Robust Social Event Detection in Twitter

Final Presentation

S. E. Bekçe & F. Amira

CS533 Spring 2016

Motivation & Goals

- Protests in Turkey are common but they don't often broadcasted by news agencies because of political pressure
- It is often not possible to understand the intensity and effects of a protest via an external view of point
- Twitter is highly used amongst people to report nearby activity
- We can use those tweets to detect such events
- Our input data contains 160M Turkish-only raw tweets from 2011 to 2014

Preprocessing: Pre-filtering

1. Discard ‘retweet’s
   Retweets contain no original information
2. Discard replies to other tweets
   Replies often introduce information redundancy and offer no original information
3. Discard non-conforming tweets: empty body, empty date, etc. (dataset was noisy)

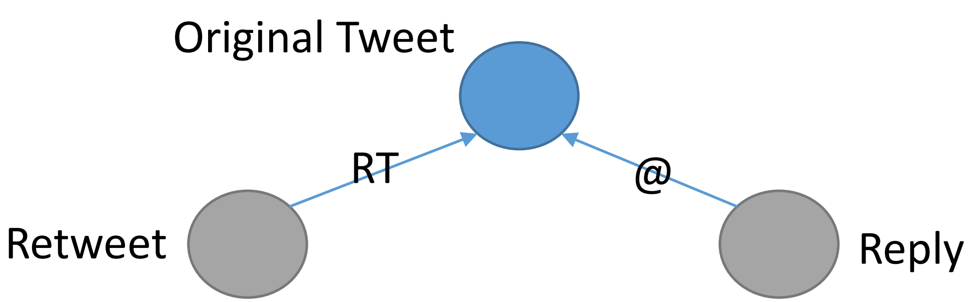


Figure 1: Retweet, Reply vs Original Tweet

Preprocessing: Standardization

1. Replace links with <url>
   Keep tweets with links because they may be photos or other important things, such as location or emergency link, etc
2. Replace mentions with <mention>
3. Convert to lowercase
4. Convert non-alphanumeric characters to spaces
5. Remove excess whitespace
6. Tokenize tweets via Zemberek [1]
   Important for finding stems for our classifier: ‘protestolar’ -> ‘protesto’

Classifier

- Design a simple classifier to classify each tweet as 'eventful' or 'not eventful'
- Based on possible keywords: (protesto, eylem, toma, saldırı, barikat, direniş)
- Also search for present tense '-yor' in the tweet
  "Ankara Kızılay'da madenci heykeli önünde Soma'daki iş cinayeti protesto ediliyor."
- Prefer tweets with embedded photos
  - First class (eventful) contains the Tweets which mention the keywords
  - Second class (not eventful) contains irrelevant Tweets

Detection – Characteristic Func.

- Generate time series (histogram) data by aggregating tweets by 5 minute intervals and counting them
- Apply Characteristic Function
  - $C(t) = \frac{STA}{mLTA+b}$ [2]
  - Short Term Average (STA): 15 minutes -> possible event
  - Long Term Average (LTA): 3 hours -> background noise
  - *m* and *b* are parameters
  - Declare event when *C(t)* > 1
  - C(t) requires higher signal levels (STA) to trigger at higher noise levels (LTA)

Serialization and Compression

- Processed ~500GB of raw (json) data and produced efficient serialized and compressed form
- Compressed form only takes 9.7GB for 160M tweets

