method was that it leveraged the relationship between labels, which made sense for a multilabel scenario. The authors implemented their approach on MOA and evaluated it using nine datasets. In addition, the authors compared it to 12 other methods, obtaining the best results in most datasets having only 20% of the data. The authors evaluated their approach in terms of hamming loss, example-based accuracy, and micro-average recall. In addition, compared to other approaches, their method was conservative regarding memory.

Another significant number of papers approached the *Topic drift* problem [78, 90, 118, 119, 141, 144, 173, 202–204, 210, 211]. Topic drift primarily refers to short-text-related tasks, which commonly require additional steps to provide satisfying results, *e.g.*, data enrichment step or use of statistical information of the application context. Li et al. [118] proposed a method for short-text classification using feature space extension. Probase [209], an open semantic network, was used for the extension. According to Li et al. [118], Probase was selected by the availability of several super-concepts. It means that, in order to enrich a short text, they could obtain more information from Probase, *e.g.*, super-concepts(Apple) = [company, tech giant, large company, manufacturer], and add it to the short text. Rakib et al. [173] developed EStream, a method for efficient short-text clustering. Their approach used lexical, *e.g.*, bigrams, unigrams, biterms, and semantic information from GloVe [157] to define the clusters. Changes in proximity between text and clusters over time were used to determine whether a concept drift occurred.

Both Murena et al. [141] and Hu et al. [90] used LDA [24] to address their challenges (short-text classification and topic modeling, respectively). As Li et al. [118], Hu et al. [90] enriched data using external sources. They employed LDA to mine hidden information from these external sources to add the top representative words in the short texts. Drifts were flagged by calculating the semantic distance between each short text in the current and subsequent chunks. Similarly to Hu et al. [90], Murena et al. [141] used LDA for topic modeling in document streams. In this case, the authors integrated an ADWIN to LDA to detect topic drifts.

Li et al. [119] presented a method for short-text classification in text stream scenarios. The authors enriched short texts by using representations from BERT and Word2Vec. Both were trained using massive corpora, which, according to the authors, should be highly consistent with the topics related to the datasets the authors evaluated. In addition, the authors proposed a distributed LSTM-based ensemble method that includes a concept drift factor. The concept drift factor was used to determine the importance of an LSTM layer in the final result.

Wang et al. [204] provided a method called TCR-M for topic change detection and adaptation in textual data streams, which leveraged an ensemble and an extra classifier for error corrections. The topic change recognition process, according to the authors, not only detects the changes but also scores the severity of the change. The authors first used LDA for topic extraction, which was limited to ten. To detect drifts, the authors measured changes in topic probability between the current, the previous, and the next chunks. A potential change was scored by the statistical test with 0.1 of significance. If the p-value was below 0.1, the change was considered severe. The degree of severity defined whether the extra classifier should be retrained. The authors evaluated their method against a bagging model, Learn++.NSE [50], and LeverageBagging. Their method was presented in two versions: TCR-M, which reconstructed a bagging model at each time point, and TCR-M (retrain), in which the extra classifier was retrained based on the results. It is not clear if TCR-M had its bagging model fully reconstructed at each time point. The authors used an Amazon review dataset but split it into six subsets related to categories. Their method, mainly TCR-M (retrain), obtained the best accuracy values in four out of six subsets. However, in terms of F1-Score, the same method performed best only in one subset. The discrepancy of results across the metrics is not discussed, although the authors mentioned that the method was "only a preliminary attempt for text stream learning".

4.2.3 Semantic shift.

Semantic shift regards changes in the meaning of tokens over time. It is most commonly handled in papers that study linguistic changes over several years, decades, or even centuries. Generally, the datasets that support these tasks are entitled *diachronic*. However, semantic changes can also occur within a short time, such as in weeks [189] or minutes/hours [65]. The semantic shift was briefly introduced and discussed in Section 2.

Amba Hombaiah et al. [7], Lu et al. [127], and Periti et al. [159] approached the problem of semantic shift. Amba Hombaiah et al. [7] discussed the semantic shift as an analysis of whether it occurred. In the specific task of hashtag prediction, the authors evaluated the shift in top contextual words of the hashtags #china, #uk, and #usa, considering the years 2014 and 2017. The authors agreed that, in 2014, the contextual words related to #usa were related to the World Cup, while in 2017, the words were related to US politics. However, Periti et al. [159] aimed at detecting semantic shifts incrementally. In this case, the authors applied clustering methods, such as affinity propagation, to generate clusters in time slices. The authors determined a semantic shift by measuring the distance between embedding sets using metrics such as Jensen-Shannon divergence [147] and the distance between prototype embeddings. Lu et al. [127] presented a word-level graph-based method to generate dynamic word embeddings. The fundamental concepts were around maintaining long-term and short-term word-level knowledge graphs. These graphs preserved the co-occurrence between words. The relations between words helped define the occurrence of *semantic shifts*. For semantic shift detection, the authors evaluated the closest words to *apple* (in the New York Times dataset) and *network* (in the Arxiv dataset). In addition, the authors evaluated their method by considering trend detection and text stream classification. Although the aforementioned papers selected a small number of words to evaluate, there are shared tasks that monitored an entire vocabulary over time, *e.g.*, Zamora-Reina et al. [212], allowing participants of the shared task to develop their solutions, either considering the text streaming constraints or not.

4.2.4 Virtual drift.

According to Gama et al. [62], *virtual drift* regards changes in data distribution without changing the boundaries between classes. Using a similar notation as in Section 4.2.2, virtual drift happens when $p(X)$ changes but $p(y|X)$ does not. In addition, Gama et al. [62] stated that different definitions exist for virtual drift in the literature. Suprem and Pu [194] and Rabiu et al. [170][169] illustrated the *virtual drift* category. Virtual drifts must be tracked, particularly in cases where no classes or clusters' labels y are available.

Suprem and Pu [194] proposed a method for landslide detection. The method relied on social media data and governmental reports. Section 4.2.2 already cited this paper together with [193] and [192]. However, Suprem and Pu [194] explicitly emphasized their concern about handling the *virtual drift* problem. They highlighted that model fine-tuning is sufficient in this case, compared to model re-creation. Nonetheless, no reason for their concern about virtual drifts was provided. Rabiu et al. [170] presented a two-component method for concept drift detection applied to sentiment analysis and opinion mining. Similar to Suprem and Pu [194], Rabiu et al. [170][169] handled virtual drift. Although it is not explicit in the papers, the drift detection method used two windows to evaluate possible concept drift based on a distance metric to be selected. Different from most works that coupled a concept drift detector with a classifier to utilize the classification errors as a proxy for the detector, Rabiu et al. [170][169] used the input data, thereby using the concept drift detector to check $p(X)$.

4.3 Drift Detection Methods

We considered two categories for drift detection methods: *Adaptive* and *Explicit*. Fig. 8 depicts the categorization regarding the type of drift detection. In subsequent subsections, we describe selected papers from each drift detection scheme.

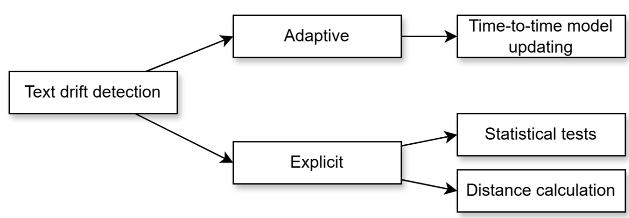

Fig. 8. Drift detection categories.

4.3.1 Adaptive. Adaptive corresponds to a self-updating model without explicitly detecting drift but rather from time to time. This category was called *blind adaptation* in [62]. A substantial number of papers considered *adaptive* approaches [2, 3, 7, 10, 16, 27, 39, 42, 81, 84, 105, 127, 133, 144, 164, 173, 192, 193, 202, 203, 210, 211]. Pohl et al. [164] proposed a batch-based method with an application for crisis management in social media. The method was based on active learning, and a user was queried whenever the classifier failed to confidently determine whether the input text was relevant to the task. The authors selected two events corresponding to two subsets from a more extensive dataset, *i.e.*, Colorado floods and Australian bushfires, and 1000 data points were labeled via crowd-sourcing. In addition, the authors mentioned that labeling data was particularly costly in streaming scenarios; however, it still required a human in the loop in a task such as crisis management. According to the authors, this model can adapt itself in the case of concept drift using the characteristics of the ML technique. For example, although they applied their scheme using k-nearest neighbors (k-NN) and support vector machines (SVM), it could be any other classifier. For instance, the authors claimed that when using k-NN and SVM, the continuous calculation of the boundaries results in drift adaptation.

Amba Hombaiah et al. [7] split the social media data by considering the years of publication. The work considered two datasets, corresponding to three different tasks: (i) 2014 Country hashtag prediction, (ii) 2017 Country hashtag prediction, and (iii) OffensEval 2019. The authors compared seven methods for each scenario: two static BERT models and five dynamic BERT models. Considering the static BERT models, one was trained with data from the previous year, and the other used data from the current year. For example, considering the 2014 Country Hashtag prediction task, one model was trained with tweets from 2013, and the other (a model checkpoint from the first model) was updated with an amount of data from 2014. The dynamic BERT models were fine-tuned using sampled tweets from the current year using different sampling methods, *e.g.*, uniform random, weighted random, token embedding, sentence embedding, and token masked language modeling (MLM) loss. The sampling methods defined different strategies for the model to overcome drifts/semantic shifts over time. The uniform random sampling was regarded as a sampling method in which the tweets from the current year were sampled randomly. In addition, the weighted random sampling method was used to sample the tweets from the current year randomly, considering the number of wordpieces generated by the tokens in the current year's tweets. However, token embedding, sentence embedding, and token MLM loss differ. The token embedding method assigned higher weights to tweets that contained new tokens and random samples from the current year's tweets. The sentence embedding method calculated the cosine distance between the updated and the current models. Both cosine distance and tweet length were used to determine a score, and then the sampling was performed. The token MLM loss method considered the last layer from the BERT model, masked out 15% of the tokens, and used the surrounding words to predict the masked ones. A high loss value may indicate drifts.

Yin et al. [211] proposed two algorithms for short-text stream clustering: MStream and MStreamF, a concise version that deletes outdated clusters. The algorithms receive document batches and are one-pass, in which the first document creates a new cluster, and the subsequent either selects one of the clusters to be assigned to or creates a new cluster. This assignment occurs after the batch is processed. The authors argued that concept drift is handled by assuming that the documents were generated by a Dirichlet Process Multinomial Mixture (DPMM) [9] and thus derived the probabilities of documents belonging to existing clusters.

D'Andrea et al. [39], Bechini et al. [16], Bondielli et al. [27], and Ducange et al. [48] tackled three different problems similarly: stance detection about vaccination, the Green Pass (as the EU Digital COVID Certificate is known), and body shaming detection, all in Italy. The authors in the first work categorized the application into *stance detection*, a branch of sentiment analysis. In these cases, the tweets were classified in a three-class fashion as either (i) in favor, (ii) neutral, or (iii) not in favor. Ducange et al. [48] addressed the task of binary classification regarding the use of body shaming language. D'Andrea et al. [39] and Bechini et al. [16] analyzed public opinion about vaccines in Italy based on tweets. D'Andrea et al. [39] addressed concept drift by incrementally retraining the model, such as an SVM model. However, they emphasized that considering their dataset, incremental retraining could not outperform a static SVM in terms of accuracy. Bechini et al. [16] handled concept drifts similarly to D'Andrea et al. [39]. However, the tweets from the new batch were semantically weighted according to previous events. Thus, the authors reached better values than other approaches, *e.g.*, static model, regular retrain, DARK [36], and the proposed semantic scheme. Although [16] was published in 2021, it was applied to regular vaccinations unrelated to COVID-19. However, Bondielli et al. [27] covered the opinion about the Green Pass concerning COVID-19. The authors evaluated different schemes to handle concept drift, including retraining with sliding windows and an ensemble of classifiers. The complete retraining led to the best average accuracy. Still, the highest feature space was reached due to the data accumulation and the utilization of TF-IDF as a text encoding method that generates a very high-dimensional representation. Ducange et al. [48] presented an approach for body shaming detection in Twitter posts between 2021 and 2022. Interestingly, the authors evaluated approaches considering the concept drift problem. However, the authors leveraged a "regularly retraining" approach rather than explicitly detecting concept drifts. The authors used TF-IDF representation to test the Complement Naive Bayes (CNB), Logistic Regression, and SVM. In the experiments, considering static, incremental, and sliding approaches, the best results were obtained by the CNB using the sliding approach. The sliding approach used a queue to manage the storage of new and past data.

Assenmacher and Trautmann [10] presented an online method for textual clustering, *i.e.*, textClust. In order to overcome concept drifts, the method leveraged a fading factor. It helped the model to exclude stale clusters. In addition, there was another parameter *tr* that dynamically determined the distance limit for a cluster to merge with another. This was also used to help determine whether a new input instance should be incorporated into a given cluster.

4.3.2 Explicit. *Explicit* approaches directly detect the drift via statistical tests or distance calculation. As examples of statistical tests used in the selected papers, we mention the Page-Hinkley test [153, 184], and ADWIN [20]. As examples of distance calculation metrics, we cite the Jensen-Shannon divergence [147], the Kullback-Leibler divergence test [109], and the cosine distance. Several approaches explicitly handled concept drift [33, 41, 55, 65, 78, 85, 90, 118, 119, 137, 141, 159, 168–170, 191, 194, 195, 204, 206].

Concerning explicit detection using statistical tests, Mohawesh et al. [137] tested four concept drift detectors: ADWIN [20], DDM [61], EDDM [12], and Page-Hinkley test [153, 184]. We considered ADWIN a statistical test because, in the original paper, the authors indicated that their statistical test verifies whether the observed average in subwindows is above a defined threshold [20]. In addition, DDM [61] and EDDM [12] performed evaluations based on the statistical properties of a stream and thus were considered in this work a *statistical test*. Mohawesh et al. [137], simulated concept drift by splitting the temporally ordered dataset into five chunks and rearranging them. The concept drift detectors used the calculated accuracy over the most recent input data as a proxy, *i.e.*, a window size of 200. ADWIN and EDDM had the best accuracy (coupled with a classifier) among the scenarios tested in the study.

Heusinger et al. [85] proposed a method that uses random projection for dimensionality reduction using text streams as input. In their experiments, preprocessing was done offline for the whole dataset to generate TF-IDF and embedding representations. Thus, their process was not fully incremental, except for the dimensionality reduction method, which was incremental (in batches). Considering the real-world dataset, *i.e.*, NSDQ, proposed in the same paper, the authors obtained a vector representation of 3442 dimensions using TF-IDF. Using their online dimensionality reduction method, NSDQ was projected onto 200 dimensions. The authors concluded that random projection could reduce the run time, even considering the offline preprocessing time. To detect concept drift, the authors used KSWIN [167], based on the Kolmogorov-Smirnov test [106, 186]. In this case, KSWIN monitored every dimension of the vector representation. In addition, the authors mentioned that different types of concept drift might be present because NSDQ is a real-world dataset [85]. Their assessment of concept drift detection relied on true positives and false positives. However, it is unclear how both metrics were calculated due to the absence of labeled drifts in the dataset. The results indicated more concept drifts detected in the original space, an expected outcome because KSWIN monitors each dimension separately. Finally, the authors mentioned that models trained with original and projected feature spaces maintained the same level of accuracy. Both Suprem and Pu [194] and Heusinger et al. [85] used t-SNE [201] plots to support the existence of concept drift in the datasets on which they applied their proposed methods.

Garcia et al. [65] evaluated the use of drift detectors for sentiment drift detection, using collected data during a soccer match in South America. The authors used the Incremental Word Context (IWC) [29] to trace back the events that generated the sentiment drifts. Using IWC, it was possible to determine which events generated the drift, who participated in them, and the atmosphere of the moment. The authors evaluated three drift detectors: ADWIN [20], EDDM [12], and HDDM

[56] (in the averaged and weighted versions). ADWIN was the most precise method, having a delay of around 2 minutes and raising only one false alarm.

Considering the *Explicit* detection with the aid of distance metrics, Li et al. [118] developed a method for short-text classification in the presence of topic drifts. As explained in Section 4.2, the approach automatically enriched the short texts using Probase. The topic drift detection was performed as follows: the short-text stream was received in chunks, and after they were clustered, the label distribution could be evaluated using the clusters. Subsequently, the distance between the cluster centers in sequential chunks was calculated using the cosine distance. According to the value obtained, the method categorized it either into: (a) no drift, (b) noisy impact, or (c) topic drift. In addition, the authors simulated topic drifts by generating datasets with topic changes after fixed periods. Their detection method was compared to nine drift detectors. Regarding false alarms, missing drifts, and delay, the proposed method obtained high average rankings, which were statistically equivalent (using the Bonferroni-Dunn test) to the best drift detectors in each metric.

Rabiu et al. [169, 170] developed an ensemble classifier coupled to a novel mechanism for drift detection-based adaptive windows (DDAW). Their method suited text streams, especially users' sentiments and opinions. Their approach can be divided into two components: (i) drift detection and (ii) classification. In many applications, classification errors are used as a proxy for the drift detector. However, the drift detection component compared the data distribution considering two windows. Thus, it was possible to measure drift by evaluating the dissimilarity between the windows. An intriguing aspect of this approach was that it allowed for distance metrics and statistical tests. In the paper, the authors compared the Hellinger distance [82], Kullback-Leibler divergence [109], Total Variation distance, and the Kolmogorov-Smirnov test [106, 186]. Their approach, coupled with the Hellinger distance, obtained the best values regarding false alarms, detection rate, and accuracy, even compared to other drift detection methods, *i.e.*, AEE [107], RDDM [40], and Page-Hinkley [153, 184]. It was unmentioned how the drifts were labeled or whether the data was rearranged to simulate drifts.

Suprem and Pu [194] developed a system for landslide detection, a physical event that causes destruction and for which there are no physical sensors to detect. The authors combined data from social media and governmental agencies to perform the detection. Concept drift was detected using the Kullback-Leibler divergence test [109] to evaluate the distribution of two batches. The model was updated by generating or updating the classifiers to handle the concept drift.

Li et al. [119] presented a distributed long short-term memory (LSTM)-based ensemble method for short-text classification in text stream scenarios. The short texts were enriched by using BERT and Word2Vec models. The LSTM-based method included a concept drift factor used as a threshold to compare the distance between the LSTM layer trained with the previous batch and the layer trained with the current batch. If the concept drift factor was above the threshold, the weight of the current layer would be bigger to generate the combined final output.

Fenza et al. [55] proposed a fuzzy-formal-concept-analysis-based index for concept drift detection and applied the method to a fake news classification problem. Although the concept drift detection was not directly approached, the authors calculated the correlation between the classifier's performance and the proposed index. The index was calculated from a fuzzy lattice, *i.e.*, a fuzzy hierarchical knowledge structure, while the classifier's performance was calculated using F-Score and accuracy. Their results demonstrated a high (Sperman's and Pearson's) correlation, between 69% and 87%. The authors claimed that the method had the potential to be used as a proxy for the model update process. In addition, the fuzzy lattice seemed never to be updated, which may hamper the model from properly working over a long time. Susi and Shanthi [195] proposed a sentiment drift analysis system based on BERT models, namely TSDA-BERT. According to the authors, the system receives data in a sliding window fashion corresponding to four days. The authors calculated the positive and negative scores per window based on the proportion of them in the window. From these values, a sentiment drift measure was calculated by simply subtracting the number of negative from the number of positive tweets. This measure was used for sentiment drift detection by calculating it between time periods; if the score was negative and later went positive or vice-versa, it indicated a drift.

Rabinovich et al. [168] proposed a model-agnostic framework for drift detection. More specifically, the authors focused on data drift, *i.e.*, virtual drift. This paper presents a dataset comprising texts used as requests to a virtual assistant. In this case, drifts may occur due to novel topics and deviation from previous topics, and these may result from real problems such as external trends, new features/services introduced by a company, *etc.* In addition, the authors mentioned the difficulty of obtaining datasets regarding this scenario, and therefore, they created a novel dataset, mimicking user requests and then introducing drifts. The introduction of drifts was performed using the Parrot paraphrasing framework [38] and LAMBADA [8]. Although the authors frequently mentioned in the paper the terminologies *stream*, *text stream*, and *short-text stream*, their approach was not fully incremental. Their approach consisted of training an autoencoder to learn the data distribution of a dataset of interest. From this point, their approach was able to compute the similarity between data chunks and the original distribution, learned by the autoencoder. Later, the drift and change point detection was performed. Additionally, their method contains a module for drift interpretation based on a clustering algorithm. Their method contained a single parameter corresponding to a threshold for the cosine distance between the original representation and the reconstructed (through the autoencoder) to detect drifts.

4.4 Model Update Method in Text Stream Settings

We also looked closely for information regarding the model update scheme from the analyzed papers. Fig. 9 depicts the organization. We found four mechanisms: (i) *Ensemble update*, in which the base learners are substituted or removed over time; (ii) *Incremental*, which corresponds to the model incrementally learning new data without a retraining process, splitting regarding the amount of data used to learn: one input at time or batches; (iii) *Keep-compare-evolve*, which corresponds to methods that generate and evolve new models to adapt to drifts and uses the old model to measure the similarity between information from both models; and (iv) *Retraining*, which can occur after detecting a concept drift, or time-to-time, which does not detect drifts but adapts to them.

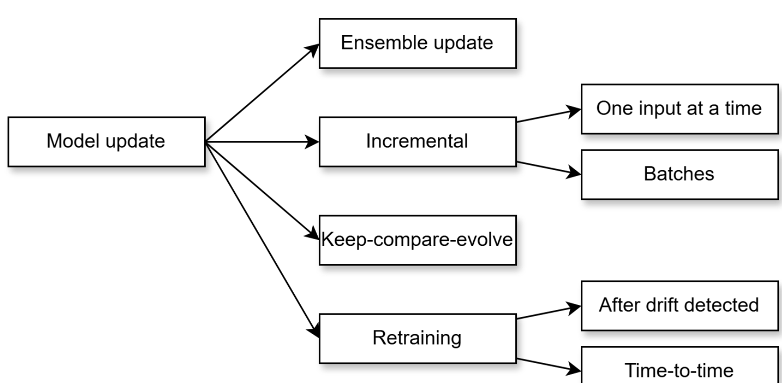

Fig. 9. Model update methods used when handling text streams bound to concept drift.

4.4.1 Ensemble update. In this category, the works proposed techniques that create, update, and combine multiple models, the so-called ensembles. Over time, an ensemble can be updated by removing outdated base learners while adding new base learners trained on newly arrived data. Suprem et al. [192], the presented system for landslide detection used batches to update the model. The landslide detector used a classifier, which was an ensemble. The authors mentioned that they used two approaches for selecting base learners: relevancy and recency. When using relevancy, a k-NN search was performed to discover the most relevant base learners from a pool of trained base learners, considering the centroid of the data used to train these learners. However, the recency scheme returned the most recent base learners used to compound the ensemble. In addition, the weighting scheme can be configured as an unweighted, weighted, or model-weighted average. The unweighted average considered the base learners equally to provide an output. The weighted average considered weights provided by domain experts, and the model-weighted scheme considered the base learners' prior performance to weigh them.

