# Event-driven Sentiment Drift Analysis in Text Streams: An Application in a Soccer Match

Cristiano Mesquita Garcia
*Graduate Program in Informatics (PPGIa)*
*Pontifícia Universidade Católica do Paraná (PUCPR)*
*Instituto Federal de Santa Catarina - Caçador*
Curitiba, Brazil
cristiano.garcia@ifsc.edu.br

Alceu de Souza Britto Jr., Jean Paul Barddal
*Graduate Program in Informatics (PPGIa)*
*Pontifícia Universidade Católica do Paraná (PUCPR)*
Curitiba, Brazil
{alceu, jean.barddal}@ppgia.pucpr.br

***Abstract*—Social media has been a data source for various applications, given its characteristic of working as a social sensor. Many applications in several areas, such as brand reputation and online opinion monitoring, use this valuable resource to understand the users of services and products. This paper describes an application in the soccer domain, considering data collected from a social media textual data stream. The goal is to detect possible sentiment drifts related to actual events in a soccer match. This task is challenging as we resort to short texts made available during a short time (match length). We evaluated four drift detectors using four metrics: false alarms, delay (considering the number of posts), delay, and missing drifts. Our results show that ADWIN had a stable performance in sentiment drift detection compared to other methods in timely detecting the flagged drifts, raising a small number of false alarms. Given the drifts detected, we used Incremental Word-Vectors to monitor words of interest and check their relatedness to actual events in the match. We empirically assert that the closest words trace back to the sentiment drift generator events.**

***Index Terms*—concept drift, text stream, sentiment drift, sentiment analysis, drift analysis**

## I. Introduction

Data has been considered the "new oil" given its importance in several aspects, either by explaining past events and processes or, with the aid of machine learning, predicting future events and behavior. Furthermore, some authors categorize social media as social sensors, given that they are continuously generating data [1], and thus, much attention has been paid to this specific data source. For instance, authors in [2] and [3] analyzed data from social media to understand public opinion about vaccination in Italy and opinions on politics and public health in Brazil, respectively. Also, it is possible to link real-world events to social media posts. We highlight the work of [1], in which the authors aimed at detecting landslide events by combining social media posts with governmental reports.

It is well-known that the world is not static, as changes may occur. An example is purchasing patterns before and after the COVID-19 pandemic, or even in how meetings took place since social distancing was needed to avoid virus spreads. In addition to processes, people and their opinions change over time. Consequently, the data that reflect individuals, their opinions, or processes may change over time, thus giving rise to a phenomenon called concept drift [4]. Concept drift regards changes in the data distribution over time due to changes in the actual data generator processes. Static machine learning-based methods have their performance degraded in the presence of concept drift, especially in streaming environments. In this work, we are interested in sentiment drift detection, which corresponds to detecting changes in the sentiment, e.g., opinions and attitudes, over time.

Data streams are "an algorithmic abstraction to support real-time analytics" [4]. This statement carries a pertinent aspect: data are processed individually or in small batches on the fly. By definition, data streams are fast and have the potential to be infinite, which corresponds to problems for traditional machine learning methods. In this paper, we are interested in "text streams," which are data streams formed by a sequence of texts, such as posts from social media and logs from computational systems. In particular, we focus on scenarios where machine learning-based methods degrade when classifying textual data streams. A common approach to overcome this challenge is detecting a drift using a detector and updating the machine learning model accordingly. Regarding social media, concept drift can emerge in changes in stance, sentiment, opinions, topic shift, and others. To better understand these changes in text data, we resort to text mining techniques, such as sentiment analysis, in which we target the understanding of polarities, opinions, and emotions in text [5].

This paper analyzes sentiment drift in the context of a soccer match. Soccer is one of the most popular sports worldwide, involving much passion, and the supporters' sentiments may vary rapidly and frequently. The analysis establishes links between detected changes and actual events in the soccer match. These links are measured using delay, false alarms, and missing drifts, regular metrics for concept drift detection. We further analyze the sentiment over the match to confirm the existence of sentiment drifts and investigate the closest words to words of interest that share the same context to compare with the actual events that triggered the sentiment drift.

The contribution of this work is two-fold: (i) an analysis of sentiment drift detection based on real-world events under a very extreme scenario considering short texts and a limited period, and (ii) a new dataset in Brazilian Portuguese with flagged sentiment drifts. The latter is relevant since most text

corpora available for the scientific community are in English.

The paper is organized as follows. Section II introduces the concepts used in this paper, including data and text streams, textual data stream mining, and concept drift. Section III describes the methodology, including the context, data collection, sentiment classification, implementation, and evaluation protocol. Section IV discusses the results. Finally, Section V concludes this work and states envisioned future work.

## II. Background

Data streams are "an algorithmic abstraction to support real-time analytics" [4]. In data stream settings, data arrive individually or in small batches, temporally ordered. Also, data streams can be fast and potentially infinite, thus hampering traditional machine learning methods from working correctly.

In this paper, we use the terminology "text stream" to indicate a data stream formed by a sequence of texts, respecting the properties of the regular data streams. An example of a text stream is tweets obtained in real-time using its respective API (application programming interface). Analogously, this logic can be applied to other social media.