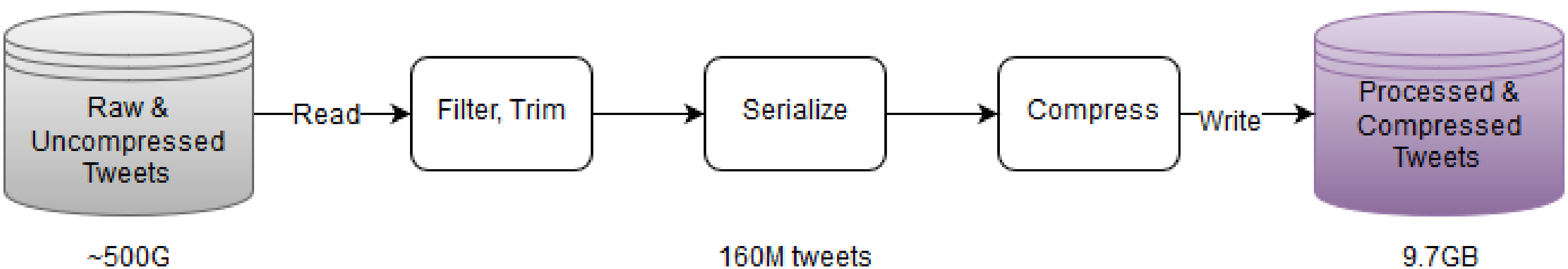


Figure 2: Serialization and Compression Steps

Algorithm – Calculating Histogram

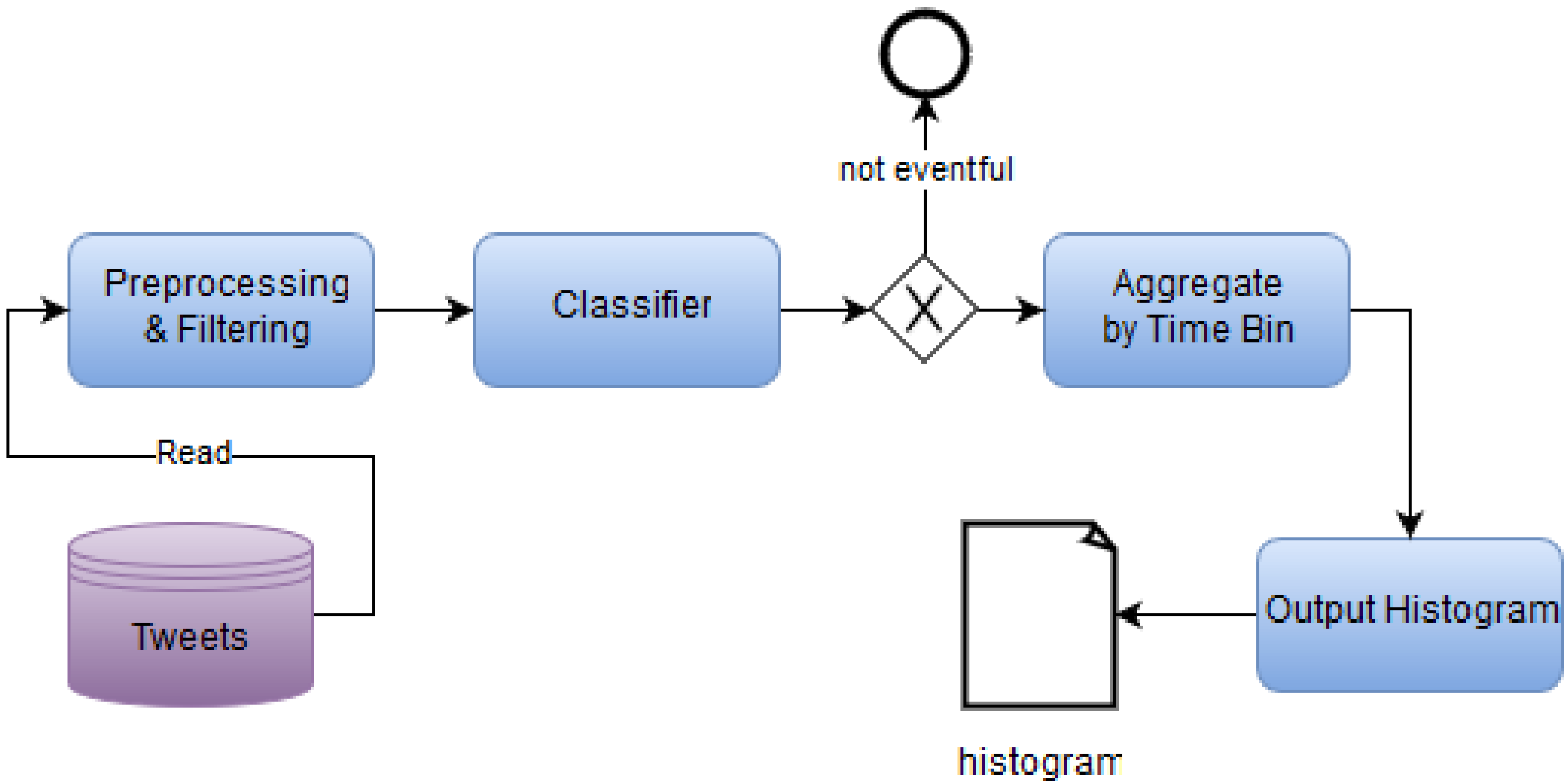


Figure 3: Histogram Calculation Steps

Algorithm – Characteristic Function and Detector

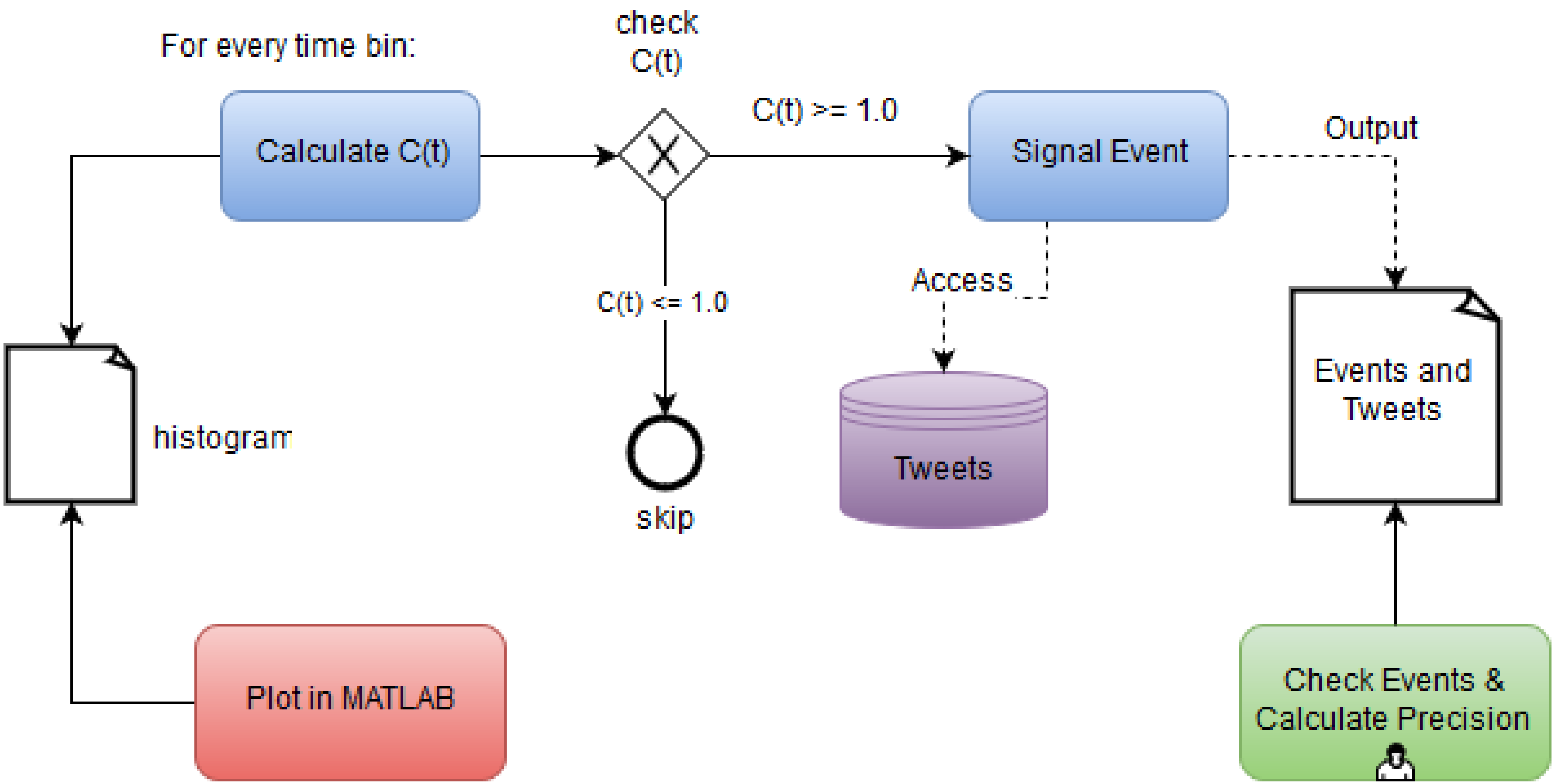


Figure 4: C(t) and Detector Steps

Design – Online Algorithm

- Designed a novel online algorithm
- Instead of working on existing tweets to detect existing events, work on ‘unseen data’ to detect ‘new events’ as they occur on the fly
- Connect to Twitter Streaming API and work on each new tweet sequentially as they arrive
- STA / LTA semantics works nicely with streaming data: Only store the last LTA window (3 hours) amount of histogram (count per window)
- It is important that standardization (tokenization, etc) operations are optimized so that the system can sustain high amount of live tweets