Sun et al. [191] described an ensemble classification model for short text classification in environments bound to concept drift. The paper emphasized three main aspects: a feature extension based on the short text features, a concept drift detection method, and an ensemble model. Considering the ensemble model, the authors used SVM as a base classifier. A new classifier was added when concept drift was detected. If the classifier pool is complete, the oldest classifier was removed to add the current one after being trained on the new batch.

Hu et al. [90] proposed a short text stream classification method based on content expansion coupled with a concept drift detector. The expansion was performed by adding information from external sources, and 100 Wikipedia pages related to 50 keywords were selected, totaling 60,600 pages. The classification task in this study was performed using an ensemble of SVMs, in which each base learner was trained per chunk using the expanded texts. The number of base learners was limited to a specific parameter H: when this number is met, the oldest learner is replaced. In specific situations, the latest learner can replace an older learner trained using semantically similar chunks.

Rabiu et al. [169, 170] presented an ensemble method for classification. Particularly, Rabiu et al. [169] tackled the sentiment classification problem. The ensemble model was updated over time by removing the worst base learner from the ensemble when it reached the maximum number of base learners. To determine the worst base learner, a weighting calculation is performed by leveraging the base learner's mean squared error on the new input data, *i.e.*, MSE_i, and the base learner's mean square error on the data from the previous batch (reference data), *i.e.*, MSE_r. The complete weight calculation for a base learner was performed as $weight = \frac{1}{\mathrm{MSE}_r + \mathrm{MSE}_i + \alpha}$, where α is a non-zero factor to avoid division by zero.

4.4.2 Incremental. The *Incremental* update scheme regards models capable of learning from new pieces of data without completely retraining the model. In our selection, several papers employed incremental models to approach their applications [2, 3, 10, 29, 42, 78, 81, 84, 85, 105, 118, 127, 133, 137, 141, 144, 159, 173, 193, 194, 202, 203, 210, 211]. However, we distinguished between the manners in which the data were inputted into the model: (i) One input at a time and (ii) In batches.

Heusinger et al. [84] proposed a method for dimensionality reduction using random projection. As already cited in Section 4.3.2, the process was not fully incremental. In this study, the authors utilized three classifiers: (i) Adaptive Robust Soft Learning Vector Quantization [83], (ii) Adaptive Random Forest [71], and (iii) Self-adjusting Memory k-NN [126]. The dimensionality reduction method uses a window of size 1000. However, when applied to the classification methods, the process in incremental *One input at time*, except for the Self-adjusting Memory k-NN, which Heusinger et al. [84] cited that they used as parameters five neighbors and a window size of 1000 to match the window size of the random projection. Mohawesh et al. [137] incrementally updated the models. In their case, they used Stochastic Gradient Descent for SVM, Perceptron, and Logistic Regression algorithms incrementally. However, similar to Heusinger et al. [84], the process was not fully incremental because it used TF-IDF and principal component analysis (PCA) for dimensionality reduction.

Abid et al. [2, 3] presented a method for text stream clustering called AIS-Clus, based on the artificial immune system [100]. This system had online and offline phases. The offline phase comprised receiving historical data to generate the first clusters. In the online phase, new data were divided into equal blocks, *i.e.*, it worked in batches. Concurrently, each instance was evaluated alone, being also capable of handling novel classes. Thus, this work could be categorized as *Incremental in Batches* or *Incremental with One input at a time*, depending on the point-of-view. Although it worked in a clustering fashion, the method performed classification tasks.

Murena et al. [141] presented the adaptive window-based incremental LDA (AWILDA), a method for topic modeling in document streams. This method contained two LDA models, one for topic modeling and another for drift detection, with the help of ADWIN. It received the data in batches, making it possible for the approach to use ADWIN as a drift detector and to resort to LDA over the batch.

Assenmacher and Trautmann [10] presented a stream text clustering method. The use of online and offline phases for algorithms that perform stream clustering is well known. The offline phase generally performs adjustments in the model, such as the stale cluster removal and merging of similar clusters. In the online phase, the method received input data and verified the most similar cluster to assign the new input data to the most similar cluster. However, a new cluster is created to accommodate the incoming text if no cluster is sufficiently similar. Due to these characteristics, this method could be categorized as *Incremental with One input at a time*. Interestingly, this method outperformed other batch-based methods in the evaluation considered in the paper.

4.4.3 Keep-compare-evolve. Amba Hombaiah et al. [7] is the single representative of this model update category. As aforementioned, this study proposed three methods for sampling to update the language models. The three methods, *i.e.*, the Token Embedding Shift method, Sentence Embedding Shift method, and Token MLM Loss method, used both current and previous models to evaluate changes to sample new data to fine-tune the current model. Thus, more significant differences between a given text representation and the representations provided by the old and current models generate higher chances for a given text to be selected for fine-tuning. Thus, in this specific case, it is costly to fine-tune using all the data because of the size of the BERT models. In addition, GPUs are necessary to speed up the training/update of these models.

4.4.4 Retraining. Some papers resorted to the complete retraining of models. The retraining can be triggered by drift detection or periodically, typically after batch processing. As noted, Chamby-Diaz et al. [33] proposed a dynamic feature selection method to handle feature drift, namely Dynamic Correlation-based Feature Selection (DCFS). This method used concept drift detectors, such as ADWIN. Concept drift detectors generally provide two levels of signaling: warning and drift. Whenever a warning signal was outputted, DCFS updated the covariance matrix incrementally. The feature-feature and feature-class correlations were calculated when a drift signal was emitted. Thus, a new Naive Bayes model was trained from scratch using the feature subset selected according to the correlation-based feature selection (CFS).

Other works also utilized the retraining scheme [16, 27, 39, 48]. All these papers compared approaches that resorted to the retraining scheme. Retraining occurs regularly and considers data from events. However, the dataset was increased incrementally to be used by the methods during the training step. For example, when event #10 concluded, the data related to this event were appended to the data regarding previous events. Thus, a new model can be trained based on the dataset, now containing the data about event #10. Concerning these four works, only Bondielli et al. [27] used an incremental approach, *i.e.*, Complement Naive Bayes [177] with the partial fit. For this approach, however, the authors used TF-IDF for vectorization, which was not updated during the online monitoring after the first event. Thus, the process was not fully incremental. In addition, the authors did not mention any strategy for maintaining a dataset in a feasible size after several incremental additions of batches. Ducange et al. [48] used the retraining scheme even with the so-called *sliding* and *incremental* strategies. In their paper, *sliding* added new data and removed old data in a data structure for model retraining, and *incremental* accumulated data over time, which directly impacts the dimension number of the TF-IDF representation.

The system proposed by Susi and Shanthi [195], *i.e.*, TSDA-BERT, also considered periodic retraining to overcome sentiment drift. Whenever a sentiment drift happens, the system uses a domain impact score, which calculates the impact of a tweet in the domain. The calculation considers the intersection of a tweet's words and the domain-specific impact words. According to the authors, if the impact was above 0.5, it indicates adherence to the domain. However, the authors did not explain how the domain-specific words were selected. Compared to D'Andrea et al. [39], Bechini et al. [16] and Bondielli et al. [27], Susi and Shanthi [195] provided a strategy to maintain the training set in a feasible size. The tweets with higher adherence to the domain were included in the training set, and the same number of tweets were removed from the training set. It means that the training set is always the same size. The authors mentioned the utilization of at most 324,685 tweets in the training set. This training set was used for fine-tuning over time.

4.5 Stream Mining Tasks applied in Text Stream Settings

In Fig. 10, we organized the stream mining tasks addressed and the respective applications in the analyzed papers, considering the information obtained from the selected papers. This subsection addresses the Research Question 2 (RQ2), *i.e.*, *"Which type of application is addressed?"*. In this study, we considered Stream mining tasks: (a) *Classification*; (b) *Clustering*; (c) *General detection*; and (d) *Topic modeling*.

4.5.1 Classification. Classification is among the most common stream mining tasks. In the general classification, the objective is to predict, with arbitrary accuracy, a unique class from a small set of values from a given input. Some applications found in the papers addressing the classification task include (i) *crisis management*; (ii) *fake news detection*; (iii) *fake review detection*; (iv) *hashtag prediction*; (v) *sentiment analysis*; (vi) *short-text classification*; and (vii) *spam detection*.

Regarding *crisis management*, Pohl et al. [164] aimed to identify the relevant tweets about two environmental disasters: the Colorado floods and the Australian bushfires. It is considered a binary classification task because the model assesses whether or not a tweet is relevant, sometimes with a human in the loop. Their approach was evaluated regarding the average error and the number of queries. Because the method presented in [164] employed active learning strategies, the label uncertainty determines whether the system should query a user. Only Pohl et al. [164] represented this application in the classification task.

Fake news detection was addressed by Fenza et al. [55]. The authors proposed an index based on fuzzy formal concept analysis, which correlates with the classifier's performance. According to the authors, the fake news detection problem is generally tackled as a binary classification, where a model should classify news as fake or real. The authors evaluated three ML methods: Random Forest, Naive Bayes, and Passive-Aggressive [37]. Although the authors proposed the method, they did not couple the index to the methods to trigger retraining. Three datasets containing news articles between 2018 and 2020 were used, *i.e.*, NELA-GT-2018, NELA-GT-2019, and NELA-GT-2020

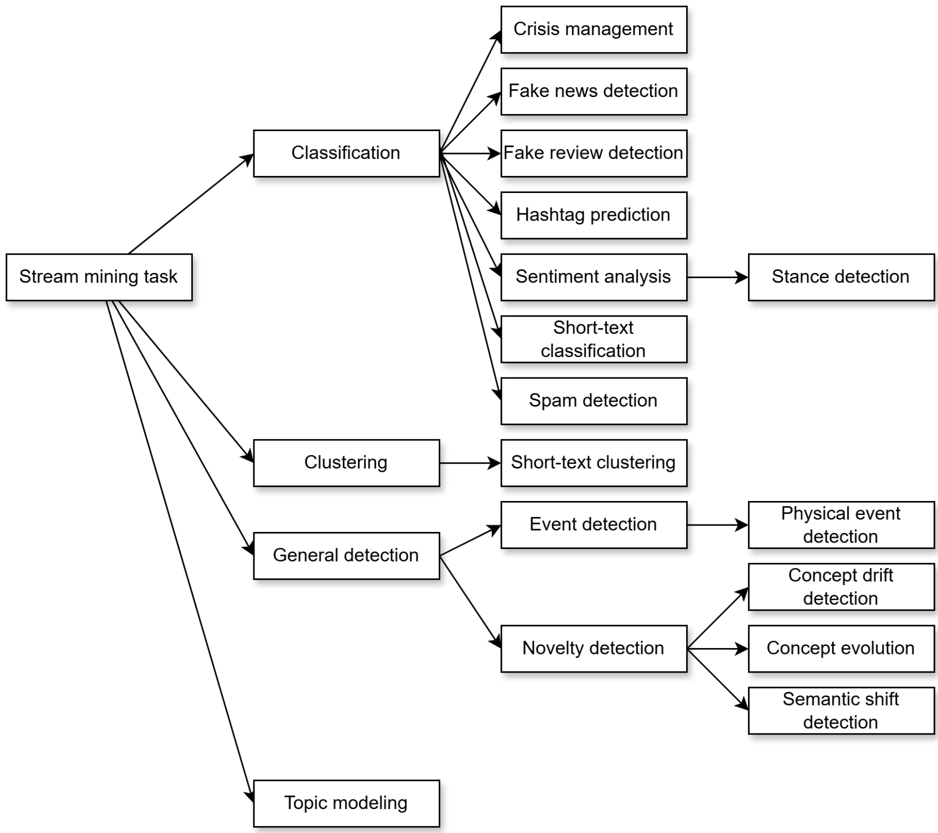

Fig. 10. Text stream mining tasks and applications found in the selected papers.

(see Section 5). According to the authors, only the Passive-Agressive algorithm was tested online, and the news between February and August 2018 were used as the training set, considering also the fuzzy lattice structure. The classifiers' evaluation was performed using accuracy and F1-score. The evaluation of the proposed index happens through visual analysis, Pearson's and Spearman's correlation, and cosine similarity. The authors argued that their method would allow early drift detection but did not provide experiments or evidence.

Considering *fake review detection*, Mohawesh et al. [137] tackled this task using three ML methods: SVM, logistic regression, and perceptron. The authors used four Yelp datasets, only one containing fake and genuine reviews. Mohawesh et al. [137] noted that the datasets "were built based on an unknown filtering algorithm and web-scraper techniques to label each review as fake or genuine". Because the idea was to determine whether or not a review is fake, it corresponds to a binary classification task. The ML methods were evaluated using accuracy and statistically assessed using the Nemenyi test [143]. The authors claimed that their work is the first to address concept drift in the fake review detection problem. Considering the selected papers, this was the only method that approached fake review detection.

Hashtag prediction was addressed in [7], [84], and [85]. Both Heusinger et al. [84] and Heusinger et al. [85] used random projection as a dimensionality reduction method for text streams. Also, a dataset, *i.e.*, NSDQ, was proposed for the problem because it generated high-dimensional data and could be reduced in real-time by random projection. Furthermore, this is the only real-world textual dataset addressed in these papers, while the others are synthetic. This dataset contains 15 classes that make the stream mining task approached by them a multiclass classification. The evaluation was performed in terms of accuracy, Cohen's Kappa, and run time. Amba Hombaiah et al. [7] tested the sampling approaches for updating BERT using two datasets: OffensEval 2019 and Country Hashtag Prediction. Approaching OffensEval constituted a binary classification task; thus, the authors used the Area Under Curve (AUC) of the Receiving Operating Characteristic (ROC) curve and F1 score. However, addressing Country Hashtag Prediction corresponded to a multiclass classification task and was evaluated using micro-F1 score, macro-F1 score, and accuracy.

In *sentiment analysis*, the objective is to develop a model capable of inferring a user sentiment from text. According to Medhat et al. [132], "sentiment analysis (SA) or opinion mining (OM) is the computational study of people's opinions, attitudes and emotions toward an entity". Similarly to sentiment analysis, *stance detection* regards the position of a given text's author about a specific topic, considering the labels *in favor*, *neutral/neither*, and *against*, sometimes expressed in literature with different labels but with similar meanings [16, 108]. Bechini et al. [16], D'Andrea et al. [39], and Bondielli et al. [27] approached *stance detection*, with [16] and [39] related to vaccination, and [27] regarded the stance about the green pass, as mentioned in previous sections. The authors in these three works collected the dataset that they needed to utilize primarily from Twitter. As aforementioned, *stance detection* classifies texts in three labels, indicating that it is a multiclass classification task. D'Andrea et al. [39] used F1 score, precision, recall, AUC, and accuracy to evaluate the method. Bechini et al. [16] evaluated models using accuracy and F1 score, and Bondielli et al. [27] used F1 score, accuracy, and the number of features in each model.

Bravo-Marquez et al. [29] proposed a sentiment lexicon inductor for time-evolving environments in a sentiment analysis context. The authors claimed that sentiments could change over time, while new words in different sentiments can emerge. In addition, the lexicon would be static in a fully incremental system without sentiment induction. In this case, from a seed lexicon, the authors processed the dataset in a stream fashion and, at the same time, inferred sentiment from tokens absent in the lexicon. Although, in practice, the system outputs a value limited by a logistic function, we presented this paper in the classification section because of the sentiment analysis application. In addition, the authors tested their approach by deliberately changing lexicon sentiment scores and measuring how long the system would take to recognize the new sentiments. Finally, the authors used accuracy and Cohen's Kappa to evaluate the classifiers applied together with their method.

Garcia et al. [65] leveraged a lexicon-based classifier for sentiment analysis in a text stream environment regarding a soccer match. The authors observed that the sentiment changes very quickly, derived from the events in the soccer match, supporting the statement that sentiments could change over time [29]. Using ADWIN, HDDM, and DDM, the authors observed that ADWIN obtained the best results in terms of missing drifts, delay (regarding time and posts), and false alarms. This work also fits the category *Concept drift detection*, because the sentiment stream classification was not the objective, but a means of evaluating the sentiment drift detections.

Aiming at improving classification performance using fewer data, some papers, *e.g.*, Roychowdhury et al. [179], adapted text stream mining tasks. Originally working on regular classification tasks, the authors proposed converting to entailment-style modeling. The method generates augmented data by creating multiple pairs of text and label hypotheses, where only one pair is true, and the others serve as negative examples. This approach enables the model to adapt to new concepts with significantly less labeled data, particularly in few-shot learning scenarios. The proposed technique was evaluated on both real-world and synthetic datasets, reaching the best values regarding macro F1-score. The authors claimed a 75% reduction in labeling costs compared to regular fine-tuning methods.

Short-text classification is addressed in [118, 119, 191] and [168]. Li et al. [118] proposed a method for short text streams bound to concept drift. This method took advantage of Probase for short text enrichment. The approach was evaluated in terms of time and accuracy. Sun et al. [191] described a method for text stream classification based on feature extension and ensembles formed by ensembles. This method can handle concept drifts by calculating the distance between each short text in the previous and new batches. Li et al. [119] proposed a method for short text classification in text stream scenarios. This method enriches text by using representations from BERT and Word2Vec. In addition, the method uses a Convolutional Neural Network (CNN) to extract high-level features. This method handled concept drift by resorting to a concept drift factor used in the systems. Both approaches in [118, 191] and [119] were applied to the same datasets, *i.e.*, Tweets, TagMyNews, and Snippets, considering text classification as topics.

Rabinovich et al. [168] addressed short stream classification, using as dataset user requests to virtual assistants. In addition, the authors mentioned that drifts emerge in this scenario due to the deployment of new features/services, external trends, or service interruption, for example. One of the challenges in this specific work was the shortness of the texts, in which most had less than five words. Therefore, to have a significant number of samples, the authors employed a generation method [171], Parrot, and LAMBADA, reaching 600,000 samples. Rabinovich et al. [168] proposed four drift scenarios for evaluation: (a) gradual drift, (b) abrupt drift, (c) no drift, and (d) short-lived anomaly. In the drift scenarios, the authors introduced drifts by positioning a subset in specific points of the stream, *i.e.*, uniformly for the gradual drift, and at the timestep t=15 for abrupt drift. Their method contained a single parameter, which corresponded to a threshold for the cosine distance between the original representation and the reconstructed (through the autoencoder) to detect drifts. Interestingly, the proposed method is the only one agnostic to the model. However, the method was not fully incremental due to the autoencoder training using the anchor dataset.

Some papers addressed *Spam detection* as experiments [33, 42, 133]. Because the goal is to classify a piece of text into either non-spam or spam, the task is considered a binary classification task. Melidis et al. [133] provided an ensemble-based mechanism for predicting a feature's probability of association with a given class by considering that words might be subject to temporal trends and a sketch-based feature space maintenance mechanism that allows for memory-bounded feature space maintenance. The approach utilized an ensemble compounded by statistical techniques to account for feature periodicities. The ensemble consisted of a Poisson model [133], a Seasonal Poisson model [88], an Auto-regressive Integrated Moving Average (ARIMA) model [28], and an Exponential Weighted Moving Average (EWMA) model [148], to capture regular, seasonal, auto-correlated, and sudden trends. A sketch-based approach was designed to maintain a concise feature space. The authors tested three versions: a baseline sketch that retains only word and occurrence counts, a fading sketch that considers the importance of frequent words, and a drift-detector-based sketch, which uses ADWIN to detect the decrease in word usage. The approaches were compared in terms of accuracy, Cohen's Kappa, and run time.

Chamby-Diaz et al. [33] proposed a method for feature selection based on correlations to handle feature drifts in data stream scenarios. The method is not exclusive to spam detection, but the spam dataset was the only text-based dataset used by the authors. The method is evaluated in terms of accuracy. de Moraes and Gradvohl [42] proposed a method for feature selection in binary text stream classification tasks, namely OFSER. The proposed method leverages adaptive regularization and weighs the input for each new data. The regularization, according to the authors, decreases the impact of the feature drift. Despite being fast and having decent overall performance, their method depends on a parameter to define the number of features to be selected from the original set. The method runs on top of a Naive Bayes classifier, chosen due to its simplicity and naive assumption of independence among the features. The approach was evaluated using F1 score, accuracy, memory consumption, and run time. Furthermore, due to "an undesired conservativeness of the Friedman test" [42], it was statistically assessed using the Iman-Davenport test [92] instead of the Friedman test [57, 58], and the Bergmann-Hommels' procedure [68] instead of the Nemenyi test [143]. OFSER ranked among the three best approaches.

As expected, the most frequent metrics in this stream mining task were accuracy, Cohen's Kappa, F1 score, AUC, and run time. Although not all methods were assessed regarding run time, it is crucial to have values for this metric due to its use in streaming scenarios, where time and memory consumption are constrained.

4.5.2 Clustering. *Clustering* is a stream mining task in which the aim is to find intrinsic clusters, according to their features [21]. The general idea is to minimize the similarity between different clusters and maximize the intra-cluster similarity [18]. Differently from classification, in the clustering task, the labels are not available before the learning process. Therefore, alternative metrics are necessary, and since there is no ground truth, the learning process is named unsupervised [21].

Three works approached the stream clustering task [2, 3, 10]. The first two papers presented similar approaches that use the artificial immune system (AIS) for text clustering, while the third presents textClust, a stream clustering method. Abid et al. [2] developed a method for text stream clustering based on the AIS called AIS-Clus. It used heuristics based on the AIS to cluster data efficiently and, by discovering these clusters, can also detect concept drift and feature evolution. The authors could also recognize new classes corresponding to the concept evolution task in the experiments. According to the authors, the AIS is analogous to the biological immune system because it receives an intruder, clones specific cells, and handles the intruder until it dies. In their approach, for each new input (analogized as antigen), a scoring function calculates its adherence to each cluster (analogized as a B-cell). The clonal selection makes copies of clusters that undergo a mutation process. Later, the negative selection mechanism makes it possible to detect noisy data. Their method does not start from scratch, having an initial static phase for preprocessed historical data clustering. The other phase is online stream processing, which receives the clusters from the first phase as input. The authors used a survival factor for each word in an aging-like scheme. Although it works in a clustering scheme, the method is evaluated in terms of F1 score, accuracy, recall, and precision. More information is provided in [3], which expands on [2], and new experiments are executed. For example, to test the approach's capacity to handle new classes, the authors arranged data in three datasets to simulate the emergence of new classes/events, one of which included texts in Arabic. When AIS-Clus is compared to other methods, *i.e.*, CluStream and DenStream, it achieves the best results regarding the precision, recall, and number of clusters, functioning as a classifier as described in [2].