### A. Text Stream Mining

Data Stream Mining concerns methods for learning from data streams. However, for machine learning methods, several restrictions must be respected [4]: (a) the input must be inspected only once; (b) the learning process cannot take too long; (c) a small amount of memory can be used; and (d) the model needs to be ready to provide prediction whenever it is requested; and (e) it must adapt to changes over time.

When learning from text streams, i.e., Text Stream Mining, the task becomes even more complex, and different details must be accounted for. For example, the vocabulary (tokens that appear in the stream) size must be limited; text representation, e.g., Bag-of-Words [6], BERT [7], or Word2Vec [8]; must be updated swiftly, among others. Besides, decisions on strategies to keep vocabulary within a fixed size also impact processing time and representation quality.

### B. Concept Drift

In any environment subject to dynamic behavior, changes in data distribution can impact processes that depend on this data. These changes are called concept drift. In [9], the authors define four types of changes: (a) sudden, in which the data distribution changes in an abrupt manner; (b) incremental, in which the data distribution changes over time; (c) gradual, in which the data distribution switches to a new distribution and back to the older distribution, until keep in the new one; and (d) reoccurring, which the data distribution goes to other distribution, and after a while, comes back to the older one.

The definitions above can also be transposed to polarity/sentiment scenarios. Fig. 1 shows a graphical representation of sentiment drift, respecting the definitions in [9]. In this paper, we are concerned with sentiment drifts similar to sudden or incremental drift, although we do not make distinctions.

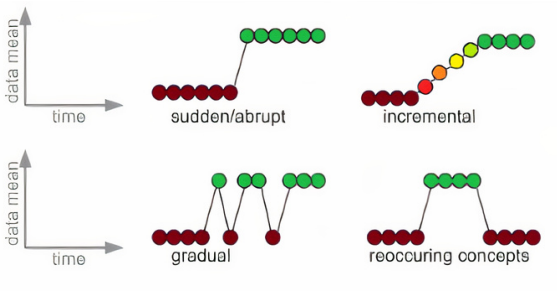


Fig. 1. Visual representation of types of polarity/sentiment drift [10].

### C. Drift Detectors

Drift detectors flag statistically significant changes while receiving the data stream. Regular concept drift detection methods have a base learner, which provides input for the drift detector [11]. In a general stream classification setting, the correct and incorrect predictions serve as input for the drift detector. If the detector identifies a significant change in the input behavior, i.e., classification rates are degrading, it is necessary to update the classifier. Considering a text stream formed by tweets, we used as input for the sentiment drift detector exactly the sentiment assigned for the tweet. Thus, the hypothesis is that the sentiment drift detector can raise an alarm when the tweets change sentiment in the stream.

In this paper, we considered four drift detectors: (i) ADWIN [12], a method that uses sliding windows of variable length and dynamically checks for distribution changes; (ii) Early Drift Detection Method (EDDM) [13], which uses an internal classifier and monitors the distance between classification errors to raise a drift; (iii and iv) Hoeffding's Drift Detection Method [14] in both flavors: weighted ($\text{HDDM}_W$) and averaged ($\text{HDDM}_A$), which uses Hoeffding's bounds to determine if a drift occurred. The difference between $\text{HDDM}_W$ and $\text{HDDM}_A$ is that the former gives more importance to recent data when comparing the moving averages, while the latter equally weights the data.

## III. Experimental Protocol

In this paper, we bring forward an experimental approach to detect sentiment drifts in a soccer match, and thus, it is essential to describe the context of this match. The soccer match of interest was Sport Club Internacional (hereafter Internacional) versus Club Social y Deportivo Colo-Colo (hereafter Colo-Colo). Internacional is a Brazilian soccer club based in Porto Alegre, whereas Colo-Colo is a Chilean club from Santiago. The dataset corresponds to the match played on July $5^{th}$, 2022 (July $6^{th}$ 00h30 UTC). This match was the second leg (out of 2) of the final 16 of *Copa Sudamericana* (second-tier Latin American championship of soccer clubs). The first match played on June $28^{th}$, 2022 in Santiago, Chile, ended Colo-Colo 2 - 0 Internacional, and its data was not collected.

After the first match, there was a consensus in the traditional Brazilian media that the result was terrible for Internacional. Conversely, *colorado* (title of the Internacional's supporters)

influencers made a sequence of Youtube videos inviting the Internacional's supporters to make a *rua de fogo* ("street of fire", in English) to welcome the Internacional's players and also to fill the *Beira-Rio*[1] stadium.

With 2-0, Colo-Colo could lose the second match by one goal of difference and be classified. Down by two goals, the decision would take place in the penalty shootout. Any negative result for Colo-Colo over two goals would qualify Internacional for the quarterfinals. The second match ended up 4-1 for Internacional. We describe some of the main events in this match. Around 12' (00h42 UTC), Internacional's goalkeeper Daniel committed a penalty. Colo-Colo's midfielder Gabriel Costa scored a goal at 14' (00h44 UTC). However, at 28' (00h58 UTC), Internacional's midfielder Alan Patrick scored a goal. Three minutes later, Internacional's midfielder Edenilson scored another goal. In the second half, at 14' (01h50), Internacional's striker Alexandre "Alemão" scored a goal. This result would take the decision to the penalty shootout. However, at 28' (02h04 UTC), Internacional's right winger Pedro Henrique scored the fourth goal, directly classifying the team for the quarterfinals. No other goals were scored in this match. The names of the soccer players were not omitted since it is crucial for the sentiment drift analysis.