Design – Online Algorithm

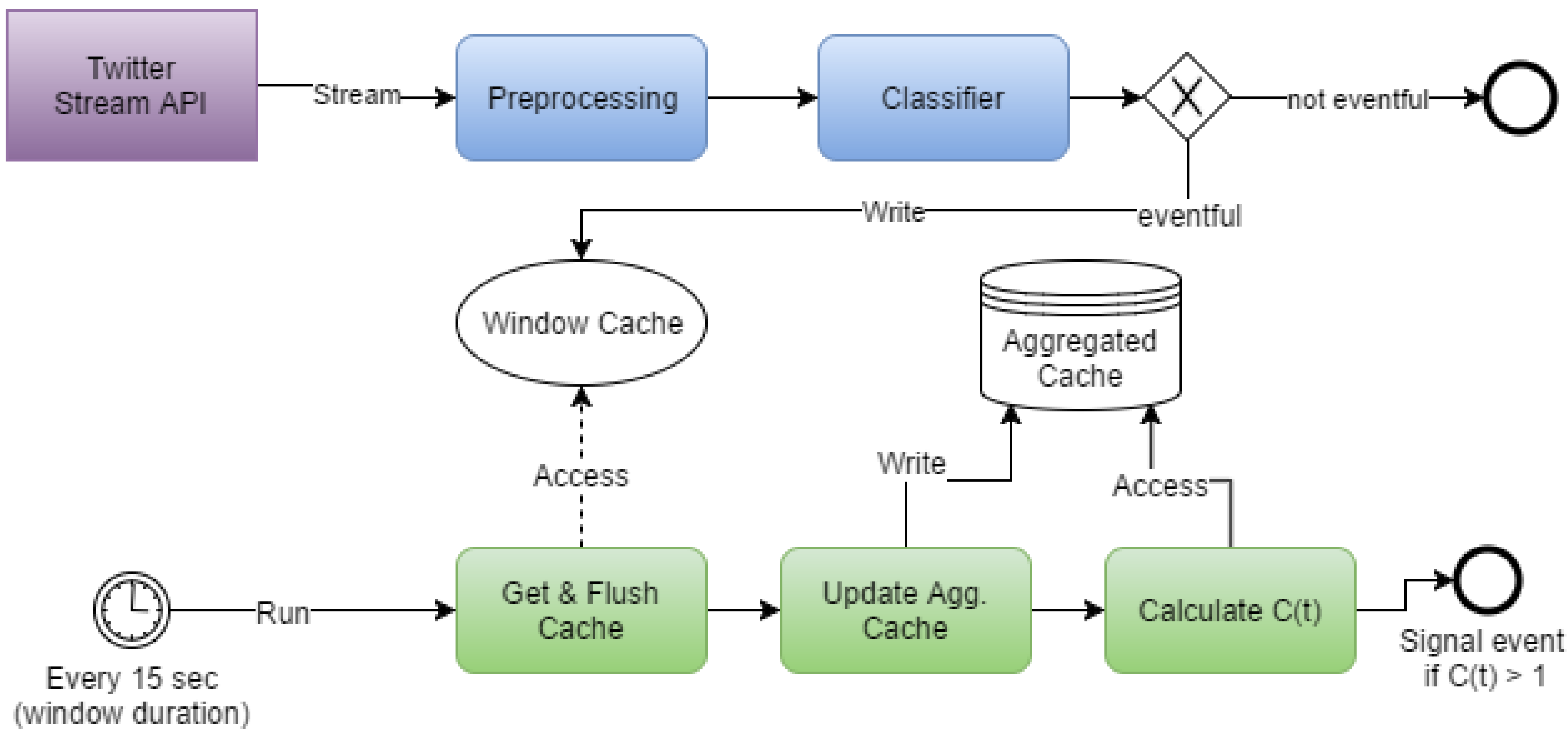


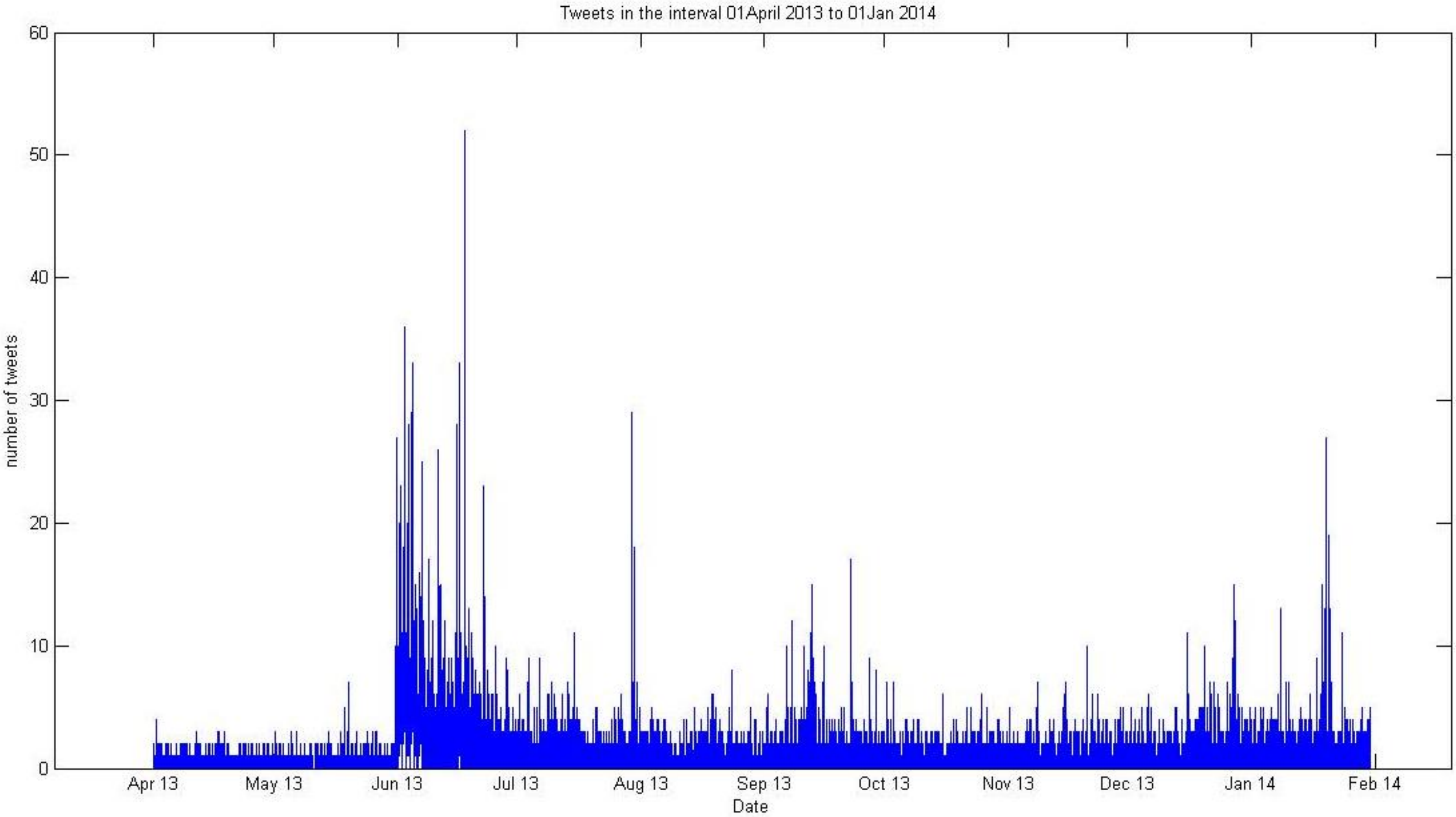
Figure 5: Novel Online Algorithm Design

Results

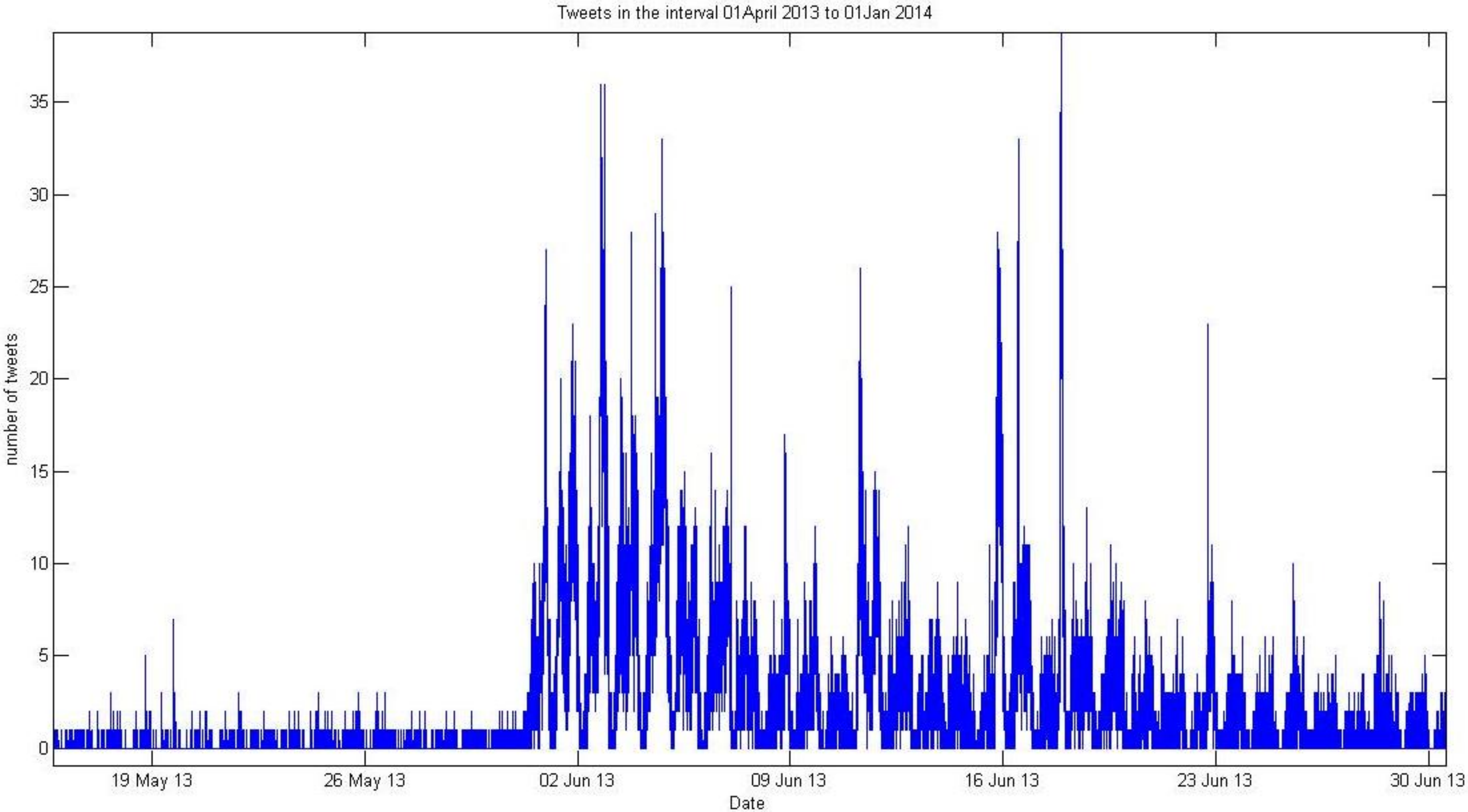
- Search interval: 01.04.2013 – 01.01.2014
  - (Comparatively) hot interval for Turkey
- Processed 25.710.914 tweets, 91.085 of them were ‘eventful’
- Obtained histogram and calculated C(t) values