Assenmacher and Trautmann [10] presented an online method for textual clustering, namely textClust. The algorithm is available within RiverML Python library [138][11]. Over time, in the offline phase, the model was maintained concisely by merging similar clusters and removing outdated ones. A fading factor for the cluster weighting is used to determine cluster staleness. The method was evaluated in terms of homogeneity, completeness, and normalized mutual information (NMI). Homogeneity evaluates how well a clustering method assigns the data points to the clusters. Reaching 1 for homogeneity means that each cluster contains data points of a single class. On the other hand, completeness measures whether the data points of a given class were assigned to the same cluster. Reaching the value 1 for completeness means that the data points of each class were assigned to a single cluster. The authors support these statements by mentioning that "completeness scores tend to be lower than the homogeneity scores", and that it "indicates that

[11] https://riverml.xyz/0.19.0/api/cluster/TextClust/

online clusters are quite pure with low entropy, but the topics are distributed over multiple clusters" [10].

Most selected works that addressed a stream clustering task focused on short-text clustering. Rakib et al. [173] proposed an efficient method for similarity-based short-text stream clustering called EStream. The method's efficiency comes from utilizing an inverted index to find the most similar clusters. The authors tested lexical (unigram, bigram, and biterm) and semantic text representations (using a pre-trained GloVe [157]). Their method has two steps: the online and the offline phases. First, each cluster is lexically represented as a cluster feature 4-sized vector consisting of the features (in unigram, bigram, or biterm), their frequencies in the cluster, the number of texts in the cluster, and the cluster identifier. The semantic representation consists of the cluster vector and the cluster center, calculated from the average of the GloVe representation of the texts. EStream was compared in terms of NMI, homogeneity, and V-measure. EStream had the best performance in 50% of the datasets used for evaluation. The authors highlighted that EStream requires less running time and that it stores more information than the other approaches, but that would be an acceptable trade-off [173]. They also highlighted that EStream might perform inadequately in more extensive texts.

Vo [203] proposed a new method called GOWSeqStream, for short text stream clustering, using deep sequential methods, graph-of-words representation, and pre-trained word-embedding models. It uses subgraph mining to extract semantic information from the texts, although it lacks information on how to use it, considering even the number of sliding windows and the support. The method also utilized Word2Vec representations to generate embeddings to serve as input for other deep encoders, such as GRU. The author also experimented using bidirectional LSTM, Doc2Vec, and BERT representations. These representations were utilized as input for a DPMM. The method was compared with five approaches using three datasets; the proposed approach achieved the best values for two. The author also compared the representation generation; the best combination was with BERT and Bi-LSTM. In addition to English, the author used a Vietnamese text dataset as a final test. In this scenario, the proposed approach achieved the best results among the competitors. As in [173], the authors used the NMI as the primary evaluation metric.

Yang et al. [210] proposed a new short text stream clustering method using an incremental word relation network. The authors highlighted their primary contribution as (a) a new method for real-time short text clustering using a bi-weighted relation: term frequency and co-occurrences, to overcome sparsity; (b) a fast method to locate core terms that represent text clusters sufficiently; (c) the mechanism to overcome topic drift, removing outdated relations and incrementally adding new terms and relations. In addition, the authors proposed a new data structure to represent the clusters, which they named *cluster abstract*. This data structure had five fields: an index, the number of short texts in clusters, the sum of timestamps, the squared timestamps sums, and a new attribute compared to EWNStream (their previous approach) called *pd*, containing a core term set. The method used data windows and specific calculations to update the model to add new data, exclude outdated data, and merge clusters. Besides, the method had a decay scheme to control the forgetfulness of old clusters. In essence, the method develops a graph containing terms and relations, and the clusters were obtained from groups of closely related words. The method searches for a cluster abstract with the most intersection of words considering the input data to predict a cluster to newly inputted data. Using a dataset crawled by themselves, the authors compared their proposed method against EWNStream, MStream, Sumblr, and Dynamic Topic Model. EWNStream+ outperformed its previous version (achieving roughly 86% of NMI accuracy) and was approximately 30 percentage points better than MStream, the third in the ranking. In addition, the run time was very modest across different stream lengths.

Yin et al. [211] proposed two text stream clustering algorithms: (a) MStream, a one-pass clustering method that utilizes Dirichlet Multinomial Mixture Model (DPMM) and an update process per batch; and (b) MStreamF, which deletes outdated clusters, maintaining a concise model. Considering the clustering process of the MStream algorithm, there is the assumption that the new documents arrive sequentially, and each is processed only once. The initial document generates a new cluster, and subsequent documents choose one of the existing clusters or create a new one. The authors' updating process proves beneficial in the batch processing of text streams. The process was designed such that each document gets assigned and then temporarily deleted from the cluster so that the similarity of the other documents in the same batch is not impacted. After completing the batch process, all documents are assigned to their original cluster. For MStreamF, the authors developed a deleting scheme that works for batch processing by adding a new parameter B_s, which accounts for the number of batches. When the number of processed batches meets the B_s parameter, the new batches are processed after the documents related to the oldest batch are deleted. As the iterations go by, it is expected that some clusters will become empty, indicating that they are outdated and could be deleted. The approaches were assessed in terms of NMI, run time, and number of clusters. They concluded that MStreamF is faster than MStream due to the conciseness of the former model. Comparing the proposed and the state-of-the-art models, MStream and MStreamF outperformed their competitors. MStreamF performed best with temporally ordered datasets, whereas MStream performed best with unordered datasets. The run time of all algorithms increased linearly with the size of the datasets, while the single-pass algorithms were faster.

In summary, the NMI, run time, and the number of clusters were the most often used metrics for stream clustering and short-text stream clustering. The latter may be considered a measure of conciseness, which directly corresponds to one of the constraints of streaming scenarios, *i.e.*, memory consumption, and may indirectly impact run time. NMI, a Shannon-entropy-based metric, measures the similarity of two sets and, concerning clustering, the similarity of the ground-truth and the model-generated clusters [51, 211]. Other metrics may appear, such as completeness and homogeneity. Those metrics vary between 0 and 1, where the higher, the better. Homogeneity evaluates how well a clustering method assigns the data points to the clusters. A perfect homogeneity, *i.e.*, 1, indicates that each cluster contains data points of a single class. As aforementioned, completeness evaluates whether the data points of a given class were assigned to the same cluster. A perfect completeness value suggests that the data points of each class were assigned to a single cluster.

4.5.3 *General detection.*

In this category, we grouped papers that tackled *event detection* and *novelty detection*. According to [53], novelty detection is "the ability to identify an unlabeled instance (...) that differs significantly from the known concepts". As suggested in [53], we considered *concept drift detection*, *semantic shift detection*, and *concept evolution* as sub-categories of *novelty detection*. We separated this section to encompass approaches that focused on detection rather than incorporated detection methods in classifiers or clustering methods, for instance.

We also considered *physical event detection* a sub-category of *event detection*. As mentioned previously, Suprem and Pu [193], Suprem et al. [192], and Suprem and Pu [194] described distinct aspects of a system for landslide detection. They utilized governmental reports as trustworthy sources and social media posts as social sensors (also named strong and weak signals, respectively). The system was described as fully autonomous and continuously evolving, becoming unnecessary human intervention. Although the works were similar in several aspects, there were minor variations in the evaluation metrics. Suprem and Pu [193] selected precision and F1 score metrics. The event detection was assessed using false positives and false negatives, where the original variant of the system was used as ground truth. Suprem et al. [192] used F1 score, precision, recall, and the number of events detected as metrics. There was no ground truth regarding the number of events: only the events counted. Suprem and Pu [194] used accuracy to evaluate classifiers' performance across data windows.

Kolajo et al. [105] proposed a framework for real-time event detection using social media as a data source. The interesting highlights in this paper regard the tweets' enrichment for slang, abbreviations, and acronyms based on external sources. The method creates a local vocabulary using data from various external sources. In addition, the authors utilized spelling correction and emoticon replacement. The authors used an incremental clustering algorithm to cluster events and then rank these events based on important words for each event. The authors evaluated their method using two experiments: (a) comparing it to the General Social Media Feed Preprocessing Method (GSMFPM) to determine if the enrichment layer performs effectively; and (b) event detection from social media. In experiment (a), the authors represented the tweets using unigrams and bigrams, supposedly later converted to GloVe (unclear in the paper). Later, the vectors are applied as input to a Feedforward Neural Network (FNN) and a CNN. These approaches are not incremental, thus presenting concerns about the process' timeliness regarding real-time events. In this experiment, they measured the cross-entropy loss across the training epochs for both Twitter Sentiment Analysis and Naija datasets. Their method outperforms GSMFPM. The second experiment measures accuracy over events in social media, using precision, recall, and F1 score. The authors used a dataset called Event2012, which contains annotations about events. The proposed method obtained a higher F1 score than the other approaches.

Regarding *novelty detection* and its subdivision in this study, only one paper exclusively considers *concept drift detection* [41]. Three included the *concept evolution* problem [3, 33, 206], and another mentioned the *semantic shift detection* [159]. Considering the *concept drift detection*, de Mello et al. [41] used a cross-recurrence quantification analysis (CRQA) to detect concept drifts. The author's idea was to highlight the most significant hashtag-related events. Cross-recurrence quantification analysis was used to compare the changes in trajectory. This outcome is achieved by assessing the longest diagonal line of two consecutive windows and whether they follow the same generating process over time. All operations occurred inside a system called TSViz. The experiments discussed in the paper were on drift detection related to hashtags from Brazilian politics. The authors concluded that the drifts detected directly trace back facts from the news. According to the authors, recurrence analysis "characterizes the behavior of dynamical systems by reconstructing produced data in phase spaces". The authors used Normalized Compression Distance (NCD) to compute the similarity among texts and Naive Bayes to perform sentiment analysis; however, the authors did not detail the classification process. The results were visually assessed.

Instead of providing a concept drift detector, Zhang et al. [214] provided a framework for concept drift prediction. Although the approach focused on time series, the proposed framework is model-agnostic and monitors loss distribution drift to predict drift occurrence, which could also be interesting for text streaming scenarios. The method was evaluated in a prequential manner, *i.e.*, train-then-test. When receiving new data, their framework makes a prediction with the model, updates the model f_δ, stores (temporarily) the respective data (x, y), stores loss $\mathcal{L}$ in B, and updates the memory bank $\mathcal{M}$. If the length of the B is bigger than a predefined window size, and the z-score considering B and the last window is bigger than a threshold, the fine-tuning process is triggered. The proposal also encompasses a parameter to control the frequency of fine-tuning, even if the aforementioned conditions are not met. Their method obtained the best performances in several of the tested scenarios.

The *concept evolution* problem regards the increase in the number of classes over time. For a model to be updated, it must internally account for these novel classes [53]. Traditional ML methods require prior knowledge of the number of classes. Abid et al. [2][3] proposed a method for text stream clustering based on AIS. These papers were previously referenced in this work. They also managed concept evolution (under the name of novelty detection). These methods addressed the concept evolution problem by cloning and mutating existing clusters, a heuristic of the clonal selection principle. If the novel data do not fit into a cluster, they are sent to the outlier buffer, where they are examined periodically to detect novel classes. Abid et al. [2] evaluated the quality of concept evolution handling using the M_{new} metric, which measures the rate of novel class instances misclassified as from an existing class. In addition, the authors plotted the F1 score, accuracy, and recall over time, demonstrating the emergence of new classes and how their method recovers from concept evolution. The run time was not measured. Abid et al. [3] employed a similar plot as Abid et al. [2] for two datasets. In addition, they plotted the number of existing classes and identified classes by the method over time. The metric M_{new} is also used, and the number of missed classes is computed.

Wang et al. [206] proposed ESACOD, a framework for streaming classification with concept evolution and subject to concept drift. Their work aimed to learn satisfying parametric Mahalanobis-based metrics in real time. According to the authors, the objective was to identify a feature space projection in which its constraints generate properties of cohesion and separation [206]. Cohesion is the ability of data points to occur close to others from the same class. In contrast, separation is the ability of data points to be distant from others from different classes [206]. Their method trains an open-world classifier with a small dataset with an initial metric established. When new data arrives from the stream, the metric is applied to it, generating data in a new feature space, and the prediction is made afterward. If the prediction indicates that the data does not belong to a novel class, the prediction remains unchanged. On the contrary, if the classifier assumes the data are from a potentially novel class, the data are added to a buffer. When this buffer is filled, it is checked for concept evolution and concept drift. An arbitrary percentage (between 0 and 30%) of data with their respective labels is required. Finally, the evolution class metric is computed using paired constraints based on this randomly selected data. Later, a k-means algorithm [125] is applied, and a label propagation [216] method is performed apparently to the other data in the buffer. If a concept drift or concept evolution is detected, a new classifier is trained with the data to replace the older classifier. The authors concluded that their approach could address the challenges of multiple novel class detection and stream classification bound to concept drift and with few labels available. The method was evaluated in terms of accuracy and run time. Concerning concept evolution, the metrics used were M_{new} and F_{new}, which measure the instances of an existing class misclassified as a novel class, A_{new}, which is the accuracy of novel class classification, and A_{known}, which is the accuracy of known class classification.

Regarding *semantic shift detection*, Periti et al. [159] addressed this problem in an incremental way. The authors used incremental clustering techniques (such as affinity propagation) to generate representation clusters in time slices. The word contexts in the past were clustered into several clusters, serving as a memory for posterior observations. To generate representations, the authors tested BERT and Doc2Vec. BERT provided contextual representation, whereas Doc2Vec provided pseudo-contextual embeddings. The approach selected documents in which target words emerged, fine-tuned the embedding model to add new arriving documents, extracted the embeddings, clustered the representations, and refined the clusters by removing clusters of single or old representations. The authors tested their approach using representations generated by BERT and Doc2Vec for two datasets from SemEval 2020: CCOHA and LatinISE. The authors evaluated alternatives based on affinity propagation. The incremental version of the affinity propagation (IAPNA) performed adequately on the LatinISE dataset using BERT representations and on the English dataset using Doc2Vec representations. In contrast, the affinity propagation a posteriori had satisfying results in the opposite situations. The authors were surprised that Doc2Vec obtained decent results and consumed less time than contextual models.

Castano et al. [31] also proposed a variation of the affinity propagation algorithm named APP. The authors evaluated their method against affinity propagation, IAPNA, and used the Iris, Wine, Car, and KDD-CUP datasets. The methods were assessed in terms of purity and NMI, a frequent metric in clustering settings. For the semantic shift detection task, the authors provided a thorough case study based on a diachronic corpus of Vatican publications in Italian containing around 29,000 documents, split into six subcorpora. From the corpora, texts written by Pope John Paul II were removed due to the variety and richness of his documents, according to the authors. Tracking a previously selected word, *i.e.*, *novità* (novelty), the authors could find the use of this word in a negative sense (in the first subcorpora) to its use in the context of innovation in the Catholic Church.

Although not directed to text stream scenarios, Ishihara et al. [93] proposed a metric for semantic shift named semantic shift stability, improving decision-making on when to fine-tune a model. This method consisted of creating word embeddings, setting anchor words, introducing a rotation matrix, and calculating the stability. The stability was calculated using the cosine similarity between words in two rotated matrices. Also majorly unrelated to text stream scenarios, Periti and Tahmasebi [162] extended the problem of semantic shift detection. According to the authors, frequently semantic shift detection in batch scenarios is addressed by considering two periods of reference. Periti and Tahmasebi [162] proposed five methods for tracking semantic shift, considering consecutive time intervals, consecutive time periods, clustering over all time periods, incremental clustering over time periods, and scaling up form-based approaches. From the proposed methods, only incremental clustering over time seems to be suitable for text stream scenarios. Other papers also developed methods for detecting semantic shifts or adapting to them in batch scenarios, *e.g.*, Hofmann et al. [87], Kim et al. [102], Liu et al. [123], to mention a few. Since we are interested in text stream scenarios, we mentioned these papers because they can inspire the development of incremental versions that are suitable for text stream learning, considering its constraints presented in Section 2.

4.5.4 Topic modeling. Topic modeling consists of statistical tools to examine textual data and identify the most relevant terms related to each theme. This approach facilitates the exploration of the interconnections among these themes and their temporal evolution [23]. It is also considered a text mining task [101]. Four selected papers approach *topic modeling* [90, 141, 144, 202].

Murena et al. [141] proposed an approach mixing LDA and ADWIN to overcome the problem of topic modeling in document streams, entitled AWILDA. LDA [24] is a common method for topic modeling. The authors mentioned that LDA had gained much attention, and it also has an online version. However, one problem with the online version is setting window sizes because drifts may happen in a smaller period than the window size. Thus, the authors defined the window with the aid of an ADWIN module, which can assist in determining topic drifts and the new window for LDA to consider. Two classes of algorithms were mentioned: the passive, which updates a model for each observation, and the active algorithms, which attempt to detect the drift and update the model only when the drift is detected. We can draw parallels between these classes of algorithms and the detection methods presented in Section 4.3, *i.e.*, *adaptive* and *explicit*, respectively. The author's idea was to separate the task of topic modeling and topic drift detection. There are two LDA models inside AWILDA: one for language modeling (LDA_m) and the other for drift detection (LDA_d). In this approach, for each document received from the stream, AWILDA reckons the likelihood for LDA_d and adds it to the ADWIN module. If a drift is detected, LDA_m is trained on the subwindow ADWIN selects. LDA_m is updated whenever a new document arrives from the stream. The authors evaluated their proposed method using the perplexity metric for document modeling and the latency between the actual current drift and the detection. According to the authors, perplexity is "used by default in language modeling to measure the generalization capacity of a model on new data" [141]. The authors concluded that AWILDA could recognize all drifts in the synthetic datasets and one version of the real-world dataset. In addition, the method can select the documents window to be used for updating. AWILDA can detect abrupt drifts and works sufficiently for gradual drifts. Compared to online LDA, it worked similarly until a drift occurred. When a drift occurs, AWILDA is retrained, which increases perplexity, but it ultimately outperforms the online LDA.

Hu et al. [90] proposed a short text stream classification method that uses content expansion and includes a concept drift detector. According to the paper, the external sources must satisfy two criteria: to be large and sufficiently rich to comprise most contents in the short text stream that will be classified and highly topic-consistent with the text stream. The method mines hidden information from the external corpus by using LDA because, according to the authors, LDA performs adequately on longer texts. From the LDA model, top representative words for the topics are selected to be added (once or several) times to a short text according to the topic distribution and word probability of belonging to a topic. The topic distribution represents each short text. The method was evaluated regarding accuracy (classification task) and the drifts, using false alarms, missing drifts, and delays. The datasets were arranged to simulate drift; however, the method was unspecified. The authors concluded that their approach surpassed the accuracy of all the competitors, demonstrating more stability. In addition, their approach could recover from drift earlier than other approaches and outperformed the competitors regarding delay and missing drifts.

Van Linh et al. [202] proposed a graph convolutional method for topic modeling, considering short and noisy text streams. The authors leveraged Word2Vec representations and Wordnet knowledge graph to improve the predictions of their method, called GCTM. The authors claimed that their method could balance the knowledge graph and the knowledge obtained from the previous data batch. This ability can be valuable when handling concept drift. GCTM integrates a graph convolutional network (GCN) into an LDA model to exploit a knowledge graph, and both are updated simultaneously in the streaming environment. The authors tested their approach using six short text datasets and two regular text datasets. Using previous knowledge allowed GCTM to output satisfying predictions and recover more quickly from concept drift. The authors simulated concept drift by rearranging the topics sequentially. The metrics selected for evaluation were the Log Predictive Probability (LPP) [86] and the Normalized Pointwise Mutual Information (NPMI) [114]. These methods measure the model generalization and the coherence of the topics, respectively. GCTM was evaluated in two ways: utilizing Word2Vec (GCTM-W2V) and the knowledge from the Wordnet graph (GCTM-WN). GCTM-WN and GCTM-W2V outperformed the competitors in LPP across all the datasets, even in the presence of concept drift. The authors also performed an ablation study.

Nguyen et al. [144] proposed an LDA-based topic modeling approach with mechanisms for balancing stability and plasticity, namely BSP. Stability-plasticity is a dilemma involving maintaining old knowledge (stability) and learning new knowledge (plasticity) [62, 144]. Balancing both prevents concept drift from impacting performance and catastrophic forgetting [144]. The authors used TPS and iDropout combined into an LDA-based topic modeling method. TPS [198] aided the model with external knowledge, *i.e.*, Word2Vec representations. iDropout [146] created variables β^t, updated whenever a new mini-batch is inputted. Because both are different mechanisms, the authors modified the calculation of β to comprise information from both mechanisms. They performed experiments on eight datasets: one long, two regular, and five short-text. The authors compared their method to six different approaches. The hyperparameters were selected using a grid search. Similar to Van Linh et al. [202], the authors contrasted LPP and NPMI. The authors tested using the datasets shuffled and ordered chronologically whenever possible. Their method achieved the best values for LPP in four out of six datasets tested. It is worth noting that their method achieved satisfactory results very rapidly at the highest levels. Their method maintained high levels of performance while using chronological datasets. The authors tested the stability and plasticity by simulating drifts by sorting the topics in order of classes, similarly to Van Linh et al. [202]. BSP could reach the best values when testing for catastrophic forgetting and maintained the highest levels when recovering from concept drift. As in [202], the authors performed an ablation study to understand the impact of some parameters.

4.6 Text Representation Methods

This subsection describes the text representation methods used in the selected papers, aiming at answering the Research Question 3 (RQ3), *i.e.*, *"Which type of token/word/sentence representation is used in the study?"*. Besides collecting the aforementioned types, we also aimed to check how and if they are updated (see Section 4.7). Fig. 11 depicts the three main categories: (i) Embedding-based methods, such as Word2Vec and BERT; (ii) Frequency-based methods, which include Bag-of-Words, TF-IDF; and (iii) Keywords.

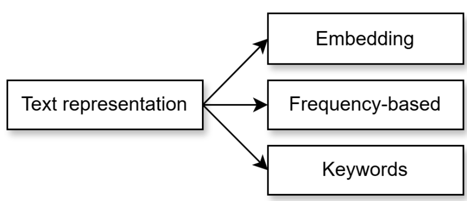

Fig. 11. Categories of text representation methods used in the papers.

In our categorization, we considered *embeddings* the dense vectors generally generated by neural-based approaches, such as Word2Vec, BERT, or even large language models; therefore, in this case, these language models are used as feature extractors [200]. These vectors are capable of representing word semantics and capturing the connotation of words [200]. *Frequency-based* methods are those that resort to methods that leverage word counts and derive text representations. Sometimes, the word counts are used directly as a vector representation, *e.g.*, Bag-of-words, or used as a means to calculate word importance, such as in TF-IDF [79]. In both cases, they are generally used in a structured way because most machine learning methods are not able to handle variable-length input vectors. The category *Keywords* regards the use of words themselves without resorting to vector representations, therefore maintaining a list of keywords to represent items.