Regarding so many ups and downs, we assume that this context reflects changes in the sentiment of Internacional supporters during the match. We now describe our experimental protocol, detailing data collection, sentiment classification implementation, and evaluation that took place.

### A. Data Collection

The dataset was collected using Twitter's API[2], and corresponds to tweets between July $5^{th}$ 18:00 UTC and July $6^{th}$ 06:00 UTC. The match happened between July $6^{th}$ 00:30 UTC and July $6^{th}$ 02:30 UTC. The query for collecting Brazilian Portuguese tweets using the API was *(#scinter OR #scinternacional OR #colorado OR #inter OR @scinternacional OR inter) lang:pt -is:retweet*. These terms were tested empirically and filtered by the Portuguese language, removing retweets due to the bias it could cause in the sentiment drift analysis.

The dataset obtained contained 37,126 tweets. The sentiment classification was performed on an instance-basis, as explained in Section III-B. The original dataset contained an average length of 10.74 $\pm$ 8.97 words. After treatment (described below), the new length was 5.75 $\pm$ 4.77.

Due to the similarity between the language of both teams' supporters, i.e., Portuguese and Spanish, many tweets in Spanish appeared in the collection, even with the filter for Portuguese directly in the Twitter API. Thus, we tested the LangDetect[3] library applied as a second filtering method, yet, it did not yield significant improvements. Also, once we filtered the query for Portuguese, we analyzed possible sentiment drifts from the Internacional supporters' perspective. Later, we lowercased tweets, converted numbers to 0, and removed special characters. The *hashtags* had the # character removed. We did not fix words, e.g., "inteeeeer" $\rightarrow$ "inter", to keep the stream the closest possible to the original and show that the approach can be easily used in other streams and contexts. Also, the tokens were incrementally encoded using the Incremental Word-Vectors (IWV) [15]. This approach keeps vector representations updated. These representations were used to find the closest words (considering the cosine distance) to the words of interest whenever a sentiment drift was detected. One of IWV advantages is that it requires limited storage and memory, suitable for streaming scenarios. Since we used IWV only to generate incremental vector representations for text to check the similarity among the words, we did not test the sensibility of the parameters.

We defined two words of interest: *scinternacional*, and *inter*. The first is related to the @scinternacional, official Twitter profile, and *inter* is the choice since it is sort of a nickname for Internacional. We filtered the input to IWV by ignoring tokens with less than three characters to obtain meaningful results. We used the k-Nearest Neighbors [16] to find the ten closest words in each detection of sentiment drift. For this activity, scikit-learn's [17] implementation of k-Nearest Neighbors was used. Section IV-D describes the results of this analysis.

### B. Sentiment Classification

Classification of the users' sentiments in tweets was done in a lexicon-based approach. We used a predefined dictionary containing positive and negative words as input to label each token in each tweet. This dictionary is made available alongside the source code used in experimentation. To determine the sentiment of a tweet, Equation 1 was used, where the number of neutral, negative, and positive tokens were counted and subtracted. The number of negative (`nw`) and positive (`pw`) tokens in the tweets are counted and then subtracted. If the result is lower than 0, the tweet is considered negative; bigger than 0, positive; and equal to 0 is considered neutral.

$$\text{sentiment} = \begin{cases} \text{negative}, & \text{if } |nw| - |pw| < 0 \\ \text{positive}, & \text{if } |nw| - |pw| > 0 \\ \text{neutral}, & \text{otherwise} \end{cases} \quad (1)$$

During the processing of the text stream, the sentiment of each tweet is calculated using the aforementioned Equation, and the result is used as input for drift detectors.

### C. Implementation

In this paper, we used River [18] for experimentation since it provides reliable implementations of drift detectors and metrics. Our approach is described in Algorithm 1. The algorithm requires a text stream and a concept drift detector. It also initializes an Incremental Word-Vector (IWV) in line 1. The parameters used in IWV were: vocabulary_size = 10000, dimension_size = 100, window_size = 7. The values of vocabulary_size and window_size for IWV were chosen based on the original paper [15]. In [15], the authors state that window_sizes are commonly between 3 and 17, smaller windows can capture syntactic information, and bigger windows

[1]Internacional's soccer stadium

[2]https://developer.twitter.com/en/docs/twitter-api

[3]https://pypi.org/project/langdetect/

**Algorithm 1** Implementation

**Require:** $TS \leftarrow Text\ Stream\ (tweets)$
**Require:** $CDD \leftarrow$ Concept Drift Detector
1: $IWV \leftarrow$ IncrementalWordVectors(vocab $=$ 10000, dim $= 100$, window_size $= 7$)
2: **while** $TS \neq \varnothing$ **do**
3: $tweet \leftarrow$ next tweet from $TS$
4: Preprocess $tweet$
5: Update IWV with the new tweet
6: Compute tweet's sentiment
7: Update CDD with the new tweet's sentiment
8: **if** CDD detects a sentiment drift **then**
9: Check most similar words with the words of interest
10: **end if**
11: **end while**

ease capturing semantics. Thus, a window_size of 7 would help capture more information than syntactic only. At last, the dimension_size was chosen empirically. In the original paper, the minimum value for dimension_size in the experiments was 500. However, the aim of generating numeric representation regards only verifying essential words during the drifts in the analysis. Considering that we handled a short textual stream, the shorter dimension_size has little effect on the process. While receiving the text stream, the tweets are obtained from the stream and treated (lines 3 and 4). In line 5, the IWV is updated. The sentiment is computed in line 6, as described in Section III-B. The sentiment drift detector receives, in line 7, the sentiment computed in line 6. In line 8, the sentiment drift detector is checked for changes. If a drift is flagged, the closest words to the words of interest are analyzed (line 9).

