Review of MERLIN: Parameter-Free Discovery of Arbitrary Length Anomalies in Massive Time Series Archives

Adib Habbou, Teddy Alexandre

ENS Paris-Saclay

Context

- **Anomaly Detection :** significant research topic in time series analysis
- **Wide range of applications :** healthcare, finance, industry, robotics...
- **Discord :** simple way of detecting anomalies by finding subsequences that are far from their nearest neighbours, requires one parameter : the subsequence length
- **Problem :** selecting the optimal subsequence length in order to detect anomalies of various length without relying on specific user defined parameters series
- **MERLIN :** based on DRAG is able to select the optimal subsequence length

Definitions

- **Time Series :** sequence of n real values : $T = t_1, ..., t_n$
- **Subsequence :** subset of values of length L starting at i : $T_{i,L} = t_i, ..., t_{i+L-1}$
- **Non-Self Match :** $M = T_{q,L}$ is a non-self match to $C = T_{p,L}$ if $|p - q| \geq L$
- **Discord :** D is a discord if it has the largest distance to it's nearest non-self match

  $\forall$ subsequence C of T, non-self match M_D of D and non-self match M_C of C :

$$min(Dist(D, M_D)) > min(Dist(C, M_C))$$

Candidate Selection (DRAG)

- **Objective :** identification of potential discord candidates in a time series
- **Parameters :** takes time series T, subsequence length L, and range of discords r
- **Algorithm :** parses subsequences, if their distance to nearest non-self match surpasses r, they're considered as potential discords ; shorter ones are rejected
- **Threshold Role :** r acts as a discriminator, admitting potential discords
- **Output :** set of candidates C may contain true discords and possible false positives ; algorithm proceeds to phase two if C is not empty

Discord Refinement (DRAG)

- **Objective :** refine discord candidates using the prior set of candidates C
- **Parameters :** same as candidate selection plus the set of candidates C
- **Iteration Process :** compares candidates against the true set of discords D
- **Discord Identification :** consider each subsequence distance to every member of C doing a best-so-far search for each candidate's nearest neighbour
- **Guaranteed Discord :** the algorithm ensures discovery of at least one discord
- **Output :** set of discords D with their index and distance to nearest non-self match

MERLIN

- **DRAG Difficulty :** determining parameter r poses the primary challenge
- **Estimation Strategy :** r should be *"a little less"* than the discord distance, so the estimation will be guided by the variance in the last few discord values
- **Objective :** adjusts r to optimize DRAG by introducing *MinL* and *MaxL*
- **Parameters :** takes time series and subsequence length lower and upper bounds
- **MERLIN Phases :**
  1. Finds first discord by decrementing r until discovery (initialized at $2\sqrt{MinL}$)
  2. Locates subsequent discords by iteratively adjusting r such as $r = 0.99 \times r$
  3. Identifies remaining discords by iteratively adjusting r using $r = r - std(\text{distances})$

Existing Datasets

- **New York City Taxi Dataset :** extracted from Numenta Anomaly Benchmark, limited to 1400 samples from November 2018 with two anomalies : Daylight Saving Time and Thanksgiving (at the beginning and the end of the month)
- **Dataset Characteristics :**
  - Exhibits weekly periodicity but lacks a dominant frequency implying no seasonality
  - DFT indicates peaks at the start and end, suggesting potential traffic shifts
- **Yahoo Database Dataset :** based on production traffic, comprised of real and synthetic time-series with tagged anomaly points from Yahoo services
- **Dataset Characteristics :**
  - Synthetic time series vary in trend, noise, and seasonality
  - Box plot shows extreme values significantly distant from the median

Generated Datasets

- **Toy Examples :** used for initial DRAG and MERLIN tests
  1. Constant time serie of length 1000 with one anomaly of length 50
  2. Constant time serie of length 1000 with two anomalies of lengths 20 and 50
  3. Sinusoidal time series of length 1000 with one anomaly of size 50
- **Ultra Subtle Anomalies :** sinusoidal time series of length 500 with slight signal modifications of lengths 2 to 5 at specific positions
- **Twin Freak :** time series of length 1000 with two similar anomalies of length 50 introduced using polynomial functions in specific intervals with similar coefficients