Table 7 lists the text representation methods used across the papers. Seven approaches were categorized as *frequency-based*, six as *embedding*, and one as *words*. Two papers have not provided the text representation method, while one provided *file compression*, which cannot be directly classified among the categories but could adapt to *words* because the compression is performed over a file containing a set of words. de Mello et al. [41] used *file compression* and calculated text similarity by using a formula that considers the sizes of the zipped file containing the two texts and the zipped files containing each of the given texts separately. As mentioned, this method is named NCD [35]. Although it appears reasonable for files and images, NCD is also used for texts in some works. For example, NCD is listed as a similarity metric in structured data [151], and in texts [166], even in the presence of noise [32]. At first sight, it may appear unreasonable because two different files containing distinct texts may result in similar file sizes. However, according to [151], if two given files are similar, compressing them together results in an approximate file size to compressing only one. Thus, NCD calculation utilizes this aspect to determine the similarity between two files containing raw text.

Although it did not appear across the studied papers, we acknowledge the existence of other incremental/online methods, such as Hashing Tricks [11]. Beginning from a zero-filled representation vector, Hashing Tricks leverage hash functions to convert tokens into hash values. Each hash value is then divided by the length of a pre-defined representation vector, and the result of the modulo operation defines the index of the representation vector to have its value increased. Although it has no learning at all, it performed competitively in stream scenarios [197]. This method is available in tools such as Vowpal Wabbit [185][12].

Table 7. Text representation used in the studied papers.

Text representation method	**Papers**	**Category**
Bag-of-words [80]	[118][133][33][39][42][16][191][173] [85][27][10][204]	Frequency-based
BERT [46]	[16][7][203][159][119][195][69]	Embedding
Bigram	[173][10]	Frequency-based
Biterm	[90][173]	Frequency-based
Co-occurrences	[210]	Frequency-based
Doc2Vec [115]	[203][159]	Embedding
FastText [26]	[39]	Embedding
GloVe [157]	[193][39][173][144][105]	Embedding
Graph-of-words	[210][202][203][66]	-
Incremental Word Context	[29][65]	Frequency-based
PSDVec [120]	[188]	Embedding
Sent2Vec [136]	[105]	Embedding
TF-IDF [181]	[164][118][84][16][85][27][55][169] [10][48]	Frequency-based
Word2Vec [135]	[81][193][194][206][39][84][202][144] [85][203][119][69]	Embedding
Word frequency	[211][210]	Frequency-based
Words	[141][2][78][3][65][66][110]	Keywords

Regarding the representations, several papers used more than one method, sometimes combined, *e.g.*, Bag-of-words + TF-IDF. However, they were divided in Table 7. In addition, Word2Vec and Bag-of-words (BOW) were used in 12 papers and TF-IDF in 10 papers. Finally, words were used directly in seven papers as a representation method. We briefly described the methods as follows, considering the chronological order of each method.

4.6.1 Bigram. Bigram adheres to the Bag-of-words concept, in which it is possible to organize texts in two dimensions: columns as words and rows corresponding to documents. The cells contain the count of a given word in a specific document. The difference is that a pair of sequential words is represented in each column instead of the words. For example, the sentence "he has been here" will generate three columns: (he, has), (has, been), and (been, here). The challenge incurred from utilizing bag-of-words in streaming scenarios also happens to bigrams, *i.e.*, the dimensions regard fixed words and do not evolve. Rakib et al. [173] used three representation methods while testing their proposed method for short-text stream clustering: unigram, *i.e.*, bag-of-words, bigram, and biterm. Assenmacher and Trautmann [10] also used both unigram and bigram for text representations.

[12]Available at https://github.com/VowpalWabbit/.

4.6.2 *Biterm.* According to Hu et al. [90], a *biterm* corresponds to unordered word-pair co-occurrences. Furthermore, Hu et al. [90] highlighted that biterms were more sparse than regular bag-of-words and utilized external sources to reduce the sparseness. Considering the biterm definition and using the same example as in a bigram, the biterms generated from the sentence "he has been here" would be (he, has), (he, been), (he, here), (has, been), (has, here), and (been, here). Considering the text stream scenario, it encounters challenges similar to those of bag-of-words and bigrams. To overcome this, Hu et al. [90] developed an ensemble based on base learners trained using data chunks, each with its biterm topic model. Rakib et al. [173] also used biterm as text representations. To evaluate their short-text stream clustering method, the authors used biterm, unigram, and bigrams. Biterms performed better than bigrams and unigrams, considering NMI values.

4.6.3 *Co-occurrences.* Co-occurrences count simultaneous occurrences of two particular words. Yang et al. [210] developed a bi-weighted word relation network that considers both the co-occurrences and the word frequencies. Although co-occurrences and word frequencies are not representations, we opted to include them as a single representation because they will be part of a graph, *i.e.*, graph-of-words, which is an actual representation.

4.6.4 *Graph-of-words.* Graph-of-words (GOW) is a textual representation that transforms documents into graph-based structures [203]. According to the author, it can maintain long-term relationships between words. After generating the graphs regarding specific documents, frequent subgraph mining techniques were applied, and later, the mined frequent subgraphs were used as feature representations. In [203], GOW had two parameters: sliding window and minimum support. GOW appears to have the capability of being updated in real time. However, its use with a pre-trained Word2Vec model (that can be outdated after an arbitrary period) made the process not fully incremental. Although they did not use the terminology *graph-of-words*, Yang et al. [210] developed a corpus-level word relation network, namely EWNStream+, which retained the co-occurrence counts and word frequencies. According to the authors, EWNStream+ is incremental by receiving data batches. Van Linh et al. [202] proposed a novel graph convolutional topic model (GCTM) based on graph convolutional networks and LDA. The initial graph was formed using words and their relations. GCTM was tested using Word2Vec representations and WordNet. GCTM did not support incremental-fashioned training, implying that the text models could become obsolete.

4.6.5 *Word frequency.* Word frequency is the word count. Yang et al. [210] included word frequency as part of their word relation network, which also considered the word co-occurrences. This representation was also used to determine whether a word was outdated in the graph representation.

4.6.6 *Keywords.* Several papers chose to use the words themselves rather than any text representation. In this case, since the words are not structured as in a bag-of-words representation, for example, we named it *keywords*. Murena et al. [141] presented AWILDA, an LDA-based method integrated with ADWIN for topic drift detection. The authors used the keywords lowercased and stemmed. Abid et al. [2][3] described AIS-Clus, an incremental clustering method. Initially, the authors used DBSCAN [52] to generate the cluster, and then sketches were developed to summarize each cluster. The sketches contained lists of keywords and outliers present in a cluster. Hammer and Yazidi [78] presented a method to handle concept drift in an abruptly changing environment. The authors used keywords to monitor probabilities in topics. Considering the updating scheme, keywords could easily be added or removed from sketches. Therefore, we considered it possible to use it in streaming scenarios, although it could become complex and time-consuming to maintain a list of keywords in every sketch, as demonstrated in Murena et al. [141] and Abid et al. [3], if not limited to respecting the constraints of data stream environments.

4.6.7 Bag-of-words. Bag-of-words [80] is probably one of the simplest methods for text vectorization, as it divides the text into tokens or words. Considering rows and columns, these tokens function as columns while the rows represent each text, such as tweets. There will be the counts of the tokens corresponding to a particular column in a text corresponding to a specified row in each cell. An evident characteristic is that bag-of-words representation in a unigram way does not represent the order of words, which can be leveraged in some applications. In streaming scenarios, it inhibits ML methods from performing properly. For example, suppose a bag-of-words representation is generated whenever each new text is inputted. In that case, the number of columns may increase, and most ML methods cannot handle dimension-changing inputs. Furthermore, even if the process runs in batches, the words of the bag-of-words may change. If the first batch defines the words for the bag-of-words representation, it may not recognize changes and new words, *i.e.*, new dimensions, over time.

4.6.8 TF-IDF. Term-Frequency-Inverse Document Frequency (TF-IDF) is a statistic from the information retrieval area used for determining the importance of words to a document or a set of documents [181]. The calculation considers the frequency of a term and the inverse document frequency, which defines how informative a term is across several documents. Generally, TF-IDF is used in the stream setting to encode data batches. It is worth noting that the term frequency calculation is remarkably similar to the bag-of-words procedure. Thus, it is common to discover the use of bag-of-words with TF-IDF. Pohl et al. [164], Bechini et al. [16], and Bondielli et al. [27] used TF-IDF after obtaining a data batch to encode the terms and the texts from the stream. Li et al. [118] utilized TF-IDF to generate vector representations from the data batches so that a base learner could be trained and incorporated into the ensemble. Heusinger et al. [84] and Heusinger et al. [85] performed TF-IDF in an offline mode to generate a very high-dimensional vector so that they could test their dimensionality reduction strategy. Mohawesh et al. [137] executed TF-IDF before all the processing. Later, the authors employed PCA to reduce the dimensionality of the datasets by selecting the 10,000 most meaningful components. Since TF-IDF works together with bag-of-words, it is impossible to update it incrementally without changing the number of dimensions. Assenmacher and Trautmann [10] used TF-IDF to decide the proximity of incoming text to existing microclusters in the online phase. This calculation is also used in the offline phase, particularly when evaluating the merging of existing clusters. Fenza et al. [55] used TF-IDF representations to generate the fuzzy lattice structure. Rabiu et al. [169] leveraged TF-IDF to compute the input vectors to train base learners. An interesting aspect regards preprocessing in [169]: the authors utilized the Stanford CoreNLP [128] to segment words, part-of-speech tagging, and stemming. The authors used only the first three tags of noun, verb, and adjective. According to the authors, these tags "carry the most valuable information regarding reviewed items". However, no evidence is provided.

4.6.9 Word2Vec. Word2Vec [134] corresponds to two distinct model architectures for learning distributed representations: Continuous Bag-of-words (CBOW) and Skip-gram. Both are neural network architectures, where the number of neurons is the same in the input and output layers, and the single hidden layer corresponds to the embedding size. Each neuron in the input and output layers can correlate to the words in the vocabulary. The representations, after training, are often obtained by taking the connection weights between a neuron (representing a word) in the output and the hidden layers. The difference between CBOW and Skip-gram is the training step aim: CBOW aims at predicting a specific word given its surrounding words, whereas Skip-gram does the opposite, *i.e.*, predict the word in the middle based on the surrounding words [134]. The papers that utilized Word2Vec used it for text representation only. Li et al. [119] leveraged Word2Vec for reduction of data sparsity. The authors developed their method for short-text classification, and one of the general approaches for this problem was to enrich the data. The authors evaluated both

Word2Vec and BERT for the short-text representation, which was later applied to a CNN to extract higher-level feature information. Although Word2Vec is a neural architecture, it has incremental versions by using gensim[13] [176] or other methods in the literature [94, 95, 129].

4.6.10 Doc2Vec. Le and Mikolov [115] proposed Doc2Vec to obtain documents as distributional vectors. Doc2Vec is a generalization of Word2Vec. Similarly to Word2Vec, Doc2Vec is constituted by two architectures: Paragraph Vector - Distributed Memory (PV-DM) and Distributed bag-of-words version of Paragraph Vector (PV-DBOW). In PV-DM, the document vectors are trained with the word vectors in the architectures, while in PV-DBOW, the aim is to predict the words of a document from a document ID. Periti et al. [159] used a Doc2Vec model trained with the CCOHA and LatinISE datasets. The model was not updated during the process and may become obsolete as time passes. It was unclear whether Vo [203] utilized a pre-trained model, trained a model from scratch, or if the model was updated over time. Since Doc2Vec is a neural architecture, training and updating it can be computationally costly.

4.6.11 GloVe. Global Vectors (GloVe) is a method for generating co-occurrence-based word vector representations [157]. According to the authors, GloVe utilizes global matrix factorization and local context window methods. The method is trained in a batch manner. D'Andrea et al. [39], Rakib et al. [173], and Nguyen et al. [144] used GloVe for semantic representation by using pre-trained models. Kolajo et al. [105] claimed that GloVe is used for feature extraction. Suprem and Pu [193] mentioned that the proposed system, *i.e.*, Adaptive Social Sensor Event Detection (ASSED), supports GloVe. The authors in the original paper [157] did not describe any incremental or adaptive training. Therefore, the vector representations can become outdated over time, constituting a potential disadvantage in streaming scenarios.

4.6.12 FastText. FastText [26], an extension of the Skip-gram method, is one of the Word2Vec architectures. Instead of accounting for the entire words, FastText considers subword partitions using n-gram vectors. Using an example from the original paper, encoding the word *where* in a 3-gram fashion results in a 5-sized vector containing (wh, whe, her, ere, re). In addition, the approach incorporates the word *where* integrally. This method of splitting words in n-grams helps the model handle words unseen in the training step, also named out-of-vocabulary (OOV) words. An incremental update method is not mentioned in the paper. D'Andrea et al. [39] utilized a pre-trained FastText model [26] as a text encoding method. In addition, FastText is used statically, implying that no method is presented in D'Andrea et al. [39] for the incremental update of the text representations. However, D'Andrea et al. [39] concatenated FastText representations to *bag-of-words* representations generated in each step of an incremental procedure of accumulating data from past events.

4.6.13 BERT. Bidirectional Encoder Representation from Transformers (BERT) is a multi-purpose language model that enables several NLP tasks [46], such as sentiment analysis, sequence-to-sequence, paraphrasing, and question answering. In addition, BERT can provide vector representations of text to be used in a particular downstream task. Bechini et al. [16] used an Italian version of the pre-trained BERT model, *i.e.*, AlBERTo [165], for measuring semantic similarity between tweets. In [7], BERT was the primary model. The authors tested different sampling methods for fine-tuning to pursue an incremental update of the model. BERT was also used as a text encoding method in [203], where the authors enhanced short-text clustering by combining pre-trained BERT's representations with a BiLSTM and a graph-of-words representation. Periti et al. [159] used BERT for word representation generation in both English and Latin by using pre-trained models.

[13] https://radimrehurek.com/gensim/

Considering the aforementioned papers, only Amba Hombaiah et al. [7] had an updating scheme for the representations. It was achieved by using fine-tuning strategies, which could enable the use of BERT in streaming scenarios, but it may also become a bottleneck in the process. Susi and Shanthi [195] leveraged BERT and variations in two moments. First, the authors used a pre-trained RoBERTa model [124] specifically suited for sentiment classification. The RoBERTa model enabled automated training data generation. However, another BERT model was fine-tuned in the system whenever a sentiment drift happened. Li et al. [119] used BERT to enrich short texts. Short texts are very sparse, and, according to the authors, using embeddings may improve the representation quality.

4.6.14 Sent2Vec. Moghadasi and Zhuang [136] proposed a sentence embedding method that considers the sentiment score behind the sentence. Kolajo et al. [105] used the Sent2Vec embeddings to compute the semantic representation of the input texts and then cluster these texts. If a new tweet was different from the histograms of the clusters, a concept drift was deemed to have occurred, and a new cluster was created for it. Kolajo et al. [105] did not describe an updating scheme. Thus, the Sent2Vec model can become obsolete over time, necessitating retraining.

4.6.15 Incremental Word Context. Bravo-Marquez et al. [29] proposed a vector representation method for texts that can be considered a table-like representation, with the columns corresponding to words and rows similarly corresponding to words. However, the column (in the original paper, called context) and the words (called vocabulary) can have different sizes. The number of contexts defines the dimension size of the vector representation. The authors calculated the positive pointwise mutual information (PPMI) in each cell, considering the words and their co-occurrences. Although the vocabulary (rows) can be updated, similarly to bag-of-words, if the contexts are fixed, the system may incur obsolescence after the context words decrease or stop appearing. Furthermore, if certain context words are exchanged with other words, the changed dimensions will not represent the same contexts, and this will be reflected in an ML model dependent on vector inputs.

4.6.16 PSDVec. Li et al. [120] proposed the Positive-Semidefinite Vectors (PSDVec) as a toolbox for incremental word embedding. PSDVec is an eigendecomposition-based method. Similarly to Incremental Word Context, PSDVec uses a pointwise mutual information matrix. According to the authors, PSDVec has several advantages, including the ability to learn new words incrementally based on an original vocabulary. In their experiments, Li et al. [120] reached good results in the word similarity and analogy tasks.

In this subsection, we analyzed the text representation methods used in the selected papers. However, we did not extrapolate the same analyses to incremental versions. Thus, when we discussed that a particular method only worked at least in batches, we did not extend the same conclusions to other versions, including incremental/adaptive versions when available. Although not listed among the text representation methods found across the selected works, recently studied alternatives that could enable concept drift detection can be encountered in the literature, such as lexical replacements [158], word senses representations [70], and the use of large language models (LLMs) for topic modeling [139].

4.7 Updating Mechanism of Text Representation Methods

We also considered the updating mechanism of the text representation methods. Observing how the text representation behaves over time in text stream scenarios is critical. Because of stream characteristics, *i.e.*, fast and potentially infinite, a static text model is a problem. It is even severe in text stream scenarios under concept drift because a representation vector may become obsolete, losing quality and, thus, negatively impacting the stream mining task. Therefore, we also obtained information on the text representation updating method. Fig. 12 depicts the organization regarding the updating scheme of text representation methods. We organized in two dimensions: *incremental* and *non-incremental*. In *Incremental*, we considered that the representation method can be updated over time, whether in batches or instances. In *Non-Incremental*, we assumed that the text representation method was either static during the entire process or required complete retraining to be updated.

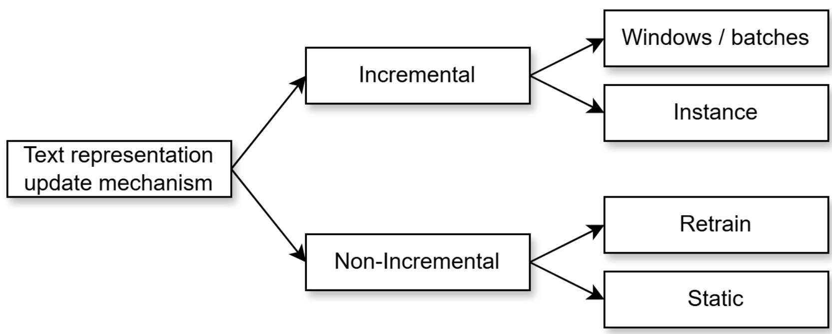

Fig. 12. Categories of mechanisms for text representation updating found in the selected papers.

4.7.1 Incremental. We list text representation methods with incremental update capabilities organized in *windows/batches* or *instance*. Considering the update in windows/batches, this indicates that the text representation method requires a new amount of data to either be worth updating or satisfy a specific constraint of the text representation method. Using BERT as in [7] and [195] are examples of this category. Amba Hombaiah et al. [7], the BERT model is fine-tuned using texts selected by the sample methods proposed by the authors. Susi and Shanthi [195] perform the fine-tuning through an updated training set. The training set is updated whenever a sentiment drift is deemed to have occurred.

Considering the incremental methods that can be updated in instances, it implies that it is unnecessary to accumulate data to update the text representation method: a single piece of information can be used for that. For example, we mention Incremental Word Context [29]. Furthermore, given a single new input, the Graph-of-Words [210] can be updated in real time.

4.7.2 Non-Incremental. Considering the text representation methods that do not allow any update but are retrained from scratch, we list bag-of-words, bigrams, biterm, and TF-IDF. However, while in use, a few text representation methods were kept static in the text streams: FastText, Doc2Vec, and GloVe. Most were used as pre-trained models, and they can become obsolete after some time, demanding complete retraining to maintain the performance of the dependent ML model. Heusinger et al. [85], Vo [203] and Li et al. [119] also leveraged static BERT and Word2Vec models.

5 DATASETS

Recalling the Research Question 4 (RQ4), *i.e.*, *"Which datasets were used to evaluate the proposed approach(es)?"*, we also included a list of real-world datasets to which the methods for stream mining tasks from the selected papers were applied. The synthetic datasets were excluded since they are generally numeric or contain a sequence of unrecognizable topics. Considering Table 8, several datasets were used; however, most appeared in only one paper. In addition, some papers that shared datasets in common frequently shared authors (or co-authors) or the task, *e.g.*, short-text classification and topic modeling. All the links in the column *Information / Access* were verified on 19th September 2024. In addition, some datasets were flagged as *obtained by the authors*. It means that the authors collected the datasets, either manually or through APIs, but the datasets are not publicly available for download.

Regarding the datasets as depicted in Table 8, some may share the same name, such as Twitter, and New York Times. However, it was impossible to assert that they are the same dataset. Thus, we added a new line in the table instead of aggregating data regarding a particular dataset. In addition, at least three mechanisms were referred to as API providers for data collection: Twitter[14], The Guardian[15] and The New York Times[16]. Thus, since the queries can be performed ranging from different dates and keywords, the datasets of the same name may correspond to different datasets.

5.1 Datasets description

Below we provide short descriptions of each dataset listed in Table 8. We highlight that some datasets included raw texts, while a few contain the bag-of-words representation of texts, *i.e.*, preprocessed texts.

5.1.1 20NewsGroup. This dataset contains approximately 20,000 news across 20 groups. In the link provided in this paper, there are three versions of this dataset, with slight variations.

5.1.2 Arxiv. According to [127], this dataset contains approximately 2 million abstracts of papers published comprising the years between 2007 and 2021.

5.1.3 CLINC150 and CLINC150-SUR. In the context of task-oriented dialog systems, CLINC150 [113] is a crowdsourced dataset containing 22,500 in-scope queries regarding 150 intents from 10 general domains and 1,200 out-of-scope queries. CLINC150-SUR [168] is an extension of the CLINC150 dataset, in which Rabinovich et al. [168] generated more instances, added rephrased instances (generated with Parrot [38], and upsampled with LAMBADA [8], reaching 600,000 instances.

5.1.4 CrisisLexT26. Pohl et al. [164] cited that CrisisLexT26 [150] is a collection of datasets related to several crises worldwide. However, Pohl et al. [164] used only the datasets related to the Colorado Floods, containing 751 relevant and 224 irrelevant tweets, and Australian Bushfires, containing 645 relevant and 408 irrelevant tweets.

5.1.5 EmailingList. This dataset contains 1500 samples with 913 dimensions, *i.e.*, boolean bag-of-words, corresponding to email messages, to be classified as junk or interesting. According to Katakis et al. [98], these samples were collected from Usenet posts existing inside the 20Newsgroup dataset.

5.1.6 EveTAR. EveTAR is an Arabic dataset that contains 1392 tweets on three terrorist events: (i) a suicide bombing in Ab, Yemen; (ii) Air strikes in Pakistan; and (iii) the Charlie Hebdo attack. Abid et al. [3] used this dataset to evaluate the ability of AIS-Clus to receive texts and detect events in languages other than English.