M	b	C(t) drop thr	Count
2	3	0.5	69
2	5	0.25	24
4	6	0.25	7

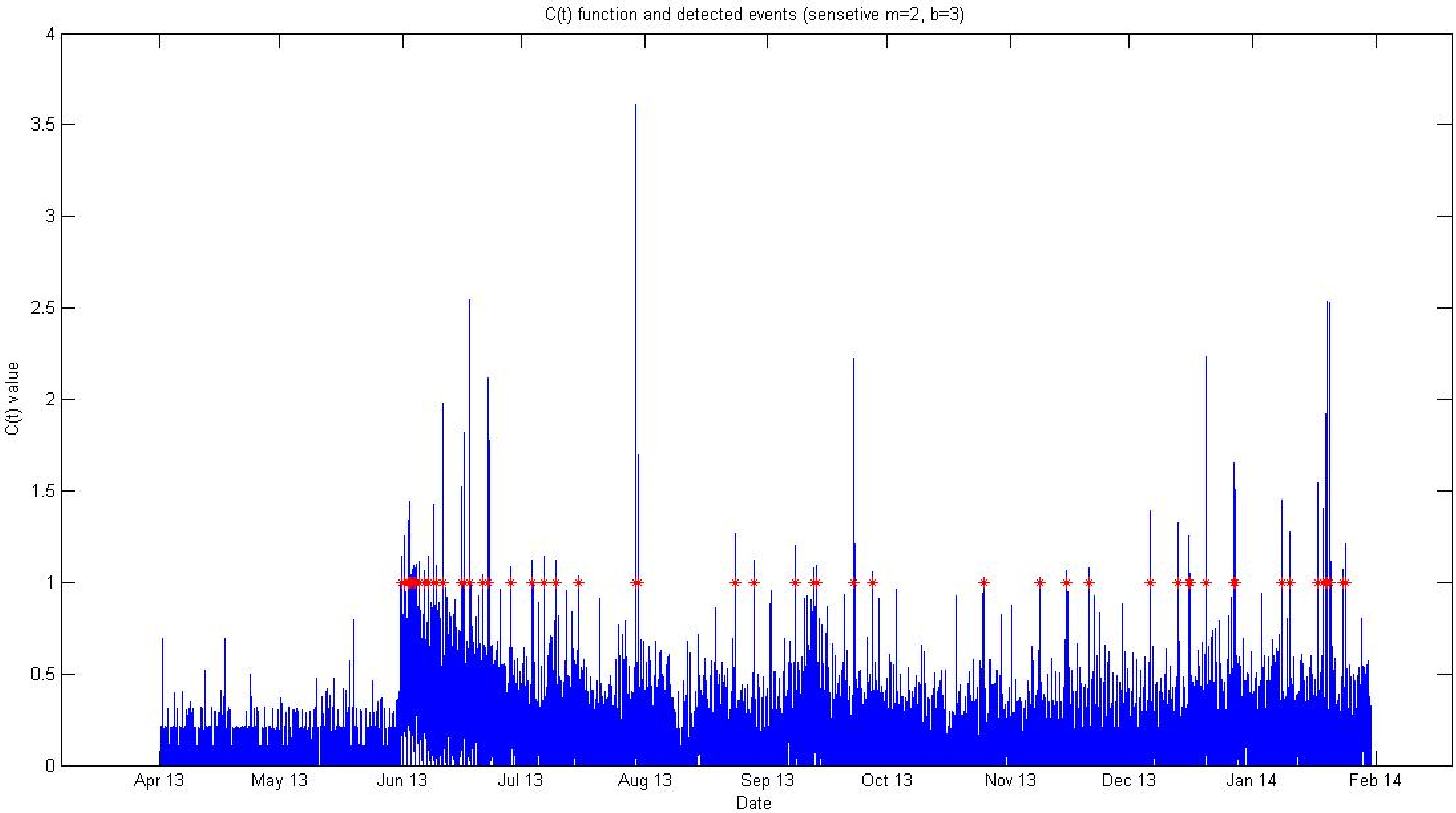
Histogram - Whole



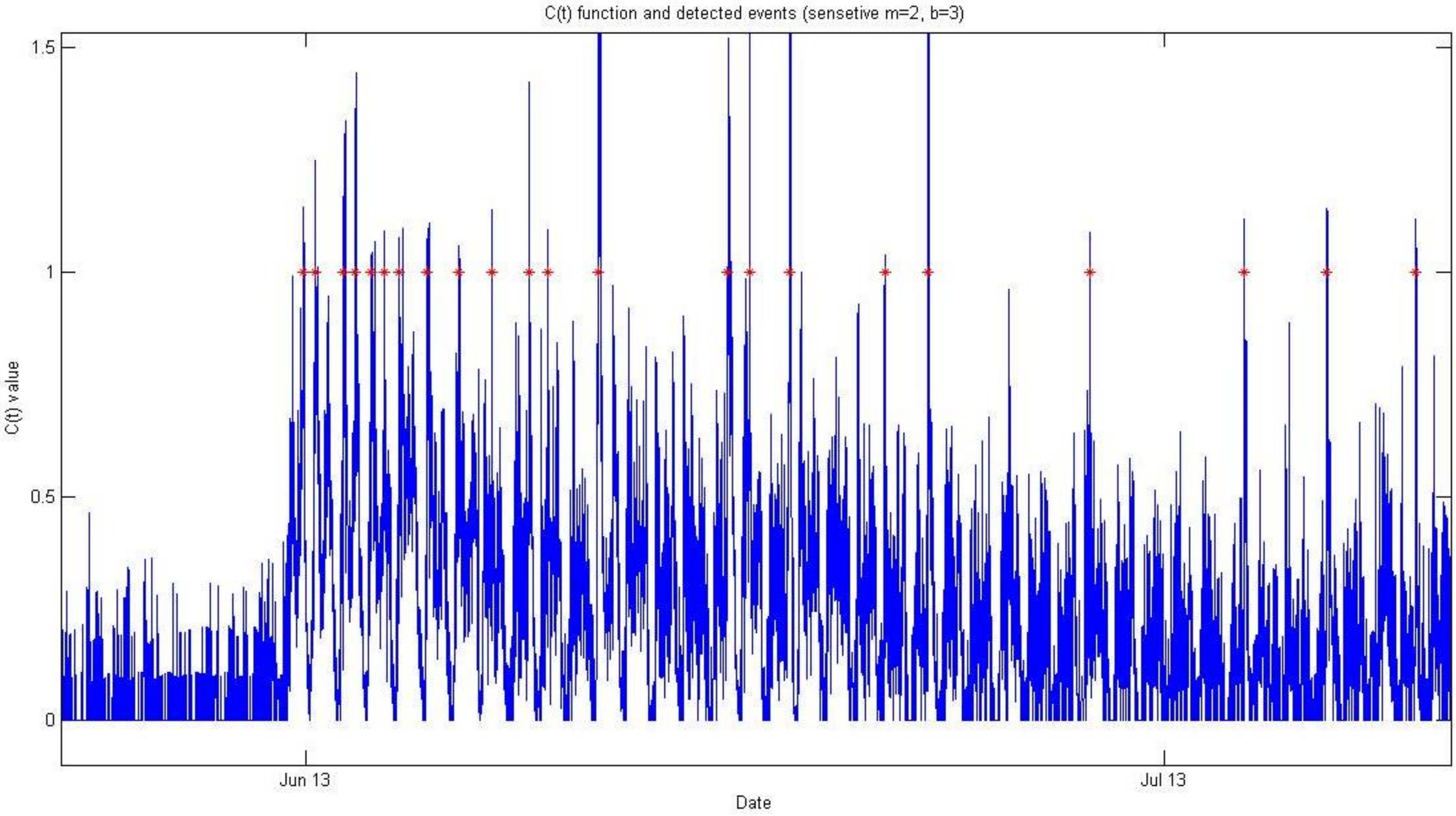
Histogram - Zoomed



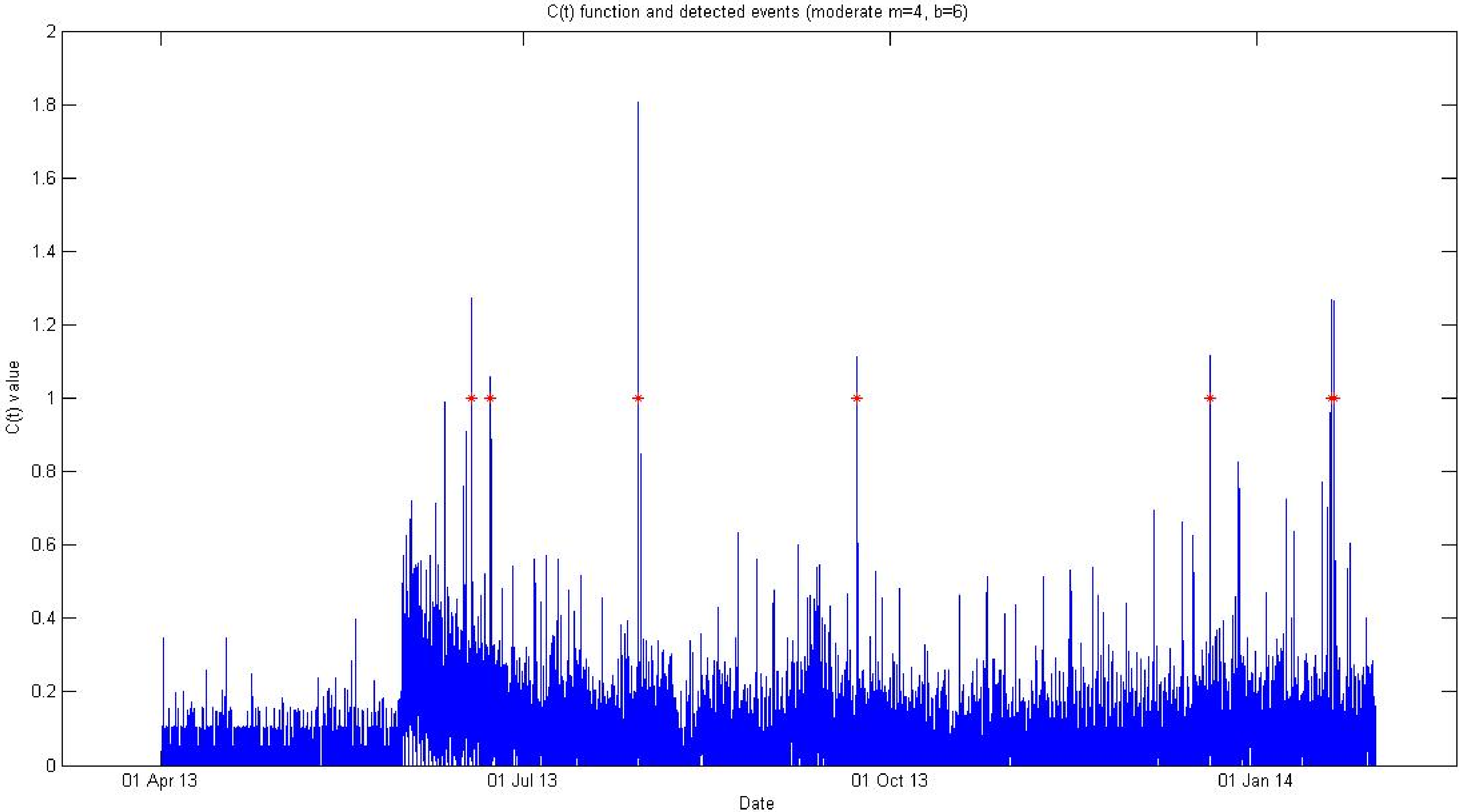
Sensitive Detector



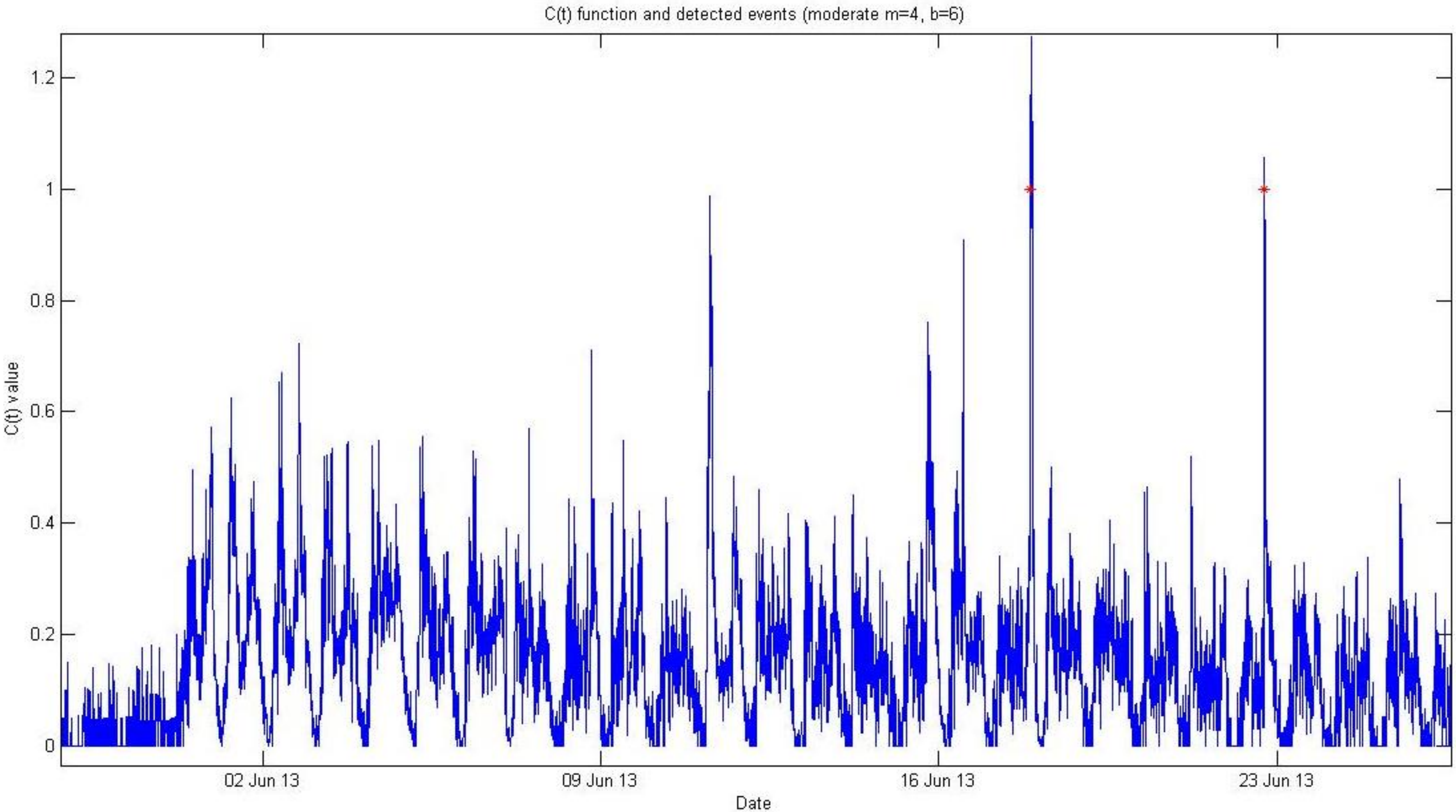
Sensitive Detector - Zoomed



Moderate Detector



Moderate Detector - Zoomed



Some tweets from a detected event on 2013-08-06 23:25:00 EEST

Id	Tweet Text
364845589659254784	sükrü saraçoglunda her yer taksim her yer direnis sloganlari
364845474114576385	her yer taksim her yer direnis sükru saraçoglu inliyor
364845494712803328	kadiköy de her yer taksim her yer direnis sesleri halkin takimiyiz