Toy Examples

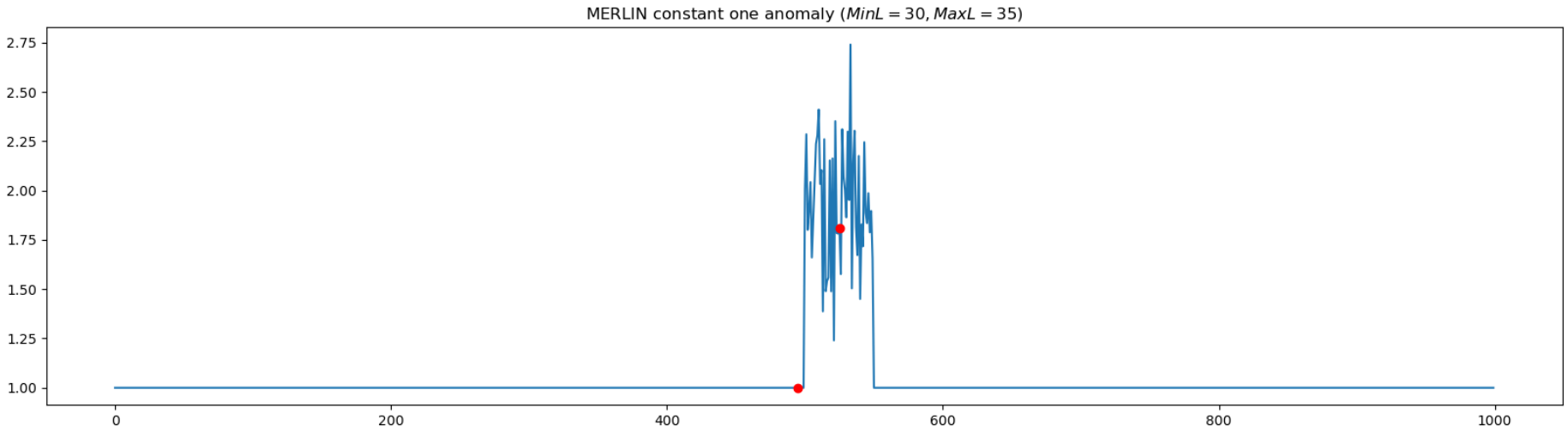

Figure – MERLIN on constant toy example with one discord

Observation

MERLIN detects accurately the discords (one falls in, one is detected slightly before).

Toy Examples

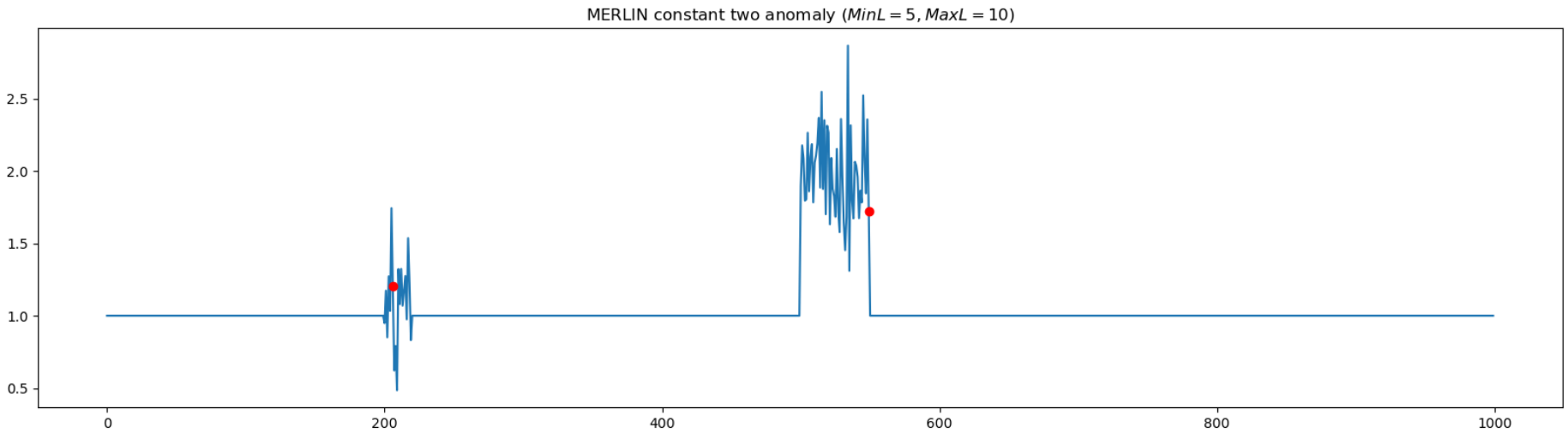

Figure – MERLIN on constant toy example with two discords

Observation

MERLIN detects accurately the discords despite them being of different lengths.