[14]https://developer.twitter.com/en/docs/twitter-api
[15]https://open-platform.theguardian.com/
[16]https://developer.nytimes.com/apis

Table 8 List of datasets used in the papers and their respective resources, when available.

Dataset	Papers	Information / Access	Stream Mining Tasks
20NewsGroup	[170]	http://qwone.com/$\sim$jason/20Newsgroups/	Short-text clustering, Classification
	[203]		
	[169]		
	[188]		Multilabel classification
	[69]	Versions with gradual and abrupt drifts generated by the authors	Classification
Arxiv	[127]	https://www.kaggle.com/datasets/Cornell-University/arxiv	Classification
CLINC150	[168]	https://github.com/clinc/oos-eval [113]	Short-text classification
CLINC150-SUR	[168]	Based on the original CLINC150 dataset, with simulated user requests https://huggingface.co/datasets/ibm/clinic150-sur	Short-text classification
CrisisLexT26	[164]	obtained from https://archive.org/details/twitterstream [150]	Crisis management
EmailingList	[133] [42]	http://mlkd.csd.auth.gr/datasets.html	Classification
EveTAR	[3]	http://qufaculty.qu.edu.qa/telsayed/evetar	Event detection
Guardian, The	[206]	obtained by the authors	Classification
Irish Times, The	[202] [144]	https://www.kaggle.com/datasets/therohk/ireland-historical-news	Topic modeling
NELA-GT-2018	[55]	https://doi.org/10.7910/DVN/ULHLCB [149]	Classification
NELA-GT-2019	[55]	https://doi.org/10.7910/DVN/O7FWPO [72]	Classification
NELA-GT-2020	[55]	https://doi.org/10.7910/DVN/CHMUYZ [73]	Classification
New York Times, The	[206]	obtained by the authors	Classification
	[81]	https://ir-datasets.com/nyt.html	Classification
	[202]	http://archive.ics.uci.edu/ml/datasets/Bag+of+Words	Topic modeling
	[127]	https://www.dropbox.com/s/nifi5nj1oj0fu2i/data.zip?dl=0	Classification
NOAA	[193] [192] [194]	not provided but probably from https://data.noaa.gov/dataset/	Event detection
NSDQ	[84] [85]	https://github.com/ChristophRaab/NASDAQ-Dataset	Classification
OffensEval	[7]	https://competitions.codalab.org/competitions/20011	Classification
RCV1	[81]	Available via Scikit-learn library[17].	Classification
Reuters-21578	[141]	https://archive.ics.uci.edu/ml/machine-learning-databases/reuters21578-mld/	Topic modeling
	[188]		Clustering, classification

[17] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_rcv1.html

Table 8 List of datasets used in the papers and their respective resources, when available. *(continued)*

Dataset	Papers	Information / Access	Stream Mining Tasks *
SemEval2020 - Subtask 2 (CCOHA)	[159]	https://www.english-corpora.org/coha/ [6, 183]	Semantic Shift Detection
SemEval2020 - Subtask 2 (LatinISE)	[159]	https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2506 [130, 183]	Semantic Shift Detection
SO-T	[173]	obtained by the authors	Short-text clustering
SpamAssassin	[42]	http://mlkd.csd.auth.gr/datasets.html	Classification
SpamData	[133] [33] [42]	http://mlkd.csd.auth.gr/datasets.html	Classification
Ts-T, Tw, Tw-T, Tweets, Tweets-T	[211] [173] [203] [10]	https://trec.nist.gov/data/microblog.html	Short-text clustering
Tweets, TweetSet, Twitter	[118]	obtained by the authors	Short-text classification, Event detection
	[191]	obtained by the authors	Short-text classification, Event detection
	[105]	obtained by the authors	Short-text classification, Event detection
	[39]	obtained by the authors	Stance detection
	[16]	obtained by the authors	Stance detection
	[27]	obtained by the authors	Stance detection
	[7]	https://archive.org/details/twitterstream	Classification
	[210]	obtained by the authors	Short-text clustering
	[41]	obtained by the authors	Concept drift detection
	[119]	[205]	Short-text classification
	[10]	obtained by the authors	Short-text clustering
	[195]	obtained by the authors	Sentiment drift detection
	[188]	https://github.com/jackyin12/GSDMM/	Clustering, classification
	[65]	https://github.com/cristianomg10/sentiment-drift-analysis-text-stream-football	Short-text classification, Sentiment drift detection
	[66]	https://github.com/cristianomg10/temporal-analysis-of-drifting-hashtags-in-textual-data-streams-a-graph-based-application	Topic modeling, Clustering
	[48]	obtained by the authors	Classification

Table 8 List of datasets used in the papers and their respective resources, when available. *(continued)*

Dataset	Papers	Information / Access	Stream Mining Tasks *
TwitterSentiment	[133]	https://bit.ly/twitter-sentiment-link	Classification
UCI News	[144]	https://www.kaggle.com/datasets/uciml/news-aggregator-dataset	Topic modeling
Usenet1	[42]	http://mlkd.csd.auth.gr/datasets.html	Classification
Usenet2	[42]	http://mlkd.csd.auth.gr/datasets.html	Classification
USGS	[193] [192] [194]	not provided but probably from https://www.usgs.gov/products/data	Event detection
vg.no	[78]	obtained by the authors	Classification
Yelp datasets	[137]	https://www.yelp.com/dataset	Fake reviews detection
Y-Art, Y-bus, Y-com, Y-Edu, Y-Ent, Y-Soc	[110]	https://www.uco.es/kdis/mllresources/ [121, 142, 145, 152, 175]	Multilabel classification

5.1.7 Guardian, The. Wang et al. [206] collected a news stream from The Guardian using the API. The dataset contains 10 categories and 40,000 samples, represented using Word2Vec with 300 dimensions.

5.1.8 Irish Times, The. The Irish Times dataset corresponds to a set of 1.6 million news headlines published by the Irish Times, distributed in six classes. It comprises 25 years of publications.

5.1.9 NELA-GT. NELA-GT [72, 73, 149] corresponds to a series of datasets regarding news and media outlets. In addition, conspiracy sources are included in this dataset. The authors incorporated ground-truth ratings of aspects such as reliability, transparency, and bias. NELA-GT-2018 [149] contains 713 thousand items from 194 media outlets and conspiracy sites; NELA-GT-2019 [72] contains 1.12 million media articles from 260 mainstream and alternative sources collected in 2019; NELA-GT-2020 [73] contains almost 1.8 million news stories from 519 sources. Fenza et al. [55] used these datasets in the fake news detection, using the instances labeled as reliable and unreliable. The datasets were merged, but the temporal order was respected.

5.1.10 New York Times, The. Wang et al. [206] used The New York Times' public API to collect news articles between January 2006 and January 2018. These news articles were encoded using Word2Vec, with 300 dimensions. He et al. [81] used a dataset collected from The New York Times, containing news articles from 1987 and 2007, distributed in 26 categories [182]. Van Linh et al. [202] used only the title of news articles from the New York Times. The authors mentioned that the dataset contained 1,764,127 titles, with an average of five words per title. Lu et al. [127] utilized a dataset collected from the News York Times containing 99,872 articles dating from 1990 to 2016.

5.1.11 NOAA. The National Oceanic and Atmospheric Administration (NOAA) is an agency in the United States government. It does not correspond directly to a dataset; however, Suprem and Pu [193][194] and Suprem et al. [192] used NOAA reports as ground truth for the automatic classification of tweets. No details were offered about the reports' processing or collection.

5.1.12 NSDQ. The NSDQ dataset (named after NASDAQ) corresponds to tweets regarding 15 companies listed in NASDAQ. NSDQ was compiled by the authors in Heusinger et al. [84] and Heusinger et al. [85] and comprised the months of February to December 2019. This dataset contains 30,278 tweets.

5.1.13 OffensEval. Amba Hombaiah et al. [7] used the OffensEval 2019 dataset [213]. The dataset contains 14,000 tweets posted in 2019, categorized into offensive and inoffensive.

5.1.14 RCV1. RCV1 [117] is a dataset that contains 403,143 news from Reuters News between 1996 and 1997. The news articles are divided into three classes: industries, topics, and regions. This dataset is organized hierarchically. From this dataset, He et al. [81] obtained a corpus with 12 subtrees (labels).

5.1.15 Reuters-21578. Murena et al. [141] used this dataset, which contains articles with their respective categories temporally ordered. According to Murena et al. [141], it contains 12,902 news, each classified into several categories, totaling 90 categories.

5.1.16 SemEval2020 - Subtask 2. Periti et al. [159] used the datasets corresponding to Task 2 of SemEval2020, regarding the semantic shift detection task. The datasets used were CCOHA [6] and LatinISE [130]. CCOHA contains texts in English that range from approximately 1810 to 2000, while LatinISE has Latin texts that range from the 2nd century BC to the 21st century AD. Both have target words, which are words that can be monitored to detect the semantic shift. These datasets were discovered in the selected papers that span the longest.

5.1.17 SO-T. Rakib et al. [173] collected duplicated question titles regarding Python, Java, jQuery, R, and other programming languages/tools. In the paper, the authors carefully described the process of obtaining this dataset. In the end, this dataset contained 400,000 randomly selected pairs of question titles.

5.1.18 SpamAssassin and SpamData. These datasets correspond to emails collected from the *Spam Assassin* collection. They are represented as bag-of-words, distributed across two classes, ham and spam, in imbalanced proportions (80% and 20%, respectively). Both contain 9,324 instances; however, SpamAssassin [97] has 40,000 features, while SpamData has 499 [96]. It is noted that these datasets contain gradual drifts [42].

5.1.19 Ts-T, Tw, Tw-T, Tweets, Tweets-T. Yin et al. [211] used this dataset named *Tweets*, containing 30,332 tweets distributed into 269 groups, with 7.97 words per tweet on average. The authors also generated a variant dataset from *Tweets*, called *Tweets-T*, where the dataset is sorted by topic. Rakib et al. [173] used the same dataset *Tweets*, called *Ts-T*. Vo [203] named the same datasets presented in [211] as *Tw* and *Tw-T*, respectively.

5.1.20 Tweets, TweetSet, Twitter. Li et al. [118] used a Tweets dataset containing approximately 400,000 tweets. They stated that the data acquisition comprises November and December 2012, using the Twitter API. Sun et al. [191] also obtained a dataset through Twitter API and consists of 803,613 short texts distributed in four categories. Kolajo et al. [105] described the dataset used in their work as “Twitter sentiment analysis training corpus”, from which they filtered 10% of the data, totaling 104,857 tweets. D'Andrea et al. [39] collected tweets by using a Java library named GetOldTweets[18]. They collected 112,397 tweets posted between September 2016 and January 2017, using vaccine-related keywords. Bechini et al. [16] extended the dataset obtained in [39] until September 2019, corresponding to 806,672 tweets. Bondielli et al. [27] collected 486,688 tweets from July 2021 to December 2021 regarding the Green Pass, as the European Union COVID-19 Digital Certificate is known in Italy. Amba Hombaiah et al. [7] used tweets to perform country hashtag prediction in two different years: 2014 and 2017, consisting of 472,000 and 407,000 tweets, respectively. The tweets were obtained from the Internet Archive[19]. Yang et al. [210] experimented with their approach using a Twitter dataset, namely *TweetSet*, containing about 144,000 tweets posted in June 2019, distributed into 16 categories. de Mello et al. [41] collected tweets by monitoring a set of users and hashtags, *i.e.*, words with a # at the beginning that simulate a tag for the tweet. The authors monitored, for instance, @dilmabr (former Brazilian president) and #dolar (Portuguese for dollar). The dataset size was not mentioned. Garcia et al. [65] collected tweets regarding a specific soccer match between two South American clubs in an international cup. This dataset contains 37,126 tweets, and this is one of the very rare datasets with labeled drifts. Garcia et al. [66] collected tweets comprising 2018 to 2022, totaling 255,131 tweets. These tweets were related to the hashtag #mybodymychoice, and, in this specific study, the authors performed a community detection algorithm over an incremental graph method to detect hashtag drifts.

Although Twitter-based datasets were very frequent in the studied papers, as of February 2023, Twitter's API policies have changed[20], and it became a paid service.

5.1.21 TwitterSentiment. TwitterSentiment (or TSentiment, as in [133]) is a balanced dataset that contains 1.6 million tweets collected between April and June 2009. These tweets are labeled as positive or negative using distant supervision. In this case, emoticons were used for labeling.

[18] https://github.com/Jefferson-Henrique/GetOldTweets-java/
[19] https://archive.org/details/twitterstream
[20] Available at: https://www.forbes.com/sites/jenaebarnes/2023/02/03/twitter-ends-its-free-api-heres-who-will-be-affected/?sh=36ad308a6266. Accessed on September 17th, 2023.

5.1.22 UCINews. The UCINews dataset contains 422,937 news collected between March and August 2014. Each news item can be categorized as business, science and technology, entertainment, or health. This data collection also includes each news id, title, URL, publisher, story id, hostname, and timestamp information.

5.1.23 Usenet1 and Usenet2. Similarly to EmailingList, both Usenet1, and Usenet2 simulate a sequence of 1500 emails from the 20NewsGroup dataset to a particular user to be classified as junk or interesting [97]. According to de Moraes and Gradvohl [42], both datasets have 100 features corresponding to words.

5.1.24 USGS. United States Geological Survey (USGS) is a scientific agency from the United States. Similarly to NOAA, USGS reports do not correspond to datasets and are also used as ground truth to classify tweets automatically by Suprem and Pu [193][194] and Suprem et al. [192].

5.1.25 vg.no. Vg.no is a Norwegian news website. Hammer and Yazidi [78] obtained news from four topics: European Union, economy, sports, and entertainment. However, the authors did not mention the size of the collected dataset.

5.1.26 Yelp datasets. Mohawesh et al. [137] used four real-world datasets based on the datasets provided by Yelp, namely Yelp CHI, Yelp NYC, Yelp ZIP, and Yelp Consumer Electronics. The authors used Yelp CHI (Chicago) [140], containing more than 67,000 reviews of restaurants and hotels distributed between 2004 and 2012. Yelp NYC [174] contains approximately 322,000 reviews of restaurants in New York City. It comprises the years between 2004 and 2015. Yelp ZIP [174] contains 608,598 reviews from New Jersey, Vermont, Connecticut, and Pennsylvania. Yelp Consumer Electronics [13] contains almost 19,000 records evenly distributed between genuine and fake. These datasets include other data, such as user information, product information, rating, timestamp, and review, and were scraped/downloaded from Yelp.com.

5.1.27 Y-Art, Y-bus, Y-com, Y-Edu, Y-Ent, and Y-Soc. Kumar et al. [110] leveraged the datasets Y-Art, Y-bus, Y-com, Y-Edu, Y-Ent, and Y-Soc. These datasets are based on Yahoo, and each class has second-level categories. All datasets can be found in the link provided in Table 8, together with several multilabel datasets.

Although SpamAssassin and EmailingList have known concept drifts (gradual and abrupt)[21], an interesting aspect is that only two datasets across the papers analyzed, *i.e.*, Garcia et al. [65] and Ghahramanian et al. [69], have labeled concept drifts, due to the difficulty of defining the specific points of drift, which requires a deep study on a particular dataset. Thus, some works attempted to force concept drifts by: (i) placing data partitions temporally disordered in a stream, *i.e.*, data from 2011 and 2015 before 2012 [137]; or (ii) rearranging the data, sorting by classes or topics [118]. This aspect is extended in Section 6.

Therefore, since we could not locate repeating datasets in more than three papers, we can conclude that the research area of concept drift detection in textual streams lacks benchmark datasets. Furthermore, all the datasets used for classification are instance-level labeled, *i.e.*, sentences/tweets labeled. In addition, the resource of one of the most recurrent datasets, *i.e.*, *TagMyNews* and *Snippets* [163], could not be encountered across the papers. Also, it is closely related to short-text applications, which constitutes an entirely new research area.

6 CONCEPT DRIFT VISUALIZATION AND SIMULATION

It is challenging to clearly express or prove the existence of concept drifts in a particular textual dataset. However, a few works attempt to justify the existence of drifts by resorting to plots. In this

[21]According to http://mlkd.csd.auth.gr/concept_drift.html

section, we only provide references for the figures due to copyright restrictions. For example, Li et al. [118] used normalized stacked bar plots to demonstrate the topic distribution over several batches (Figure 4 in [118]).

Bondielli et al. [27] plotted the distribution of the stances across the analyzed period using a normalized stacked area plot, similar to the stacked bar plot, to show the topic distribution over time. The background color regards the stance of tweets about the Green Pass, distributed in positive (in blue), neutral (in white), and negative (in red). Considering the color code aforementioned, the thicker line corresponds to the average stance at each moment in the timeline. This description relates to Figure 3 in [27]. Similarly, Garcia et al. [65] depicted the sentiment distribution regarding a soccer match over time (Figure 4 in their paper) by using a stacked area plot. In addition, the authors visualized the sentiment drift splitting the match into quarters, *i.e.*, Figure 3 in [65].

However, Suprem and Pu [193] and Heusinger et al. [84][85] used dimensionality reduction methods, *i.e.*, either t-SNE or PCA, to reduce high-dimensional representations to two dimensions, which can easily be plotted. Thus, Suprem and Pu [194] and Heusinger et al. [84][85] used t-SNE to confirm that there are drifts between texts of specific hashtags. Figure 4 in [84] depicts the visual representation of concept drift. The data points of different colors in different positions indicate that texts regarding particular stock tickers have different patterns. However, it does not highlight temporal changes.

Suprem and Pu [193] used PCA for dimensionality reduction for plotting and suggesting a direction of drift based on data from 2014 and from four months in 2018. It is not possible to categorize the drifts shown by the images considering the literature presented in Section 2. Figure 10 in [193] is a plot of text representations reduced to bi-dimensional vectors using PCA. The authors colored the data points according to the month or year of the posts' timestamps. Posts from 2014 occupy the center left of the image, while the representations of the other posts published in 2018, identified as July, August, September, and October, occupy the center and bottom of the image. In addition, the authors drew an arrow to show the direction of the concept drift.

In an ad-hoc manner, we mention some interesting works that approach concept drift / semantic shift visualization. Kazi et al. [99] proposed three visualization methods that emphasize the changes over time, starting with a reference word. The first proposed method is the radial bar chart, which can show top similar words, word re-occurrence, and degree of similarity. For example, considering Figure 1 in their work, the word *cigarette*, in the 1980s, was related to *tobacco*, while in the 2020s, it was related to *vape* and *ecigarettes*. Figure 2 in their work corresponds to a second proposal regarding the spiral line chart. This chart enables visualization of similar words, word re-occurrence, and continuity. Therefore, it eases understanding the appearance and fade of words related to a reference word. More specifically, Figure 2 in their paper considers the word *anxiety*[22]. To enhance the visualization of geographical information, the authors proposed a word cloud using maps of countries as silhouettes. Figure 3 in their paper shows this method. The authors analyzed the word *divorce*, hypothesizing that the use of this word in the 1970s/1980s regarded the divorce itself, while in recent years, *i.e.*, 2010s/2020s, it regarded the consequences of divorce, such as violence and self-harm. Periti et al. [161] also provided interesting highlights by using visualization. Although this work did not appear among the selected works, it uses WIDID [159], which was among the selected papers. Periti et al. [161] studied the semantic shifts of the Italian parliamentary speeches over time. The authors exemplified using the word *clean*. One visualization represents polysemy and the semantic shift of a word itself over time, *e.g.*, Figure 4a in their paper. On the other hand, Figure 4b in their paper emphasizes the prominence and sense shift of the sense nodules of a

[22]Available at: https://public.tableau.com/app/profile/raef6267/viz/SpiralLineChartConceptDrift/SpiralLineChart. Accessed on September 23rd, 2024.

given word over time. Although unrelated to streams and drifts, Huang et al. [91][23] provided an interesting visual survey for embedding visualization. We included this work in this discussion since a considerable number of selected papers leveraged embeddings as text representation methods and, therefore, Huang et al. [91] may inspire the development of new visualization methods towards drift visualization.

As aforementioned, concept drift in texts is common and can occur over time. However, depending on the characteristics of the approach and datasets, it may be challenging to execute the experiments due to the lack of certainty of the existence of drift, their potential positions, and their behavior over time. Therefore, some papers simulate drifts. For example, Li et al. [119], Murena et al. [141], Van Linh et al. [202] and Rabiu et al. [169] rearranged the topics sequentially in the stream. Thus, when a new topic emerges from the stream, it is considered a drift. Mohawesh et al. [137] simulated drift by dividing the datasets into partitions and rearranging them in different orders. For example, one of the datasets is initially ordered temporally and divided into five partitions, *i.e.*, $D1$, $D2$, ..., $D5$. Thus, in a specific scenario, the authors merged $D1 - D3$ for training and used the other partitions, *i.e.*, $D2$, $D4$, and $D5$, for testing sequentially. Although it created a scenario of concept drift and worked for the experiment in the aforementioned papers, both scenarios are unrealistic, especially considering the temporal aspects of the partitions in the latter example.

Across the analyzed papers, a number of authors mentioned the difficulty of finding datasets with labeled drifts ([168] and [69], to mention a few). To prove this aspect, considering the 48 papers analyzed, only two papers mentioned the existence of labeled drifts in their datasets: [65] (tweets regarding a soccer match) and [69] (for AGNews and 20NewsGroup), although those presented in [69] had their drifts (gradual and abrupt) artificially generated. This leads to the development of text drift generation methods to allow testing text classification methods and text drift detectors. Ghahramanian et al. [69] introduced drifts based on a procedure initially developed by Katakis et al. [97]. Garcia et al. [67] presented four text drift generation methods, *i.e.*, class swap, class shift, time slice removal, and adjective swap, based on Bravo-Marquez et al. [29], in which the former three involve manipulating classes, while the latter manipulates the sentence meaning by swapping adjectives with their antonyms.

Ultimately, depending on the sort of text drift, it can be challenging to visualize due to several factors, such as the inherent high dimensionality of the most frequent text representations. In addition, visually representing changes in text behavior over time can be challenging. Furthermore, developing scenarios to force concept drift in text streams can be complex, depending on the type of text drift. Generally, the datasets are described in the papers; however, sometimes, they lack evidence for the existence of text drift. Thus, it is necessary to resort to data rearrangement to simulate drifts and data visualization to search for changes in temporal patterns. However, to maintain consistency, it may be essential to consider the temporal order, especially concerning streaming scenarios.