### D. Evaluation Methods

As evaluation methods, we used regular metrics for drift detection. They are: (a) missing drifts, which counts the undetected drifts across the stream; (b) delay (time), which measures the delay between the ground-truth date time UTC and the detection moment; (c) delay (posts), which counts the tweets between the ground-truth date and time UTC and the moment of detection; and (d) false alarms, which counts the detection of false drifts flagged. This paper considers a drift missing if the drift detector does not detect it within 10 minutes after the ground-truth date and time.

## IV. Results

In this section, we first analyze the collected data and the evidence for sentiment drift, describing the results obtained by applying different drift detectors. Considering the collected data, as expected, the volume of tweets increase dramatically during the match. This behavior can be viewed in Fig. 2, where we cropped the original timeline to show the part concerning the match $\pm$ 30 minutes. Before the beginning of the match, the frequency of posts was 15.96 $\pm$ 10.32 per minute. After finishing the match, the volume of posts returns to the original

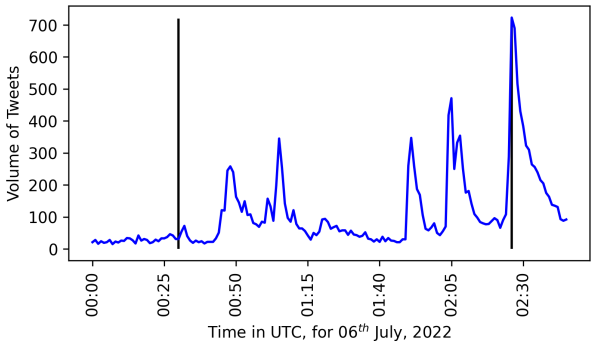


Fig. 2. Timeline of the match: the vertical lines indicate the beginning and the end of the match, respectively.

in around 2 hours. Also, we can infer that the spikes can roughly represent significant events in the match.

### A. Evidence of Sentiment Drift

Regarding sentiment drift, we binned the soccer match into six bins, corresponding to 15-minute intervals. Since every half of the match has 45 minutes plus stoppage time, there are three quarters per half-match, leading to 6 bins. In Fig. 3, we show a percentage of posts for each sentiment, disregarding neutral posts. For instance, 1H2Q means the first half and second quarter (between 15' and 30'). Also, as an example, in the 1H2Q, it is possible to notice that 64% of the non-neutral posts were negative and 36% were positive. We removed the neutral posts to ease the interpretation.

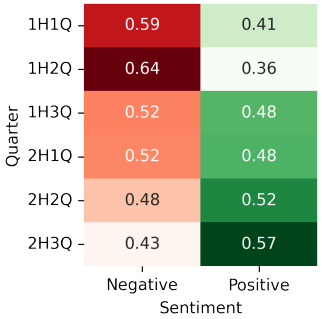


Fig. 3. Sentiment distribution during the match quarters.

Analyzing Fig. 3, we see that Internacional's supporters' sentiment was more negative in 1H1Q and 1H2Q. It can be linked to the event of Colo-Colo's goal. However, the posts' sentiments are roughly balanced at the end of the first and the beginning of the second half. With the third and fourth Internacional goals, the sentiment definitely drifts to positive.

Fig. 4 depicts similar perspectives, with data grouped by 5-minute bins and the percentages of negative, neutral, and positive tweets over time stacked, considering the match $\pm$ 30 minutes. We also plotted the sentiment average (the thickest line), corresponding to the rate of positive tweets over non-neutral tweets. The dotted line references the sentiment average, i.e., 50% positive/50% negative. Considering that the

match started at 00h30, we see a decrease in the average sentiment, which is linked to Colo-Colo's goal. The sentiment average surrounds 50% until after 02h00 when Internacional scored the fourth goal. From this point, the average sentiment keeps increasing.

### *B. Drift Detectors applied to Sentiment Drift*

To evaluate the drift detectors as indicators of sentiment drift, we initially applied four techniques: Adaptive Windowing (ADWIN) [12], Early Drift Detection Method (EDDM) [13], HDDM$_W$ and HDDM$_A$ [14], using most of the default parameters from River library, described in the Table I. We highlight the exceptions in bold. The *two_sided_test* parameter in both HDDMs was changed to true to monitor both the increase and decrease in the sentiment average. The default parameter monitors only the decrease.