364845693296312320	sükrü saraçoglu nda her yer taksim her yer direnis sesleri helal olsun fenerbahçe salzburg kadiköy
364845710258077697	fenerbahçe maçinda bütün stad her yer taksim her yer direnis diye inliyor

Evaluation

- Precision is calculated by checking the correctness of the detected events
  - P = TP / TP + FP
  - 16 out of 24 reported events were correct!
  - P = 16 / 24 = 0.666
- Calculating Recall is a different task: Need a structured and reliable way to get list of events
  - Can be done by manually crawling some (reliable) news sites or by human annotators
  - Interesting topic for another project!

Conclusion

- This project proves that Event Detection via Twitter is possible and it can produce credible results
- Both offline and online algorithms can be employed
- More sophisticated filtering and tokenization methods are needed: Work from multiple independent projects can be merged (emotion detection, location estimation, topic classification)
- Finer tuning for m and b must be done to find optimal values with different STA and LTA values
- Working with a structured and credible list of tagged events would make it possible to calculate recall and make auto correlation between real events and detected ones

References

[1] Zemberek NLP
https://github.com/ahmetaa/zemberek-nlp

[2] P. Earle, D. Bowden and M. Guy, "Twitter earthquake detection: Earthquake monitoring in a social world", Annals of Geophysics, vol. 54, no. 6, pp. 708-715, 2011