7 CONCLUSION, OPEN CHALLENGES, AND FUTURE DIRECTIONS

In this study, we performed a systematic literature review on concept drift adaptation, specifically in text streams scenarios. A text stream is a specialization of data streams in which several texts arrive sequentially at high speeds. Sequentially handling texts is challenging due to the constraints of data stream settings, *i.e.*, processing time and memory consumption. In addition, we can mention characteristics of text-related settings, such as vocabulary maintenance, NLP, and text representation maintenance; ideally, these tasks should be performed on the fly.

[23]Available at: https://va-embeddings-browser.ivis.itn.liu.se/. Accessed on September 23rd, 2024.

We selected 48 papers and extracted information according to the defined criteria. We evaluated and categorized the papers regarding categories of drift, types of drift detection, the ML model update scheme, the stream mining tasks applied, the text representation method utilized, and the update scheme of the text representation methods. In this study, we also provided the metrics used in each stream mining task.

Text drift may happen due to several reasons. The natural evolution of writing can lead to drift, such as the emergence or disappearance of new words. In addition, texts generally reflect changes in the real world. Garcia et al. [65] mentioned that the drifts were generated by the goals scored by a team, leading to a positive sentiment. In Suprem et al. [192], Suprem and Pu [193, 194], the change in the volume of tweets regarding landslides could indicate the occurrence of the actual event. Li et al. [118] used cosine distance between clusters generated from different chunks to indicate the existence of topic drifts. A topic drift may happen due to changes in user interests over time. Heusinger et al. [84] mentioned the existence of drift in the dataset comprising tweets regarding different stocks from NASDAQ. These changes may happen due to the increase of posts because of actual news posts, any positive event such as an increase in the profit, announcement of dividends, or even negative events, such as scandals and corruption. Mohawesh et al. [137] worked on an adversarial problem, *i.e.*, fake reviews detection, in which a classifier model needs to be updated frequently to overcome new writing patterns from unlawful reviewers. Therefore, drifts, in this case, corresponded to those changes in writing patterns to bypass the classifier. In Bechini et al. [16], D'Andrea et al. [39], the drifts were the changes in stance distribution regarding specific topics, such as vaccination. de Mello et al. [41] considered drift the changes in the volume of tweets regarding news on Brazilian politics. In Pohl et al. [164], the drifts corresponded to changes in writing patterns to define whether the post was relevant to crisis management. In Garcia et al. [66], the drifts regarded the hashtag *#mybodymychoice* in different uses other than its original context. Rabinovich et al. [168] mentioned that drifts in their scenario regarded the failure of a newly deployed feature in systems. To summarize, many different reasons can cause text drifts, generally reflected by actual changes in the real world. Although it is a frequent phenomenon, text drifts are rarely labeled. It is a clear outcome of the difficulty of finding the exact point of many of those scenarios mentioned above. To confirm this statement, only two papers provided datasets with labeled drifts [65, 168]. However, in Rabinovich et al. [168], the drifts were introduced in the datasets to evaluate their method.

Regarding categories of drift, we differentiated the types into *real*, *virtual*, *feature drift*, and *semantic shift*. Most works (44) approached the real drift problem, corresponding to the mapping changes between X and y over time. Only four works considered the virtual drift, and another three tackled the semantic shift problem. Please note that a work can approach more than one drift category simultaneously. Considering the drift detection method, we investigated the papers and observed that it is possible to categorize them into *adaptive*, where the method adapts to the concept drift without detecting it, and *explicit*, where there is an explicit concept drift detection that can trigger the ML model update.

Furthermore, we investigated the strategies employed by the methods and systems to update ML models as needed. We categorized the studied papers considering the ML update scheme into four groups: (i) ensemble update, (ii) incremental, (iii) keep-compare-evolve, and (iv) retraining. In addition, we analyzed the applications approached in the papers according to a stream mining task categorization. The stream mining tasks found in the studies were categorized into classification, clustering, general detection, and topic modeling. Several applications were found, such as fake review detection, sentiment analysis, and novelty detection.

In addition, we organized and presented the text representation methods since they are crucial for text streams subject to concept drift. Sixteen text representation methods were identified, where Bag-of-words and Word2Vec were the most frequent methods (each appeared in 11 studies). Moreover, when available, the update mechanisms of the text representations were also listed. Only two methods are fully incremental, while most studies used static text representation methods/language models. Therefore, it constitutes an open challenge.

Additionally, we listed the real-world datasets with their links when available and discussed concept drifts visualization and drifts simulation. Some papers argued that the datasets in use have drift, although such drifts are unlabeled or uncategorized. A few papers resorted to visualization techniques or data rearrangement to simulate drift to justify the existence of drifts. Concept drifts in text streams can manifest in various ways, including feature drift, semantic shift, real and virtual drifts, and topic drift. Thus, different approaches are required to manage these types of drifts.

It is worth mentioning the extraordinary advances that have been made regarding LLMs recently. There is some discussion about the requirements for a language model to be considered large. For example, BERT is considered an LLM [111], although a pre-trained BERT large uncased has 340 million parameters. Considering BERT-like families, some papers addressed the temporal adaptation in these language models. For example, Hu et al. [89] developed a framework to address temporal shifts in news posts. Su et al. [190] also directed their efforts to address semantic changes using language models. Agarwal and Nenkova [4], on the other hand, evaluated the temporal effects on pre-trained language models. Amba Hombaiah et al. [7] also addressed concept drift but with a focus on text stream scenarios.

More recently, other works mentioned that LLMs are generally constituted of billions of parameters capable of performing tasks based on prompts, sometimes in a zero-shot fashion [104, 172]. However, combining LLMs such as Llama and GPT-3 (or more recent versions) in text stream scenarios subject to concept drift is still open.

7.1 Open Challenges and Future Directions

During this study, we discovered aspects that can be addressed in future research and remain as open challenges.

7.1.1 ***Text drift visualization***. The research area still requires visualization methods that highlight the existence of text drift. There is no standard for generating those visualizations, especially regarding changes over time. Due to the variety of tasks and applications, the existence of different visualizations with no standard is understandable. However, developing visualization methods that are easy to interpret and generate may help justify the presence of drifts.

7.1.2 ***Benchmark for text streams datasets subject to concept drift***. As verified in this paper, there is no benchmark to compare the ability of learning methods in text stream scenarios subject to concept drift. In particular, only two papers provided datasets with labeled drifts. Therefore, different approaches for text drift simulation have been used in the literature. Standardization in these processes may be an advantage, enabling faster development of the research area. In addition, as it could be seen early in this section, the source of text drifts can be domain-specific, demanding further analysis for a deep understanding of the phenomenon.

The authors in the selected papers collected many datasets; however, the most frequent datasets across the papers were related to short-text scenarios or topics. Thus, it is crucial to develop benchmark datasets for text drift detection focused on text stream scenarios in the future.

7.1.3 ***Incremental methods for semantic shift detection***. Considering the semantic shift, it can be advanced in linguistics and be studied in depth. According to the information obtained from the papers that approach semantic shift detection studied in this work, a challenging aspect is the need to monitor all the words in the vocabulary. Thus, it appears that methods that can indicate words that suffer semantic shift in text streams are desired to reduce computational load. Besides, a reduced number of papers approached semantic shift in text stream scenarios, *e.g.*, [159]. Given that text streams have their constraints and incremental approaches are more suitable to these scenarios, producing incremental methods for semantic shift detection may help develop the area.

7.1.4 ***Incremental text representation methods***. As verified in this study, a few text representation methods were able to embed updates over time. For example, frequency-based approaches, such as Bag-of-Words and TF-IDF, may suffer from the appearance and disappearance of words over time, considering the case of defining the reference tokens at the beginning of the text stream processing. Being able to incorporate updates over time to representations provided by pre-trained language models or effectively modeling dense representations over time without creating a bottleneck in the process may be a future direction regarding this regard.

7.1.5 ***Text drift detection in LLM environments***. Although studying LLMs such as Llama and GPT family are outside the scope of this paper, text drift detection may be important in scenarios that leverage LLMs. For example, the tasks of preventing jailbreaks and prompt injection can be modeled as text stream scenarios, in which the input is the user interactions with the LLM, and the jailbreaks and prompt injection could be analyzed as drifts in the user input stream. Prompt injection is a type of attack that introduces instructions to manipulate the LLM to perform the attacker's intention [122]. At the same time, jailbreak, in this sense, means the input of malicious instructions to provoke undesired LLM's behavior [45]. Although there are pre-trained models for this task, such as the Prompt-Guard-86M[24], a problem in this regard is the nature of the scenario, which is clearly adversarial, meaning that incremental learning/adaptation is frequently desired.

In summary, this systematic review provides a detailed analysis and evaluation of concept drift adaptation methods in text stream scenarios, offering valuable insights that may help readers understand the strengths and weaknesses of the current techniques and open issues that need to be addressed.

ACKNOWLEDGEMENTS

We are grateful to the anonymous reviewers for their significant comments and suggestions to improve the manuscript.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.

[2] Amal Abid, Salma Jamoussi, and Abdelmajid Ben Hamadou. 2018. Handling Concept Drift and Feature Evolution in Textual Data Stream using the Artificial Immune System. In *Proceedings of the 10th International Conference on Computational Collective Intelligence (ICCCI 2018) (Lecture Notes in Computer Science, Vol. 11055)*. Springer, 363–372. https://doi.org/10.1007/978-3-319-98443-8_33

[3] Amal Abid, Salma Jamoussi, and Abdelmajid Ben Hamadou. 2019. AIS-Clus: A Bio-Inspired Method for Textual Data Stream Clustering. *Vietnam Journal of Computer Science* 6, 2 (2019), 223–256. https://doi.org/10.1142/S2196888819500143

[24] https://huggingface.co/meta-llama/Prompt-Guard-86M

[4] Oshin Agarwal and Ani Nenkova. 2022. Temporal Effects on Pre-trained Models for Language Processing Tasks. *Transactions of the Association for Computational Linguistics* 10 (2022), 904–921. https://doi.org/10.1162/TACL_A_00497
[5] Ravinder Ahuja, Aakarsha Chug, Shruti Kohli, Shaurya Gupta, and Pratyush Ahuja. 2019. The Impact of Features Extraction on the Sentiment Analysis. *Procedia Computer Science* 152 (2019), 341–348. https://doi.org/10.1016/j.procs.2019.05.008
[6] Reem Alatrash, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte Im Walde. 2020. CCOHA: Clean Corpus of Historical American English. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)*. European Language Resources Association, 6958–6966.
[7] Spurthi Amba Hombaiah, Tao Chen, Mingyang Zhang, Michael Bendersky, and Marc Najork. 2021. Dynamic Language Models for Continuously Evolving Content. In *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2021)*. ACM. https://doi.org/10.1145/3447548.3467162
[8] Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? Deep learning to the rescue!. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. AAAI Press, 7383–7390. https://doi.org/10.1609/aaai.v34i05.6233
[9] Charles E Antoniak. 1974. Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics* 2, 6 (1974), 1152–1174.
[10] Dennis Assenmacher and Heike Trautmann. 2022. Textual One-Pass Stream Clustering with Automated Distance Threshold Adaption. In *Proceedings of the 14th Asian Conference on Intelligent Information and Database Systems (ACIIDS 2022) (Lecture Notes in Computer Science, Vol. 13757)*. Springer, 3–16. https://doi.org/10.1007/978-3-031-21743-2_1
[11] Josh Attenberg, Kilian Weinberger, Anirban Dasgupta, Alex Smola, and Martin Zinkevich. 2009. Collaborative email-spam filtering with the hashing trick. In *Proceedings of the 6th Conference on Email and Anti-Spam (CEAS 2009)*. 1–4.
[12] Manuel Baena-Garcıa, José del Campo-Ávila, Raul Fidalgo, Albert Bifet, Ricard Gavalda, and Rafael Morales-Bueno. 2006. Early Drift Detection Method. In *Proceedings of the Fourth International Workshop on Knowledge Discovery from Data Stream*, Vol. 6. 77–86.
[13] Rodrigo Barbado, Oscar Araque, and Carlos A Iglesias. 2019. A Framework for Fake Review Detection in Online Consumer Electronics Retailers. *Information Processing & Management* 56, 4 (2019), 1234–1244. https://doi.org/10.1016/J.IPM.2019.03.002
[14] Jean Paul Barddal, Heitor Murilo Gomes, Fabrício Enembreck, and Bernhard Pfahringer. 2017. A Survey on Feature Drift Adaptation: Definition, Benchmark, Challenges and Future Directions. *Journal of Systems and Software* 127 (2017), 278–294. https://doi.org/10.1016/J.JSS.2016.07.005
[15] Jean Paul Barddal, Lucas Loezer, Fabrício Enembreck, and Riccardo Lanzuolo. 2020. Lessons Learned from Data Stream Classification applied to Credit Scoring. *Expert Systems with Applications* 162 (2020), 113899. https://doi.org/10.1016/J.ESWA.2020.113899
[16] Alessio Bechini, Alessandro Bondielli, Pietro Ducange, Francesco Marcelloni, and Alessandro Renda. 2021. Addressing Event-driven Concept Drift in Twitter Stream: a Stance Detection Application. *IEEE Access* 9 (2021), 77758–77770. https://doi.org/10.1109/ACCESS.2021.3083578
[17] Federico Belotti, Federico Bianchi, and Matteo Palmonari. 2020. UNIMIB@ DIACR-Ita: Aligning Distributional Embeddings with a Compass for Semantic Change Detection in the Italian Language. In *Proceedings of the 7th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2020)*. 451–455. https://doi.org/10.4000/books.aaccademia.7688
[18] Eduardo Bezerra, Emanuel Passos, and Ronaldo Goldschmidt. 2015. *Data Mining: Conceitos, técnicas, algoritmos, orientações e aplicações.* Campus, Rio de Janeiro, RJ, Brazil.
[19] Albert Bifet. 2017. Classifier Concept Drift Detection and the Illusion of Progress. In *Proceedings of the 16th International Conference on Artificial Intelligence and Soft Computing (ICAISC 2017) (Lecture Notes in Computer Science, Vol. 10246)*. Springer, 715–725. https://doi.org/10.1007/978-3-319-59060-8_64
[20] Albert Bifet and Ricard Gavalda. 2007. Learning from Time-changing Data with Adaptive Windowing. In *Proceedings of the 2007 SIAM International Conference on Data Mining*. SIAM, 443–448. https://doi.org/10.1137/1.9781611972771.42
[21] Albert Bifet, Ricard Gavalda, Geoff Holmes, and Bernhard Pfahringer. 2018. *Machine Learning for Data Streams: with Practical Examples in MOA*. MIT Press, Cambridge, USA. https://doi.org/10.7551/mitpress/10654.001.0001
[22] Albert Bifet, Geoff Holmes, Bernhard Pfahringer, Philipp Kranen, Hardy Kremer, Timm Jansen, and Thomas Seidl. 2010. MOA: Massive Online Analysis, a Framework for Stream Classification and Clustering. In *Proceedings of the 1st Workshop on Applications of Pattern Analysis (JMLR Proceedings, Vol. 11)*. JMLR.org, 44–50.
[23] David M Blei. 2012. Probabilistic Topic Models. *Commun. ACM* 55, 4 (2012), 77–84. https://doi.org/10.1145/2133806.2133826

[24] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3, Jan (2003), 993–1022.
[25] Leonard Bloomfield. 1933. *Language*. George Allen & Unwin.
[26] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistic* 5 (2017), 135–146. https://doi.org/10.1162/TACL_A_00051
[27] Alessandro Bondielli, Giuseppe Cancello Tortora, Pietro Ducange, Armando Macri, Francesco Marcelloni, and Alessandro Renda. 2022. Online Monitoring of Stance from Tweets: The case of Green Pass in Italy. In *Proceedings of the IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS 2022)*. IEEE, 1–8. https://doi.org/10.1109/EAIS51927.2022.9787753
[28] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time Series Analysis: Forecasting and Control* (5 ed.). John Wiley & Sons, Hoboken, NJ, USA.
[29] Felipe Bravo-Marquez, Arun Khanchandani, and Bernhard Pfahringer. 2022. Incremental Word Vectors for Time-Evolving Sentiment Lexicon Induction. *Cognitive Computation* 14, 1 (2022), 425–441. https://doi.org/10.1007/S12559-021-09831-Y
[30] Kellyton Brito and Paulo Jorge Leitão Adeodato. 2023. Machine Learning for Predicting Elections in Latin America based on Social Media Engagement and Polls. *Government Information Quarterly* 40, 1 (2023), 101782. https://doi.org/10.1016/J.GIQ.2022.101782
[31] Silvana Castano, Alfio Ferrara, Stefano Montanelli, and Francesco Periti. 2024. Incremental Affinity Propagation based on Cluster Consolidation and Stratification. *arXiv preprint arXiv:2401.14439* (2024).
[32] Manuel Cebrián, Manuel Alfonseca, and Alfonso Ortega. 2007. The Normalized Compression Distance is Resistant to Noise. *IEEE Transactions on Information Theory* 53, 5 (2007), 1895–1900. https://doi.org/10.1109/TIT.2007.894669
[33] Jorge Cristhian Chamby-Diaz, Mariana Recamonde Mendoza, and Ana Lúcia C. Bazzan. 2019. Dynamic Correlation-Based Feature Selection for Feature Drifts in Data Streams. In *Proceedings of the 8th Brazilian Conference on Intelligent Systems (BRACIS 2019)*. IEEE, 198–203. https://doi.org/10.1109/BRACIS.2019.00043
[34] François Chollet et al. 2015. *Keras*. Retrieved October 20, 2024 from https://keras.io
[35] Rudi Cilibrasi and Paul MB Vitányi. 2005. Clustering by Compression. *IEEE Transactions on Information Theory* 51, 4 (2005), 1523–1545. https://doi.org/10.1109/TIT.2005.844059
[36] Joana Costa, Catarina Silva, Mário Antunes, and Bernardete Ribeiro. 2017. Adaptive Learning for Dynamic Environments: A Comparative Approach. *Engineering Applications of Artificial Intelligence* 65 (2017), 336–345. https://doi.org/10.1016/J.ENGAPPAI.2017.08.004
[37] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research* 7 (2006), 551–585.
[38] Prithiviraj Damodaran. 2021. *Parrot: Paraphrase generation for NLU - v1.0*. Retrieved October 20, 2024 from https://github.com/PrithivirajDamodaran/Parrot_Paraphraser
[39] Eleonora D'Andrea, Pietro Ducange, Alessio Bechini, Alessandro Renda, and Francesco Marcelloni. 2019. Monitoring the Public Opinion about the Vaccination Topic from Tweets Analysis. *Expert Systems with Applications* 116 (2019), 209–226. https://doi.org/10.1016/J.ESWA.2018.09.009
[40] Roberto Souto Maior de Barros, Danilo Rafael de Lima Cabral, Paulo Mauricio Gonçalves Jr., and Silas Garrido Teixeira de Carvalho Santos. 2017. RDDM: Reactive drift detection method. *Expert Systems with Applications* 90 (2017), 344–355. https://doi.org/10.1016/J.ESWA.2017.08.023
[41] Rodrigo F. de Mello, Ricardo A. Rios, Paulo A. Pagliosa, and Caio S. Lopes. 2018. Concept drift detection on social network data using cross-recurrence quantification analysis. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28, 8 (2018), 085719. https://doi.org/10.1063/1.5024241
[42] Matheus Bernardelli de Moraes and Andre Leon Sampaio Gradvohl. 2021. A Comparative Study of Feature Selection Methods for Binary Text Streams. *Evolving Systems* 12, 4 (2021), 997–1013. https://doi.org/10.1007/S12530-020-09357-Y
[43] Jader Martins Camboim de Sá, Marcos Da Silveira, and Cédric Pruski. 2024. Survey in Characterization of Semantic Change. *arXiv preprint arXiv:2402.19088* (2024).
[44] Bruna Rossetto Delazeri, Leonardo León Vera, Jean Paul Barddal, Alessandro L. Koerich, and Alceu S. Britto Jr. 2022. Evaluation of Self-taught Learning-based Representations for Facial Emotion Recognition. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2022)*. IEEE, 1–8. https://doi.org/10.1109/IJCNN55064.2022.9891956
[45] Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. 2024. Multilingual Jailbreak Challenges in Large Language Models. In *Proceedings o the 12th International Conference on Learning Representations (ICLR 2024)*. 1–18.
[46] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*. Association for

Computational Linguistics, 4171–4186. https://doi.org/10.18653/V1/N19-1423
[47] Valerio Di Carlo, Federico Bianchi, and Matteo Palmonari. 2019. Training Temporal Word Embeddings with a Compass. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*, Vol. 33. AAAI, 6326–6334. https://doi.org/10.1609/AAAI.V33I01.33016326
[48] Pietro Ducange, Michela Fazzolari, Francesco Marcelloni, Martina Marino, and Roberta Matrella. 2024. Continuous Monitoring of Body Shaming Actions in Social Networks. In *Proceedings of the IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS 2024)*. IEEE, 1–8. https://doi.org/10.1109/EAIS58494.2024.10570011
[49] Nugroho Dwi Prasetyo and Claudia Hauff. 2015. Twitter-based Election Prediction in the Developing World. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media (HT 2015)*. ACM, 149–158. https://doi.org/10.1145/2700171.2791033
[50] Ryan Elwell and Robi Polikar. 2011. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks* 22, 10 (2011), 1517–1531. https://doi.org/10.1109/TNN.2011.2160459
[51] Scott Emmons, Stephen Kobourov, Mike Gallant, and Katy Börner. 2016. Analysis of Network Clustering Algorithms and Cluster Quality Metrics at Scale. *PloS ONE* 11, 7 (2016), 1–18. https://doi.org/10.1371/journal.pone.0159161
[52] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*. AAAI, 226–231.
[53] Elaine R Faria, Isabel JCR Gonçalves, André CPLF de Carvalho, and João Gama. 2016. Novelty Detection in Data Streams. *Artificial Intelligence Review* 45, 2 (2016), 235–269. https://doi.org/10.1007/S10462-015-9444-8
[54] Xingdong Feng, Xin He, Caixing Wang, Chao Wang, and Jingnan Zhang. 2024. Towards a unified analysis of kernel-based methods under covariate shift. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*, Vol. 36. 73839–73851.
[55] Giuseppe Fenza, Mariacristina Gallo, Vincenzo Loia, Alessandra Petrone, and Claudio Stanzione. 2023. Concept-drift Detection Index based on Fuzzy Formal Concept Analysis for Fake News Classifiers. *Technological Forecasting and Social Change* 194 (2023), 122640. https://doi.org/10.1016/j.techfore.2023.122640
[56] Isvani Frias-Blanco, José del Campo-Ávila, Gonzalo Ramos-Jimenez, Rafael Morales-Bueno, Agustin Ortiz-Diaz, and Yailé Caballero-Mota. 2014. Online and non-parametric drift detection methods based on Hoeffding's bounds. *IEEE Transactions on Knowledge and Data Engineering* 27, 3 (2014), 810–823. https://doi.org/10.1109/TKDE.2014.2345382
[57] Milton Friedman. 1937. The use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Amer. Statist. Assoc.* 32, 200 (1937), 675–701. https://doi.org/10.1080/01621459.1937.10503522
[58] Milton Friedman. 1940. A Comparison of Alternative Tests of Significance for the Problem of m Rankings. *The Annals of Mathematical Statistics* 11, 1 (1940), 86–92. https://doi.org/10.1214/aoms/1177731944
[59] Chin-Lun Fu, Zih-Ching Chen, Yun-Ru Lee, and Hung-yi Lee. 2022. AdapterBias: Parameter-efficient Token-dependent Representation Shift for Adapters in NLP Tasks. In *Findings of the Association for Computational Linguistics: NAACL 2022*. Association for Computational Linguistics, 2608–2621. https://doi.org/10.18653/v1/2022.findings-naacl.199
[60] João Gama, Ricardo Fernandes, and Ricardo Rocha. 2006. Decision Trees for Mining Data Streams. *Intelligent Data Analysis* 10, 1 (2006), 23–45. https://doi.org/10.3233/IDA-2006-10103
[61] Joao Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. 2004. Learning with Drift Detection. In *Advances in Artificial Intelligence: Proceedings of the 17th Brazilian Symposium on Artificial Intelligence (SBIA 2004)*. Springer, 286–295. https://doi.org/10.1007/978-3-540-28645-5_29
[62] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A Survey on Concept Drift Adaptation. *ACM Computing Surveys (CSUR)* 46, 4 (2014), 1–37. https://doi.org/10.1145/2523813
[63] Cristiano Garcia, Ahmed Esmin, Daniel Leite, and Igor Škrjanc. 2019. Evolvable Fuzzy Systems from Data Streams with Missing Values: With Application to Temporal Pattern Recognition and Cryptocurrency Prediction. *Pattern Recognition Letters* 128 (2019), 278–282. https://doi.org/10.1016/J.PATREC.2019.09.012
[64] Cristiano Garcia, Daniel Leite, and Igor Škrjanc. 2019. Incremental Missing-data Imputation for Evolving Fuzzy Granular Prediction. *IEEE Transactions on Fuzzy Systems* 28, 10 (2019), 2348–2362. https://doi.org/10.1109/TFUZZ.2019.2935688
[65] Cristiano Mesquita Garcia, Alceu de Souza Britto, and Jean Paul Barddal. 2023. Event-driven Sentiment Drift Analysis in Text Streams: An Application in a Soccer Match. In *Proceedings of the International Conference on Machine Learning and Applications (ICMLA 2023)*. IEEE, 1920–1927. https://doi.org/10.1109/ICMLA58977.2023.00291
[66] Cristiano Mesquita Garcia, Alceu de Souza Britto Jr, and Jean Paul Barddal. 2024. Temporal analysis of drifting hashtags in textual data streams: A graph-based application. *Expert Systems with Applications* 257 (2024), 125007. https://doi.org/10.1016/J.ESWA.2024.125007
[67] Cristiano Mesquita Garcia, Alessandro Lameiras Koerich, Alceu de Souza Britto Jr, and Jean Paul Barddal. 2024. Methods for Generating Drift in Text Streams. *arXiv preprint arXiv:2403.12328* (2024).