TABLE I
DEFAULT PARAMETERS OF THE DRIFT DETECTORS.

| Detector | Parameters |
|---|---|
| ADWIN | $\delta = 0.002$, clock=32, max_buckets=5, min_window_length=5, grace_period=10 |
| EDDM | warm_start=30, $\alpha = 0.95$, $\beta = 0.9$ |
| HDDM$_W$ | drift_confidence=0.001, warning_confidence=0.005, $\lambda$=0.05, **two_sided_test=True** |
| HDDM$_A$ | drift_confidence=0.001, warning_confidence=0.005, **two_sided_test=True** |

First, it is opportune to describe what we used as ground truth. Since we can see from Fig. 3 that sentiment changes during the match, we defined two ground-truth scenarios: (a) the moments according to Table II, which highlight the most critical events in the match; and (b) the moments of possible inflections from negative to positive sentiments and vice-versa. These moments are highlighted in bold in Table II.

The idea is to evaluate the drift detectors as potential event detectors and sentiment drift detectors. The (b) scenario contains the events #1 (towards negative) and #4 (towards positive). The choice for event #4 happened since it is the most critical positive event before the end of the match.

To sum up, we describe three experiments: i) sentiment drift detection considering the five events shown in Table II, using the complete stream; ii) sentiment drift detection considering the two bold events in Table II, using the limited stream (time of the soccer match $\pm$ 30 minutes); and iii) sentiment drift detection considering the two bold events in Table II, using the complete stream.

These moments shown in Table II make sense since Colo-Colo's goal (event #1) generates, from the point-of-view of the Internacional's supporters, negative posts. However, from the Internacional's second goal (event #2) on, the probability of scoring a third goal and, subsequently, the chances for classification for the quarterfinals increased, thus generating posts more positive. Event #3 is the goal that represented the possibility of going to a penalty shootout. Event #4 represented the chance of going directly to the quarterfinals. Finally, event #5 concludes the direct classification of Internacional to the competition's quarterfinals.

TABLE II
GROUND TRUTH FOR SENTIMENT DRIFT. ALL THE EVENTS OCCURRED ON JULY $6^{th}$.

| # | Time | Event | Description |
|---|---|---|---|
| **1** | **00:44 UTC** | **Colo-Colo's goal** | **Goal by Gabriel Costa** |
| 2 | 01:01 UTC | Internacional's goal | Goal by Edenilson |
| 3 | 01:50 UTC | Internacional's goal | Goal by Alex. "Alemão" |
| **4** | **02:04 UTC** | **Internacional's goal** | **Goal by P. Henrique** |
| 5 | 02:26 UTC | End of the match | End of the match |

According to Section III-D, we evaluated the drift detectors in terms of Missing drifts, Delay (time), Delay (posts), and False Alarms, considering two scenarios: (a) with the complete text stream, considering the data as described in Section III-A; and (b) with the stream limited to the time of the match $\pm$ 30 minutes. The results obtained for the complete stream are shown in Table III. For the limited stream, the results for False Alarms are available in Table IV. The best values are in bold. The other results for this scenario are the same as the ones for the complete text stream.

TABLE III
DRIFT DETECTORS USED FOR SENTIMENT DRIFT DETECTION, FOR THE COMPLETE STREAM.

| | Missing | Delay (time) | Delay (posts) | F. A. |
|---|---|---|---|---|
| ADWIN | 3 | **2m21s $\pm$ 1m10s** | **1443 $\pm$ 746** | **1** |
| EDDM | 5 | - | - | 33 |
| HDDM$_W$ | 5 | - | - | - |
| HDDM$_A$ | **0** | 4m35s $\pm$ 2m35s | 2516 $\pm$ 706 | 45 |

TABLE IV
DRIFT DETECTORS USED FOR SENTIMENT DRIFT DETECTION, FOR THE LIMITED STREAM.

| | False Alarms (F. A.) |
|---|---|
| ADWIN | **1** |
| EDDM | 3 |
| HDDM$_W$ | - |
| HDDM$_A$ | 11 |

Evaluating the results in Tables III and IV, we notice that ADWIN obtained the best results in terms of delay (both time and posts) and false alarms. Conversely, the only approach that detected all the flagged drifts was HDDM$_A$. However, HDDM$_A$ had 45 false alarms in the complete stream and 11 in the limited stream. In this point, HDDM$_A$ calls too much attention to possible drifts (most of them were false alarms), while ADWIN is stable and more accurate regarding delay. Also, EDDM detected numerous false alarms, while HDDM$_W$ could not detect any drift nor trigger false alarms.

Considering only scenario (b) described in the second paragraph of Section IV-B (considering events #1 and #4 as ground-truth), we recalculated the evaluation metrics for