[68] Salvador Garcia and Francisco Herrera. 2008. An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons. *Journal of Machine Learning Research* 9, 12 (2008), 2677–2694.

[69] Pouya Ghahramanian, Sepehr Bakhshi, Hamed Bonab, and Fazli Can. 2024. A Novel Neural Ensemble Architecture for On-the-fly Classification of Evolving Text Streams. *ACM Transactions on Knowledge Discovery from Data* 18, 4 (2024), 1–24. https://doi.org/10.1145/3639054

[70] Mario Giulianelli, Iris Luden, Raquel Fernández, and Andrey Kutuzov. 2023. Interpretable Word Sense Representations via Definition Generation: The Case of Semantic Change Analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*. Association for Computational Linguistics, 3130–3148. https://doi.org/10.18653/V1/2023.ACL-LONG.176

[71] Heitor M Gomes, Albert Bifet, Jesse Read, Jean Paul Barddal, Fabrício Enembreck, Bernhard Pfharinger, Geoff Holmes, and Talel Abdessalem. 2017. Adaptive Random Forests for Evolving Data Stream Classification. *Machine Learning* 106 (2017), 1469–1495. https://doi.org/10.1007/S10994-017-5642-8

[72] Maurício Gruppi, Benjamin D. Horne, and Sibel Adali. 2020. NELA-GT-2019: A Large Multi-Labelled News Dataset for The Study of Misinformation in News Articles. *arXiv preprint arXiv:2003.08444* (2020).

[73] Maurício Gruppi, Benjamin D Horne, and Sibel Adalı. 2021. NELA-GT-2020: A large multi-labelled news dataset for the study of misinformation in news articles. *arXiv preprint arXiv:2102.04567* (2021).

[74] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA Data Mining Software: an Update. *ACM SIGKDD Explorations Newsletter* 11, 1 (2009), 10–18. https://doi.org/10.1145/1656274.1656278

[75] Mark A Hall. 1999. *Correlation-based Feature Selection for Machine Learning*. Ph. D. Dissertation. The University of Waikato.

[76] William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Cultural Shift or Linguistic Drift? Comparing Two Computational Measures of Semantic Change. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*. Association for Computational Linguistics, 2116–2121. https://doi.org/10.18653/V1/D16-1229

[77] William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*. Association for Computational Linguistics, 1489–1501. https://doi.org/10.18653/V1/P16-1141

[78] Hugo Lewi Hammer and Anis Yazidi. 2018. Parameter Estimation in Abruptly Changing Dynamic Environments using Stochastic Learning Weak Estimator. *Applied Intelligence* 48, 11 (2018), 4096–4112. https://doi.org/10.1007/S10489-018-1205-3

[79] Hannes Hapke, Cole Howard, and Hobson Lane. 2019. *Natural Language Processing in Action: Understanding, analyzing, and generating text with Python*. Manning Publications, Shelter Island, NY, USA.

[80] Zellig S Harris. 1954. Distributional Structure. *Word* 10, 2-3 (1954), 146–162. https://doi.org/10.1080/00437956.1954.11659520

[81] Yu He, Jianxin Li, Yangqiu Song, Mutian He, and Hao Peng. 2018. Time-evolving Text Classification with Deep Neural Networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*. International Joint Conferences on Artificial Intelligence Organization, 2241–2247. https://doi.org/10.24963/IJCAI.2018/310

[82] Ernst Hellinger. 1909. Neue Begründung der Theorie Quadratischer Formen von Unendlichvielen Veränderlichen. *Journal für die Reine und Angewandte Mathematik* 136 (1909), 210–271. https://doi.org/10.1515/crll.1909.136.210

[83] Moritz Heusinger, Christoph Raab, and Frank-Michael Schleif. 2019. Passive Concept Drift Handling via Momentum based Robust Soft Learning Vector Quantization. In *Advances in Self-Organizing Maps, Learning Vector Quantization, Clustering and Data Visualization: Proceedings of the 13th International Workshop (WSOM+ 2019) (Advances in Intelligent Systems and Computing, Vol. 976)*. Springer, 200–209. https://doi.org/10.1007/978-3-030-19642-4_20

[84] Moritz Heusinger, Christoph Raab, and Frank-Michael Schleif. 2020. Analyzing Dynamic Social Media Data via Random Projection - A New Challenge for Stream Classifiers. In *Proceedings of the IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS 2020)*. IEEE, 1–8. https://doi.org/10.1109/EAIS48028.2020.9122780

[85] Moritz Heusinger, Christoph Raab, and Frank-Michael Schleif. 2022. Dimensionality Reduction in the Context of Dynamic Social Media Data Streams. *Evolving Systems* 13, 3 (2022), 387–401. https://doi.org/10.1007/S12530-021-09396-Z

[86] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. 2013. Stochastic Variational Inference. *Journal of Machine Learning Research* 14, 1 (2013), 1303–1347. https://doi.org/10.5555/2567709.2502622

[87] Valentin Hofmann, Janet B Pierrehumbert, and Hinrich Schütze. 2021. Dynamic contextualized word embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL/IJCNLP 2021)*. Association for Computational Linguistics, 6970–6984. https://doi.org/10.18653/V1/2021.ACL-LONG.542

[88] Charles C Holt. 2004. Forecasting Seasonals and Trends by Exponentially Weighted Moving Averages. *International Journal of Forecasting* 20, 1 (2004), 5–10. https://doi.org/10.1016/j.ijforecast.2003.09.015

[89] Beizhe Hu, Qiang Sheng, Juan Cao, Yongchun Zhu, Danding Wang, Zhengjia Wang, and Zhiwei Jin. 2023. Learn over past, evolve for future: Forecasting temporal trends for fake news detection. In *Proceedings of the The 61st Annual Meeting of the Association for Computational Linguistics: Industry Track (ACL 2023)*. Association for Computational Linguistics, 116–125. https://doi.org/10.18653/V1/2023.ACL-INDUSTRY.13

[90] Xuegang Hu, Haiyan Wang, and Peipei Li. 2018. Online Biterm Topic Model Based Short Text Stream Classification using Short Text Expansion and Concept Drifting Detection. *Pattern Recognition Letters* 116 (2018), 187–194. https://doi.org/10.1016/J.PATREC.2018.10.018

[91] Zeyang Huang, Daniel Witschard, Kostiantyn Kucher, and Andreas Kerren. 2023. VA+ Embeddings STAR: A State-of-the-Art Report on the Use of Embeddings in Visual Analytics. *Computer Graphics Forum* 42, 3 (2023), 539–571. https://doi.org/10.1111/cgf.14859

[92] Ronald L Iman and James M Davenport. 1980. Approximations of the Critical Region of the Friedman Statistic. *Communications in Statistics-Theory and Methods* 9, 6 (1980), 571–595. https://doi.org/10.1080/03610928008827904

[93] Shotaro Ishihara, Hiromu Takahashi, and Hono Shirai. 2022. Semantic shift stability: Efficient way to detect performance degradation of word embeddings and pre-trained language models. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (AACL/IJCNLP 2022)*. Association for Computational Linguistics, 205–216.

[94] Gabriel Iturra-Bocaz and Felipe Bravo-Marquez. 2023. RiverText: A Python Library for Training and Evaluating Incremental Word Embeddings from Text Data Streams. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2023)*. ACM, 3027–3036. https://doi.org/10.1145/3539618.3591908

[95] Nobuhiro Kaji and Hayato Kobayashi. 2017. Incremental Skip-gram Model with Negative Sampling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*. Association for Computational Linguistics, 363–371. https://doi.org/10.18653/V1/D17-1037

[96] Ioannis Katakis, Grigorios Tsoumakas, Evangelos Banos, Nick Bassiliades, and Ioannis Vlahavas. 2009. An Adaptive Personalized News Dissemination System. *Journal of Intelligent Information Systems* 32, 2 (2009), 191–212. https://doi.org/10.1007/S10844-008-0053-8

[97] Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2008. An ensemble of classifiers for coping with recurring contexts in data streams. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008)*. Frontiers in Artificial Intelligence and Applications, Vol. 178. IOS Press, 763–764. https://doi.org/10.3233/978-1-58603-891-5-763

[98] Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2010. Tracking Recurring Contexts using Ensemble Classifiers: an Application to Email Filtering. *Knowledge and Information Systems* 22, 3 (2010), 371–391. https://doi.org/10.1007/S10115-009-0206-2

[99] Raef Kazi, Alessandra Amato, Shenghui Wang, and Doina Bucur. 2022. Visualization Methods for Diachronic Semantic Shift. In *Proceedings of the 3rd Workshop on Scholarly Document Processing (SDP 2022) at 29th International Conference on Computational Linguistics (COLING 2022)*. Association for Computational Linguistics, 89–94.

[100] Jeffrey O. Kephart. 1994. A Biologically Inspired Immune System for Computers. In *Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems*. MIT Press, 130–139.

[101] Pooja Kherwa and Poonam Bansal. 2019. Topic Modeling: A Comprehensive Review. *EAI Endorsed Transactions on Scalable Information Systems* 7, 24 (2019). https://doi.org/10.4108/EAI.13-7-2018.159623

[102] Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. 2014. Temporal analysis of language through neural language models. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science (ACL 2014)*. Association for Computational Linguistics, 61–65. https://doi.org/10.3115/V1/W14-2517

[103] Barbara Kitchenham and Stuart Charters. 2007. *Guidelines for Performing Systematic Literature Reviews in Software Engineering - version 2.3*. Technical Report EBSE-2007-01. Keele University and Durham University Joint Report, Staffordshire and Durham, UK.

[104] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, Vol. 35. 22199–22213.

[105] Taiwo Kolajo, Olawande Daramola, and Ayodele A Adebiyi. 2022. Real-time Event Detection in Social Media Streams Through Semantic Analysis of Noisy Terms. *Journal of Big Data* 9, 1 (2022), 1–36. https://doi.org/10.1186/S40537-022-00642-Y

[106] Andrej N Kolmogorov. 1933. Sulla Determinazione Empirica di una Legge Didistribuzione. *Giornale dell'Istituto Italiano degli Attuari* 4 (1933), 83–91.

[107] Jeremy Z Kolter and Marcus A Maloof. 2005. Using Additive Expert Ensembles to Cope with Concept Drift. In *Proceedings of the 22nd International Conference on Machine learning (ICML 2005) (ACM International Conference*

Proceeding Series, Vol. 119). ACM, 449–456. https://doi.org/10.1145/1102351.1102408

[108] Dilek Küçük and Fazli Can. 2020. Stance Detection: A Survey. *ACM Computing Surveys (CSUR)* 53, 1, Article 12 (2020), 37 pages. https://doi.org/10.1145/3369026

[109] Solomon Kullback and Richard A Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics* 22, 1 (1951), 79–86.

[110] Jay Kumar, Junming Shao, Rajesh Kumar, Salah Ud Din, Cobbinah B Mawuli, and Qinli Yang. 2023. Online Semi-Supervised Classification on Multilabel Evolving High-Dimensional Text Streams. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 53, 10 (2023), 5983–5995. https://doi.org/10.1109/TSMC.2023.3275298

[111] Eldar Kurtic, Daniel Campos, Tuan Nguyen, Elias Frantar, Mark Kurtz, Benjamin Fineran, Michael Goin, and Dan Alistarh. 2022. The optimal BERT surgeon: Scalable and accurate second-order pruning for large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)*. Association for Computational Linguistics, 4163–4181. https://doi.org/10.18653/V1/2022.EMNLP-MAIN.279

[112] Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*. Association for Computational Linguistics, 1384–1397.

[113] Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*. Association for Computational Linguistics, 1311–1316. https://doi.org/10.18653/V1/D19-1131

[114] Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*. Association for Computer Linguistics, 530–539. https://doi.org/10.3115/V1/E14-1056

[115] Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31th International Conference on Machine Learning (ICML 2014) (JMLR Workshop and Conference Proceedings, Vol. 32)*. JMLR.org, 1188–1196.

[116] Daniel Leite, Rosangela Ballini, Pyramo Costa, and Fernando Gomide. 2012. Evolving Fuzzy Granular Modeling from Nonstationary Fuzzy Data Streams. *Evolving Systems* 3, 2 (2012), 65–79. https://doi.org/10.1007/S12530-012-9050-9

[117] David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research* 5 (2004), 361–397.

[118] Peipei Li, Lu He, Haiyan Wang, Xuegang Hu, Yuhong Zhang, Lei Li, and Xindong Wu. 2018. Learning from Short Text Streams with Topic Drifts. *IEEE Transactions on Cybernetics* 48, 9 (2018), 2697–2711. https://doi.org/10.1109/TCYB.2017.2748598

[119] Peipei Li, Yingying Liu, Yang Hu, Yuhong Zhang, Xuegang Hu, and Kui Yu. 2022. A Drift-sensitive Distributed LSTM Method for Short Text Stream Classification. *IEEE Transactions on Big Data* 9, 1 (2022), 341–357. https://doi.org/10.1109/TBDATA.2022.3164239

[120] Shaohua Li, Jun Zhu, and Chunyan Miao. 2017. PSDVec: A toolbox for incremental and scalable word embedding. *Neurocomputing* 237 (2017), 405–409. https://doi.org/10.1016/J.NEUCOM.2016.05.093

[121] Jinghua Liu, Yaojin Lin, Yuwen Li, Wei Weng, and Shunxiang Wu. 2018. Online multi-label streaming feature selection based on neighborhood rough set. *Pattern Recognition* 84 (2018), 273–287. https://doi.org/10.1016/J.PATCOG.2018.07.021

[122] Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. 2023. Prompt injection attacks and defenses in llm-integrated applications. *arXiv preprint arXiv:2310.12815* (2023).

[123] Yang Liu, Alan Medlar, and Dorota Glowacka. 2021. Statistically significant detection of semantic shifts using contextual word embeddings. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems (Eval4NLP 2021)*. Association for Computational Linguistics, 104–113.

[124] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692* (2019).

[125] Stuart Lloyd. 1982. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory* 28, 2 (1982), 129–137. https://doi.org/10.1109/TIT.1982.1056489

[126] Viktor Losing, Barbara Hammer, and Heiko Wersing. 2017. Self-Adjusting Memory: How to deal with Diverse Drift Types. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017)*. International Joint Conferences on Artificial Intelligence Organization, 4899–4903. https://doi.org/10.24963/IJCAI.2017/690

[127] Yuyin Lu, Xin Cheng, Ziran Liang, and Yanghui Rao. 2022. Graph-based Dynamic Word Embeddings. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI 2022)*. International Joint Conferences on

Artificial Intelligence Organization, 4280–4288. https://doi.org/10.24963/IJCAI.2022/594
[128] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL 2014)*. Association for Computer Linguistics, 55–60. https://doi.org/10.3115/V1/P14-5010
[129] Chandler May, Kevin Duh, Benjamin Van Durme, and Ashwin Lall. 2017. Streaming Word Embeddings with the Space-saving Algorithm. *arXiv preprint arXiv:1704.07463* (2017).
[130] Barbara McGillivray and Adam Kilgarriff. 2013. Tools for Historical Corpus Research, and a Corpus of Latin. In *New Methods in Historical Corpus Linguistics*. Narr, 1–10.
[131] Mary L McHugh. 2012. Interrater Reliability: the Kappa Statistic. *Biochemia Medica* 22, 3 (2012), 276–282.
[132] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment Analysis Algorithms and Applications: A Survey. *Ain Shams Engineering Journal* 5, 4 (2014), 1093–1113. https://doi.org/10.1016/j.asej.2014.04.011
[133] Damianos P Melidis, Myra Spiliopoulou, and Eirini Ntoutsi. 2018. Learning under Feature Drifts in Textual Streams. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM 2018)*. ACM, 527–536. https://doi.org/10.1145/3269206.3271717
[134] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the Workshop at International Conference on Learning Representations (ICLR 2013)*. 1–12.
[135] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NeurIPS 2013)*, Vol. 26. 3111–3119.
[136] Mahdi Naser Moghadasi and Yu Zhuang. 2020. Sent2Vec: A New Sentence Embedding Representation with Sentimental Semantic. In *Proceedings of the IEEE International Conference on Big Data (IEEE BigData 2020)*. IEEE. https://doi.org/10.1109/BIGDATA50022.2020.9378337
[137] Rami Mohawesh, Son Tran, Robert Ollington, and Shuxiang Xu. 2021. Analysis of Concept Drift in Fake Reviews Detection. *Expert Systems with Applications* 169 (2021), 114318. https://doi.org/10.1016/J.ESWA.2020.114318
[138] Jacob Montiel, Max Halford, Saulo Martiello Mastelini, Geoffrey Bolmier, Raphaël Sourty, Robin Vaysse, Adil Zouitine, Heitor Murilo Gomes, Jesse Read, Talel Abdessalem, and Albert Bifet. 2021. River: Machine Learning for Streaming Data in Python. *Journal of Machine Learning Research* 22, 1 (2021), 4945–4952.
[139] Yida Mu, Chun Dong, Kalina Bontcheva, and Xingyi Song. 2024. Large Language Models Offer an Alternative to the Traditional Approach of Topic Modelling. In *Proceedings of the Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC/COLING 2024)*. ELRA and ICCL, 10160–10171.
[140] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie Glance. 2013. *Fake Review Detection: Classification and Analysis of Real and Pseudo Reviews*. Technical Report UIC-CS-03-2013. University of Illinois at Chicago, Chicago, IL, USA.
[141] Pierre-Alexandre Murena, Marie Al-Ghossein, Talel Abdessalem, and Antoine Cornuéjols. 2018. Adaptive Window Strategy for Topic Modeling in Document Streams. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2018)*. IEEE, 1–7. https://doi.org/10.1109/IJCNN.2018.8489771
[142] Ricardo Nanculef, Ilias Flaounas, and Nello Cristianini. 2014. Efficient classification of multi-labeled text streams by clashing. *Expert Systems with Applications* 41, 11 (2014), 5431–5450. https://doi.org/10.1016/J.ESWA.2014.02.017
[143] Peter Bjorn Nemenyi. 1963. *Distribution-free Multiple Comparisons*. Ph. D. Dissertation. Princeton University.
[144] Tung Nguyen, Trung Mai, Nam Nguyen, Linh Ngo Van, and Khoat Than. 2022. Balancing Stability and Plasticity When Learning Topic Models from Short and Noisy Text Streams. *Neurocomputing* 505 (2022), 30–43. https://doi.org/10.1016/J.NEUCOM.2022.07.019
[145] Tien Thanh Nguyen, Manh Truong Dang, Anh Vu Luong, Alan Wee-Chung Liew, Tiancai Liang, and John McCall. 2019. Multi-label classification via incremental clustering on an evolving data stream. *Pattern Recognition* 95 (2019), 96–113. https://doi.org/10.1016/J.PATCOG.2019.06.001
[146] Van-Son Nguyen, Duc-Tung Nguyen, Linh Ngo Van, and Khoat Than. 2019. Infinite Dropout for Training Bayesian Models from Data Streams. In *Proceedings of the IEEE International Conference on Big Data (IEEE BigData 2019)*. IEEE, 125–134. https://doi.org/10.1109/BIGDATA47090.2019.9005544
[147] Frank Nielsen. 2019. On the Jensen–Shannon Symmetrization of Distances Relying on Abstract Means. *Entropy* 21, 5 (2019), 485. https://doi.org/10.3390/E21050485
[148] Kyosuke Nishida, Takahide Hoshide, and Ko Fujimura. 2012. Improving Tweet Stream Classification by Detecting Changes in Word Probability. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2012)*. ACM, 971–980. https://doi.org/10.1145/2348283.2348412
[149] Jeppe Nørregaard, Benjamin D Horne, and Sibel Adalı. 2019. NELA-GT-2018: A large multi-labelled news dataset for the study of misinformation in news articles. In *Proceedings of the 13th International AAAI Conference on Web and*

Social Media (ICWSM 2019), Vol. 13. 630–638. https://doi.org/10.1609/icwsm.v13i01.3261

[150] Alexandra Olteanu, Sarah Vieweg, and Carlos Castillo. 2015. What to Expect When the Unexpected Happens: Social Media Communications Across Crises. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW 2015)*. ACM, 994–1009. https://doi.org/10.1145/2675133.2675242

[151] Santiago Ontañón. 2020. An Overview of Distance and Similarity Functions for Structured Data. *Artificial Intelligence Review* 53, 7 (2020), 5309–5351. https://doi.org/10.1007/S10462-020-09821-W

[152] Aljaž Osojnik, Panče Panov, and Sašo Džeroski. 2017. Multi-label classification via multi-target regression on data streams. *Machine Learning* 106, 6 (2017), 745–770. https://doi.org/10.1007/S10994-016-5613-5

[153] Ewan S Page. 1954. Continuous Inspection Schemes. *Biometrika* 41, 1/2 (1954), 100–115. https://doi.org/10.2307/2333009

[154] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-performance Deep Learning Library. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vol. 32. 8024–8035.