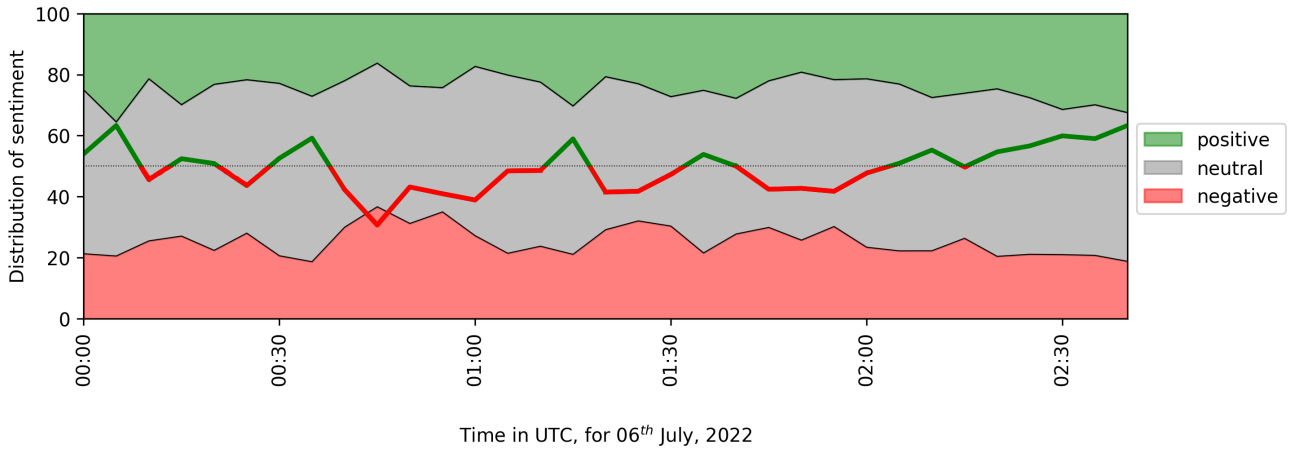


Fig. 4. Sentiment distribution during the soccer match, grouped in 5-minute bins. The thick line is the average sentiment, computed as the rate of positive tweets over non-neutral tweets.

TABLE V
DRIFT DETECTORS USED FOR SENTIMENT DRIFT DETECTION, FOR THE LIMITED STREAM, CONSIDERING SCENARIO (B).

| | Missing | Delay (time) | Delay (posts) | F. A. |
|---|---|---|---|---|
| ADWIN | **0** | **2m21s ± 1m10s** | **1443 ± 746** | **1** |
| EDDM | 2 | - | - | 33 |
| $HDDM_W$ | 2 | - | - | - |
| $HDDM_A$ | **0** | 4m35s ± 2m35s | 2516 ± 706 | 48 |

the limited stream. Table V shows the results. The results, compared to scenario (a) described in the second paragraph of Section IV-B (considering all events as ground-truth), change only in terms of Missing drifts.

With the results shown in Table V, we state that ADWIN was the most suitable sentiment drift detector among the detectors tested for the problem described in this paper. Hereafter, the analyses are performed considering the limited stream and ADWIN as a sentiment drift detector. To better understand the behaviors of EDDM and HDDM, we performed a parameters sensibility test, described in Section IV-E.

### C. Visual Inspection of Sentiment Drift Detection

Fig. 5 depicts ADWIN's ability to detect sentiment drifts. In this plot, we discarded the single false alarm.

Fig. 5 shows the timeline of tweets grouped per minute. The bars were accumulated, and only the tweets flagged as positive or negative were plotted. The neutral tweets were omitted to ease the visualization. The vertical dashed lines in black represent the ground truth of drifts, while the blue dashed lines represent the drift points detected by ADWIN. As shown in Table V, it is possible to highlight the closeness of ADWIN's detection and the ground truths.

### D. Analysis of Important Words

In this analysis, we focus on finding the closest words to two words of interest, considering sentiment drift detection moments. We used Incremental Word-Vectors to learn text

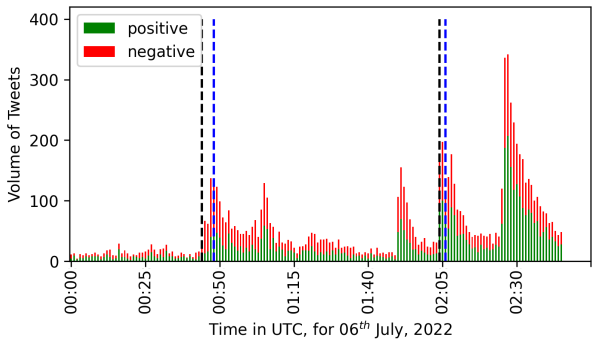


Fig. 5. Timeline of the match: the vertical black dashed lines indicate the ground truth of drifts, while the blue dashed lines are the ADWIN's drift detection.

vector representations, reset every 30 iterations. The goal is to measure cosine similarity between words of interest and words from the context, which can indicate events in the stream.

Using vector representations generated by IWV, we measured the cosine distance between the words of interest *@scinternacional* and *inter* in the moments pointed out by ADWIN as sentiment drifts. The results are shown in Table VI. The words were translated from Portuguese. We replaced the insults with an asterisk (*).

We can link the sentiment drifts with the events in the soccer match by analyzing the closest words to the words of interest, i.e., terms. For *@scinternacional*, in the event #1, we can see the existence of two insults/bad words, leading the sentiment towards the negative. Also, two names called the attention: Daniel and Moisés. Daniel was the Internacional's goalkeeper. At the play's origin, Moisés, Internacional's left back, could not cushion the ball, which went in Daniel's direction. However, Colo-Colo's striker Lucero attacked the ball before Daniel, who committed a penalty. For the term *inter*, the closest words are generic, making it hard to link to the event #1.

TABLE VI
CLOSEST WORDS TO THE WORDS OF INTEREST, IN ORDER OF CLOSENESS.

| Event | Term | Words |
|---|---|---|
| #1 | @scinternacional | "supporters", *, "to smother", "Daniel", "important", *, "scale" "got in", "necessary", "Moises" |
| | inter | "colo", "match", "tries", "pressure", "save", "pushed away", "to want", "ball", "team", "bravo" |
| #4 | @scinternacional | "Pedro", "Henrique", "alive", "stadium", "watch", "scores", "Alemão", "beira", "got in", "nobody" |
| | inter | "match", "colo", "penalty", "colorados", "scinternacional", "penalty box", "protest", "reverse the score", "penalties", "after" |

Considering the event #4 and the term *@scinternacional*, it caught the attention Pedro, Henrique, and Alemão. Pedro Henrique is the Internacional player who scored the fourth goal, while Alemão scored the third. The word "beira" also appears. It is part of the Internacional's stadium *Beira-Rio*. For the term *inter*, there are terms such as "penalty", "penalty box", and "penalties". It is unclear whether these terms relate to Colo-Colo's penalty at 14' or to the expectation of Internacional's supporters for going to the penalty shootout. However, the latter seems more probable due to the temporal proximity. There are also mentions to *colorados*, a nickname of the Internacional's supporters, and "reverse the score" (*virada*), indicating that Internacional was obtaining a positive score.

### *E. On the parameters sensibility of EDDM and HDDMs*

To investigate the results obtained primarily by EDDM and HDDM$_W$, we performed a few additional experiments. We hypothesized that: i) the input between -1 and 1 might be the source of the bad results, and ii) the default parameters might not be suitable for this problem. Regarding the former point, we then normalized the inputs to keep the input $\in [0, 1]$, yet using the default parameters of the detectors. The results are described in Table VII, considering the limited stream, i.e., during the soccer match $\pm$ 30 minutes.

TABLE VII
DRIFT DETECTORS EVALUATED USING NORMALIZED INPUT.

| | Missing | Delay (time) | Delay (posts) | F. A. |
|---|---|---|---|---|
| EDDM | 2 | - | - | 33 |
| HDDM$_W$ | 1 | 1m28s | 612 | 1 |
| HDDM$_A$ | 2 | - | - | 1 |

HDDM$_W$ found an actual change, close to the ground truth, but generated a false alarm. This false alarm corresponds to the match's end, which, in this scenario, we did not set as a ground-truth sentiment change, although it is a critical moment of the match. HDDM$_A$ could not signal the changes. However, it generated fewer false alarms (48 $>>$ 1). Finally, for EDDM, normalizing the input could not generate changes in its results. Also, EDDM did not raise false alarms during the match.

Considering the specific parameters, for HDDM$_W$ and HDDM$_A$, which share the most, we changed the drift_confidence. This parameter is set as 0.001 by default. We tested using 0.002 and 0.0005, respectively, double and a half of the original value. The results obtained for them are displayed in Table VIII, where HDDM$_W$ (0.002) means HDDM$_W$ with the drift_confidence parameter set as 0.002.

TABLE VIII
DRIFT DETECTORS EVALUATED USING NORMALIZED INPUT.

| | Missing | Delay (time) | Delay (posts) | F. A. |
|---|---|---|---|---|
| HDDM$_W$ (0.0005) | 1 | 7m58s | 2149 | 0 |
| HDDM$_W$ (0.002) | 1 | 7m58s | 2149 | 2 |
| HDDM$_A$ (0.0005) | 1 | 3m41s | 671 | 1 |
| HDDM$_A$ (0.002) | 0 | 5m23s $\pm$ 3m33s | 1295 $\pm$ 1196 | 0 |

According to Table VIII, HDDM$_W$ has small changes for both values of drift_confidence. With the smaller value, it generated fewer false alarms. Conversely, for HDDM$_A$, the higher value for drift_confidence provided better results, comparable to ADWIN. Also, both the false alarms raised by HDDM$_W$ (0.002) were close to important events in the match: the score's reversion and the match's end. For HDDM$_A$ (0.0005), the false alarm regards the end of the match.

Regarding EDDM, it has three parameters: warm_start, $\alpha$, and $\beta$. Warm_start regards the number of inputs until it can provide alarms; $\alpha$ is the threshold for warning alarm, while $\beta$ is the threshold for drift detection. For EDDM, we evaluated the values for $\beta$, which by default is set to 0.9. Higher values for this parameter suggest a more unstable behavior. Since $\beta$ is related to the sensibility to drift alarms in EDDM, we varied the values for $\beta \in [0.5, 0.9]$ using a step of 0.1. No detection is performed unless for $\beta$ = 0.9. We tested in both the limited and complete streams. To fine-grain the sensibility, we varied the values of $\beta \in [0.8, 0.9]$ stepping by 0.01, and the number of detections performed in the complete stream was 0, 3, 3, 3, 2, 3, 3, 9, 7, 19, 33. None of the detections occurred for events in the match. Fig. 6 shows the alarms raised by EDDM and the parameter $\beta$ variation, considering both the complete and the limited streams. All the alarms were raised only when we considered the complete stream. In the limited stream, no alarm is raised.