[155] Megha Ashok Patil, Sunil Kumar, Sandeep Kumar, and Muskan Garg. 2021. Concept Drift Detection for Social Media: A Survey. In *Proceedings of the 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N 2021)*. IEEE, 12–16. https://doi.org/10.1109/ICAC3N53548.2021.9725548

[156] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 85 (2011), 2825–2830.

[157] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*. Association for Computational Linguistics, 1532–1543. https://doi.org/10.3115/V1/D14-1162

[158] Francesco Periti, Pierluigi Cassotti, Haim Dubossarsky, and Nina Tahmasebi. 2024. Analyzing Semantic Change through Lexical Replacements. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*. Association for Computational Linguistics, 4495–4510. https://doi.org/10.18653/V1/2024.ACL-LONG.246

[159] Francesco Periti, Alfio Ferrara, Stefano Montanelli, and Martin Ruskov. 2022. What is Done is Done: an Incremental Approach to Semantic Shift Detection. In *Proceedings of the 3rd International Workshop on Computational Approaches to Historical Language Change (LChange@ACL 2022)*. Association for Computational Linguistics, 33–43. https://doi.org/10.18653/V1/2022.LCHANGE-1.4

[160] Francesco Periti and Stefano Montanelli. 2024. Lexical Semantic Change through Large Language Models: a Survey. *Comput. Surveys* 56, 11, Article 282 (2024), 38 pages. https://doi.org/10.1145/3672393

[161] Francesco Periti, Sergio Picascia, Stefano Montanelli, Alfio Ferrara, and Nina Tahmasebi. 2023. Studying word meaning evolution through incremental semantic shift detection. *Language Resources and Evaluation* (2023), 1–37. https://doi.org/10.1007/s10579-024-09769-1

[162] Francesco Periti and Nina Tahmasebi. 2024. Towards a complete solution to lexical semantic change: An extension to multiple time periods and diachronic word sense induction. In *Proceedings of the 5th Workshop on Computational Approaches to Historical Language Change*. 108–119. https://doi.org/10.18653/v1/2024.lchange-1.10

[163] Xuan-Hieu Phan, Cam-Tu Nguyen, Dieu-Thu Le, Le-Minh Nguyen, Susumu Horiguchi, and Quang-Thuy Ha. 2010. A Hidden Topic-based Framework toward Building Applications with Short Web Documents. *IEEE Transactions on Knowledge and Data Engineering* 23, 7 (2010), 961–976. https://doi.org/10.1109/TKDE.2010.27

[164] Daniela Pohl, Abdelhamid Bouchachia, and Hermann Hellwagner. 2018. Batch-based Active Learning: Application to Social Media Data for Crisis Management. *Expert Systems with Applications* 93 (2018), 232–244. https://doi.org/10.1016/J.ESWA.2017.10.026

[165] Marco Polignano, Pierpaolo Basile, Marco De Gemmis, Giovanni Semeraro, Valerio Basile, et al. 2019. AlBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks based on Tweets. In *Proceedings of the 6th Italian Conference on Computational Linguistics (CLiC-it 2019) (CEUR Workshop Proceedings, Vol. 2481)*. CEUR-WS.org, 1–6.

[166] Nitesh Pradhan, Manasi Gyanchandani, and Rajesh Wadhvani. 2015. A Review on Text Similarity Technique used in IR and its Application. *International Journal of Computer Applications* 120, 9 (2015), 29–34. https://doi.org/10.5120/21257-4109

[167] Christoph Raab, Moritz Heusinger, and Frank-Michael Schleif. 2020. Reactive Soft Prototype Computing for Concept Drift Streams. *Neurocomputing* 416 (2020), 340–351. https://doi.org/10.1016/J.NEUCOM.2019.11.111

[168] Ella Rabinovich, Matan Vetzler, Samuel Ackerman, and Ateret Anaby-Tavor. 2023. Reliable and Interpretable Drift Detection in Streams of Short Texts. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: Industry Track (ACL 2023)*. Association for Computational Linguistics, 438–446. https://doi.org/10.18653/V1/2023.ACL-INDUSTRY.42

[169] Idris Rabiu, Naomie Salim, Maged Nasser, Aminu Da'u, Taiseer Abdalla Elfadil Eisa, and Mhassen Elnour Elneel Dalam. 2023. Drift Detection Method Using Distance Measures and Windowing Schemes for Sentiment Classification. *Computers, Materials & Continua* 74, 3 (2023), 6001–6017. https://doi.org/10.32604/cmc.2023.035221

[170] Idris Rabiu, Naomie Salim, Maged Nasser, Faisal Saeed, Waseem Alromema, Aisha Awal, Elijah Joseph, and Amit Mishra. 2022. Ensemble Method for Online Sentiment Classification Using Drift Detection-Based Adaptive Window Method. In *Proceedings of the 6th International Conference of Reliable Information and Communication Technology (IRICT 2021) (Lecture Notes on Data Engineering and Communications Technologies, Vol. 127)*. 117–128. https://doi.org/10.1007/978-3-030-98741-1_11

[171] Adir Rahamim, Guy Uziel, Esther Goldbraich, and Ateret Anaby Tavor. 2023. Text augmentation using dataset reconstruction for low-resource classification. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, 7389–7402. https://doi.org/10.18653/V1/2023.FINDINGS-ACL.466

[172] Vyas Raina, Adian Liusie, and Mark Gales. 2024. Is LLM-as-a-Judge Robust? Investigating Universal Adversarial Attacks on Zero-shot LLM Assessment. *arXiv preprint arXiv:2402.14016* (2024).

[173] Md Rashadul Hasan Rakib, Norbert Zeh, and Evangelos Milios. 2021. Efficient Clustering of Short Text Streams using Online-offline Clustering. In *Proceedings of the 21st ACM Symposium on Document Engineering (DocEng 2021)*. ACM, Article 5, 10 pages. https://doi.org/10.1145/3469096.3469866

[174] Shebuti Rayana and Leman Akoglu. 2015. Collective Opinion Spam Detection: Bridging Review Networks and Metadata. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2015)*. ACM, 985–994. https://doi.org/10.1145/2783258.2783370

[175] Jesse Read, Albert Bifet, Geoff Holmes, and Bernhard Pfahringer. 2012. Scalable and efficient multi-label classification for evolving data streams. *Machine Learning* 88, 1-2 (2012), 243–272. https://doi.org/10.1007/S10994-012-5279-6

[176] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, 45–50.

[177] Jason D Rennie, Lawrence Shih, Jaime Teevan, and David R Karger. 2003. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In *Proceedings of the 20th International Conference on Machine Learning (ICML 2003)*. AAAI, 616–623.

[178] Gordon J Ross, Niall M Adams, Dimitris K Tasoulis, and David J Hand. 2012. Exponentially Weighted Moving Average Charts for Detecting Concept Drift. *Pattern Recognition Letters* 33, 2 (2012), 191–198. https://doi.org/10.1016/J.PATREC.2011.08.019

[179] Sumegh Roychowdhury, Karan Gupta, Siva Rajesh Kasa, and Prasanna Srinivasa Murthy. 2024. Tackling Concept Shift in Text Classification using Entailment-style Modeling. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2024)*. ACM, 5647–5656. https://doi.org/10.1145/3637528.3671541

[180] Anastasiia Ryzhova, Daria Ryzhova, and Ilya Sochenkov. 2021. Detection of Semantic Changes in Russian Nouns with Distributional Models and Grammatical Features. In *Proceedings of the International Conference on Computational Linguistics and Intellectual Technologies (Dialogue 2021)*. 597–606. https://doi.org/10.28995/2075-7182-2021-20-597-606

[181] Gerard Salton and Christopher Buckley. 1988. Term-weighting Approaches in Automatic Text Retrieval. *Information Processing & Management* 24, 5 (1988), 513–523. https://doi.org/10.1016/0306-4573(88)90021-0

[182] Evan Sandhaus. 2008. The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia* 6, 12 (2008), e26752.

[183] Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. In *Proceedings of the 14th Workshop on Semantic Evaluation (SemEval@COLING 2020)*. International Committee for Computational Linguistics, 1–23. https://doi.org/10.18653/V1/2020.SEMEVAL-1.1

[184] Raquel Sebastião and José Maria Fernandes. 2017. Supporting the Page-Hinkley test with Empirical Mode Decomposition for Change Detection. In *Proceedings of the 23rd International Symposium on Foundations of Intelligent Systems (ISMIS 2017) (Lecture Notes in Computer Science, Vol. 10352)*. Springer, 492–498. https://doi.org/10.1007/978-3-319-60438-1_48

[185] Qinfeng Shi, James Petterson, Gideon Dror, John Langford, Alex Smola, and SVN Vishwanathan. 2009. Hash kernels for structured data. *Journal of Machine Learning Research* 10 (2009), 2615–2637. https://doi.org/10.5555/1577069.1755873

[186] Nickolay Smirnov. 1948. Table for Estimating the Goodness of Fit of Empirical Distributions. *The Annals of Mathematical Statistics* 19, 2 (1948), 279–281.

[187] Eduardo Soares, Cristiano Garcia, Ricardo Poucas, Heloisa Camargo, and Daniel Leite. 2019. Evolving Fuzzy Set-based and Cloud-based Unsupervised Classifiers for Spam Detection. *IEEE Latin America Transactions* 17, 9 (2019),

1449–1457. https://doi.org/10.1109/TLA.2019.8931138

[188] Mahboubeh Soleymanian, Hoda Mashayekhi, and Marziea Rahimi. 2024. An incremental clustering algorithm based on semantic concepts. *Knowledge and Information Systems* 66, 6 (2024), 3303–3335. https://doi.org/10.1007/S10115-024-02063-0

[189] Ian Stewart, Dustin Arendt, Eric Bell, and Svitlana Volkova. 2017. Measuring, Predicting and Visualizing Short-term Change in Word Representation and Usage in VKontakte Social Network. In *Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017)*. AAAI, 672–675.

[190] Zhaochen Su, Zecheng Tang, Xinyan Guan, Juntao Li, Lijun Wu, and Min Zhang. 2022. Improving temporal generalization of pre-trained language models with lexical semantic change. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)*. Association for Computational Linguistics, 6380–6393. https://doi.org/10.18653/V1/2022.EMNLP-MAIN.428

[191] Gang Sun, Zhongxin Wang, Zhengqi Ding, and Jia Zhao. 2021. An Ensemble Classification Algorithm for Short Text Data Stream with Concept Drifts. *IAENG International Journal of Computer Science* 48, 4 (2021), 1–6.

[192] Abhijit Suprem, Aibek Musaev, and Calton Pu. 2019. Concept Drift Adaptive Physical Event Detection for Social Media Streams. In *Proceedings of the 15th World Congress on Services (SERVICES 2019) at Services Conference Federation (SCF 2019) (Lecture Notes in Computer Science, Vol. 11517)*. Springer, 92–105. https://doi.org/10.1007/978-3-030-23381-5_7

[193] Abhijit Suprem and Calton Pu. 2019. ASSED: a Framework for Identifying Physical Events Through Adaptive Social Sensor Data Filtering. In *Proceedings of the 13th ACM International Conference on Distributed and Event-based Systems (DEBS 2019)*. ACM, 115–126. https://doi.org/10.1145/3328905.3329510

[194] Abhijit Suprem and Calton Pu. 2019. Event Detection in Noisy Streaming Data with Combination of Corroborative and Probabilistic Sources. In *Proceedings of the 5th IEEE International Conference on Collaboration and Internet Computing (CIC 2019)*. IEEE, 168–177. https://doi.org/10.1109/CIC48465.2019.00029

[195] E Susi and AP Shanthi. 2023. Sentiment Drift Detection and Analysis in Real Time Twitter Data Streams. *Computer Systems Science & Engineering* 45, 3 (2023), 3231–3246. https://doi.org/10.32604/CSSE.2023.032104

[196] Nina Tahmasebia, Lars Borina, and Adam Jatowtb. 2021. Survey of Computational Approaches to Lexical Semantic Change Detection. *Computational Approaches to Semantic Change* (2021), 1–91. https://doi.org/10.5281/zenodo.5040302

[197] Bruno Siedekum Thuma, Pedro Silva de Vargas, Cristiano Garcia, Alceu de Souza Britto Jr, and Jean Paul Barddal. 2023. Benchmarking Feature Extraction Techniques for Textual Data Stream Classification. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2023)*. IEEE, 1–8. https://doi.org/10.1109/IJCNN54540.2023.10191369

[198] Bach Tran, Anh Duc Nguyen, Linh Ngo Van, and Khoat Than. 2021. Dynamic Transformation of Prior Knowledge into Bayesian Models for Data Streams. *IEEE Transactions on Knowledge and Data Engineering* 35, 4 (2021), 3742–3750. https://doi.org/10.1109/TKDE.2021.3139469

[199] Meng-Hsiu Tsai, Yingfeng Wang, Myungjae Kwak, and Neil Rigole. 2019. A Machine Learning based Strategy for Election Result Prediction. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 1408–1410.

[200] Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. 2022. *Natural language processing with transformers.* O'Reilly Media, Inc., Sebastopol, CA, USA.

[201] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605.

[202] Ngo Van Linh, Tran Xuan Bach, and Khoat Than. 2022. A Graph Convolutional Topic Model for Short and Noisy Text Streams. *Neurocomputing* 468 (2022), 345–359. https://doi.org/10.1016/J.NEUCOM.2021.10.047

[203] Tham Vo. 2022. GOWSeqStream: an Integrated Sequential Embedding and Graph-of-words for Short Text Stream Clustering. *Neural Computing and Applications* 34, 6 (2022), 4321–4341. https://doi.org/10.1007/S00521-021-06563-W

[204] Kun Wang, Jie Lu, Anjin Liu, and Guangquan Zhang. 2023. TCR-M: A Topic Change Recognition-based Method for Data Stream Learning. In *Proceedings of the 27th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2023) (Procedia Computer Science, Vol. 225)*. Elsevier, 3001–3010. https://doi.org/10.1016/J.PROCS.2023.10.293

[205] Zhenhua Wang, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. 2014. On Summarization and Timeline Generation for Evolutionary Tweet Streams. *IEEE Transactions on Knowledge and Data Engineering* 27, 5 (2014), 1301–1315. https://doi.org/10.1109/TKDE.2014.2345379

[206] Zhuoyi Wang, Hemeng Tao, Zelun Kong, Swarup Chandra, and Latifur Khan. 2019. Metric Learning Based Framework for Streaming Classification with Concept Evolution. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2019)*. IEEE, 1–8. https://doi.org/10.1109/IJCNN.2019.8851934

[207] Scott Wares, John Isaacs, and Eyad Elyan. 2019. Data Stream Mining: Methods and Challenges for Handling Concept Drift. *SN Applied Sciences* 1, Article 1412 (2019), 19 pages. https://doi.org/10.1007/s42452-019-1433-0

[208] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Huggingface's Transformers: State-of-the-art Natural Language Processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2020)*. Association for Computational Linguistics, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

[209] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. Probase: A Probabilistic Taxonomy for Text Understanding. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2012)*. ACM, 481–492. https://doi.org/10.1145/2213836.2213891

[210] Shuiqiao Yang, Guangyan Huang, Xiangmin Zhou, Vicky Mak, and John Yearwood. 2021. EWNStream +: Effective and Real-time Clustering of Short Text Streams using Evolutionary Word Relation Network. *International Journal of Information Technology & Decision Making* 20, 1 (2021), 341–370. https://doi.org/10.1142/S0219622021500024

[211] Jianhua Yin, Daren Chao, Zhongkun Liu, Wei Zhang, Xiaohui Yu, and Jianyong Wang. 2018. Model-based Clustering of Short Text Streams. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2018)*. ACM, 2634–2642. https://doi.org/10.1145/3219819.3220094

[212] Frank D. Zamora-Reina, Felipe Bravo-Marquez, and Dominik Schlechtweg. 2022. LSCDiscovery: A shared task on semantic change discovery and detection in Spanish. In *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change, LChange@ACL 2022, Dublin, Ireland, May 26-27, 2022*, Nina Tahmasebi, Syrielle Montariol, Andrey Kutuzov, Simon Hengchen, Haim Dubossarsky, and Lars Borin (Eds.). Association for Computational Linguistics, 149–164. https://doi.org/10.18653/V1/2022.LCHANGE-1.16

[213] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). (2019), 75–86. https://doi.org/10.18653/V1/S19-2010

[214] YiFan Zhang, Weiqi Chen, Zhaoyang Zhu, Dalin Qin, Liang Sun, Xue Wang, Qingsong Wen, Zhang Zhang, Liang Wang, and Rong Jin. 2024. Addressing Concept Shift in Online Time Series Forecasting: Detect-then-Adapt. *arXiv preprint arXiv:2403.14949* (2024).

[215] Eric Zhao, Anqi Liu, Animashree Anandkumar, and Yisong Yue. 2021. Active learning under label shift. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS 2021) (Proceedings of Machine Learning Research, Vol. 130)*. PMLR, 3412–3420.

[216] Xiaojin Zhu and Zoubin Ghahramani. 2002. *Learning from Labeled and Unlabeled Data with Label Propagation*. Technical Report CMU-CALD-02-107. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA.

A LIST OF ACRONYMS

In order to ease the reader to locate acronyms' meanings, we developed the Table 9. This list provides the acronyms alphabetically ordered. Please, note that we did not include acronyms without a clear meaning provided by the acronym's author(s).

Table 9. List of acronyms, alphabetically ordered.

Acronym	Meaning
AdaNEN	Adaptive Neural Ensemble Method [69]
ADWIN	Adaptive Windowing [20]
AE	Autoencoder [168]
AEE	Additive Expert Ensemble [107]
AIS	Artificial Immune System
AIS-Clus	Artificial Immune System - Clustering [2, 3]
API	Application Programming Interface
ARIMA	Auto-regressive Integrated Moving Average
ARL	Average run length
ASSED	Adaptive Social Sensor Event Detection [193]
AUC	Area Under the Curve
AWILDA	Adaptive Window based Incremental LDA [141]
BERT	Bidirectional Encoder Representation from Transformers [46]

Acronym	Meaning
BOW	Bag-of-words
BSP	Balancing Stability and Plasticity [144]
CBOW	Continuous bag-of-words [134]
CCOHA	Clean Corpus of Historical American English
CFS	Correlation-based feature selection
CNB	Complement Naive Bayes
CNN	Convolutional Neural Network
CRQA	Cross Reference Quantification Analysis [41]
CUSUM	Cumulative sum
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DC	Drift categories
DCFS	Dynamic Correlation-based Feature Selection
DD	Drift detection types
DDAW	Drift Detection-based Adaptive Window [10, 170]
DDM	Drift detection method
DPMM	Dirichlet Process Multinomial Mixture
DSM	Data stream mining
EC	Exclusion criteria
EDDM	Early Drift Detection method [12]
EWMA	Exponentially Weighted Moving Average
EWNStream+	Evolutionary Word relation Network for short text Streams clustering [210]
FAR	False alarms rate
FFCA	Fuzzy Formal Concept Analysis [55]
FNN	Feedforward Neural Network
GCTM	Graph Convolutional Topic Model [202]
GDWE	Graph-based Dynamic Word Embeddings [127]
GloVe	Global Vectors [157]
GPU	Graphic Processing Unit
HDDM	Hoeffding-inequality-based Drift Detection Method [56]
IC	Inclusion criteria
IS	Intelligent Systems
IWC	Incremental Word-Context [29]
kNN	k-Nearest Neighbors
KSWIN	Kolmogorov-Smirnov Windowing [167]
LAMBADA	Language-model-based data augmentation [8]
LDA	Latent Dirichlet Allocation [24]
LLM	Large language model
LPP	Log Predictive Probability
LSTM	Long short-term memory
MDR	Missing detection rate
ML	Machine Learning
MLM	Masked language modeling
MOA	Massive Online Analysis [21, 22]
MSE	Mean squared error
MTD	Mean time to detection
MTFA	Mean time between false alarms
MU	Model update

Acronym	Meaning
NCD	Normalized Compression Distance
NLP	Natural Language Processing
NMI	Normalized Mutual Information
NOAA	National Oceanic and Atmospheric Administration
NPMI	Normalized Pointwise Mutual Information
OBAL	Online Batch-based Active Learning [164]
OFSER	Online Feature Selection with Evolving Regularization [42]
OM	Opinion mining
OOV	Out-of-vocabulary
OSMTS	Online Semi-Supervised Classification on Multilabel Text Streams [110]
PCA	Principal component analysis
PH	Page-Hinkley
PPMI	Positive Pointwise Mutual Information [29]
PSDVec	Positive-Semidefinite Vectors [120]
PV-DBOW	Paragraph vector - Distributed bag-of-words [115]
PV-DM	Paragraph vector - Distributed memory [115]
RDDM	Reactive drift detection method [40]
RoBERTa	Robustly Optimized BERT Pre-training Approach [124]
ROC	Receiver Operating Characteristic
RQ	Research Question
SA	Sentiment analysis
SMAFED	Social Media Analysis Framework for Event Detection [105]
SPC	Statistical Process Control
SVM	Support Vector Machine
t-SNE	t-Distributed Stochastic Neighbor Embedding
TCR-M	Topic Change Recognition-based Method [204]
TF-IDF	Term frequency-Inverse Document Frequency
TR	Text representation
TRUS	Text representation update scheme
TSDA-BERT	Twitter Sentiment Drift Analysis - BERT [195]
VFDT	Very Fast Decision Tree [60]
WIDID	What is Done is Done [159]

Received 20 February 2024; revised 10 October 2024; accepted XX XXXX 2024