## V. CONCLUSION AND FUTURE WORKS

In this paper, we presented an application for sentiment drift detection in the soccer domain using a Twitter textual data stream. The data was collected considering a time range that comprised a decisive soccer match. Even though we focused on a soccer application, the same rationale can be applied in several domains, including public opinion monitoring, such as brand reputation and stance detection.

The soccer domain is a complex environment because this sport involves passion, and the match length is short, thus

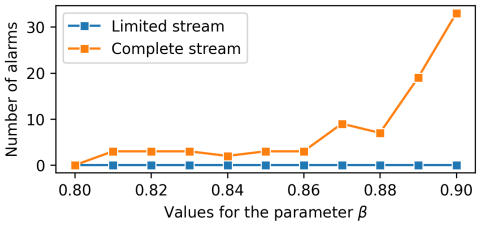


Fig. 6. Number of alarms risen by EDDM, while varying the parameter $\beta$, in the limited and complete streams.

limiting the time for real-time applications to learn. Another critical aspect that makes this work hard is that people on social media write shorter texts to express themselves as quickly as possible. The average length of the posts collected and presented in this work, i.e., 5.75 $\pm$ 4.77 words after treatment, confirms this statement.

This work applied a sentiment classifier in a text stream containing tweets on the above soccer match. This lexicon-based classifier retrieved sentiments that were used as a proxy for a concept drift detector. Considering the dataset used in this work, it is clear that sentiment drifts can happen within a short period. The results showed that ADWIN performs steadily, detecting the drifts closely to the ground truth in terms of time and posts and generating a dispensable number of false alarms. As an additional evaluation, we checked the closest words to the words of interest, namely *@scinternacional* and *inter*. We concluded that the Incremental Word-Vector incrementally generates good text representations since it retrieved words that made sense with the sentiment drift detected and, consequently, with the moment of the match.

It is important to highlight the difficulty of obtaining real-world data that contains flagged drifts. We highlight that the dataset is one of the contributions of this paper. However, this dataset is also limited since it contains only two noticeable sentiment drifts. In future works, we plan to develop a more complex sentiment classifier for text stream environments, including techniques beyond lexicon analysis. Also, we plan to collect textual data for different applications, analyze them, label their possible drifts, and make them available to the community. Other possibilities involve the classification of sentiment drift under their documented types and using a scale of sentiments instead of discrete values.

## References


[1] A. Suprem and C. Pu, "Event Detection in Noisy Streaming Data with Combination of Corroborative and Probabilistic Sources," in *2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC)*. IEEE, 2019, pp. 168–177.

[2] A. Bechini, P. Ducange, F. Marcelloni, and A. Renda, "Stance analysis of twitter users: the case of the vaccination topic in italy," *IEEE Intelligent Systems*, vol. 36, no. 5, pp. 131–139, 2020.

[3] R. F. de Mello, R. A. Rios, P. A. Pagliosa, and C. S. Lopes, "Concept Drift Detection on Social Network Data using Cross-recurrence Quantification Analysis," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, no. 8, p. 085719, 2018.

[4] A. Bifet, R. Gavalda, G. Holmes, and B. Pfahringer, *Machine Learning for Data Streams: with Practical Examples in MOA*. MIT press, 2018.

[5] W. Medhat, A. Hassan, and H. Korashy, "Sentiment Analysis Algorithms and Applications: A Survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[6] Z. S. Harris, "Distributional Structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv: 1301.3781*, 2013.

[9] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A Survey on Concept Drift Adaptation," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014.

[10] A. Bechini, A. Bondielli, P. Ducange, F. Marcelloni, and A. Renda, "Addressing Event-driven Concept Drift in Twitter Stream: A Stance Detection Application," *IEEE Access*, vol. 9, pp. 77 758–77 770, 2021.

[11] P. M. Gonçalves Jr, S. G. de Carvalho Santos, R. S. Barros, and D. C. Vieira, "A Comparative Study on Concept Drift Detectors," *Expert Systems with Applications*, vol. 41, no. 18, pp. 8144–8156, 2014.

[12] A. Bifet and R. Gavalda, "Learning from Time-changing Data with Adaptive Windowing," in *Proceedings of the 2007 SIAM International Conference on Data Mining*. SIAM, 2007, pp. 443–448.

[13] M. Baena-Garcıa, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavalda, and R. Morales-Bueno, "Early Drift Detection Method," in *Fourth International Workshop on Knowledge Discovery from Data Streams*, vol. 6, 2006, pp. 77–86.

[14] I. Frias-Blanco, J. del Campo-Ávila, G. Ramos-Jimenez, R. Morales-Bueno, A. Ortiz-Diaz, and Y. Caballero-Mota, "Online and Non-parametric Drift Detection Methods based on Hoeffding's Bounds," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 3, pp. 810–823, 2014.

[15] F. Bravo-Marquez, A. Khanchandani, and B. Pfahringer, "Incremental Word Vectors for Time-Evolving Sentiment Lexicon Induction," *Cognitive Computation*, vol. 14, no. 1, pp. 425–441, 2022.

[16] T. Cover and P. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[18] J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H. M. Gomes, J. Read, T. Abdessalem *et al.*, "River: Machine Learning for Streaming Data in Python," 2021.