Toy Examples

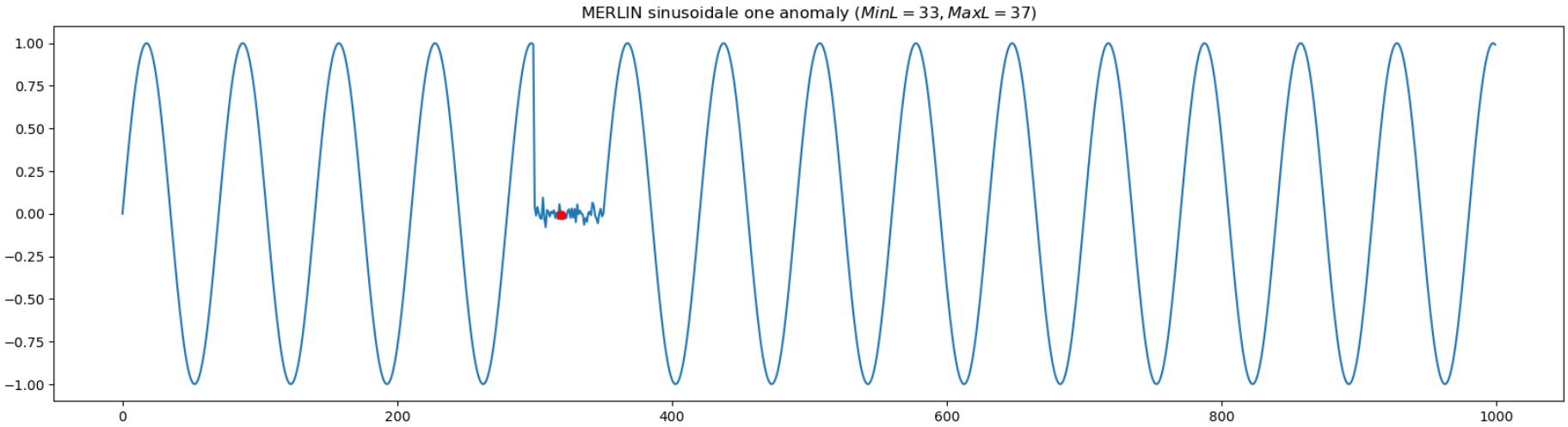

Figure – MERLIN on sinusoidal signal with discords

Observation

MERLIN detects the accurately discord, as in the experiment made by the authors.

New York City Dataset

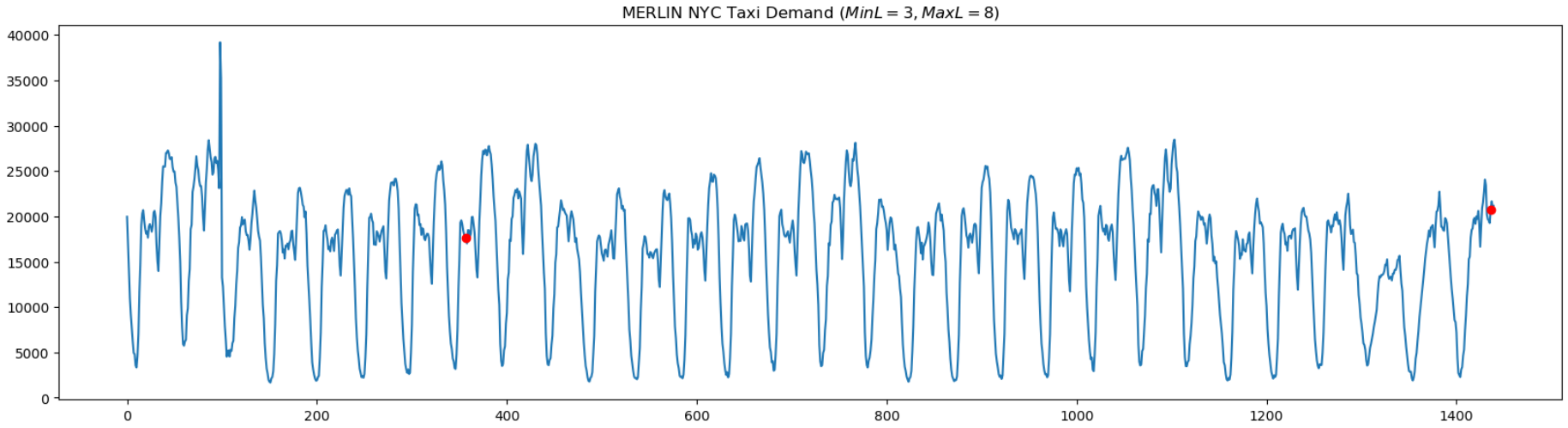

Figure – MERLIN on NYC Taxi Dataset

Observation

MERLIN detects discords that does not match those specified in the benchmark.

New York City Dataset

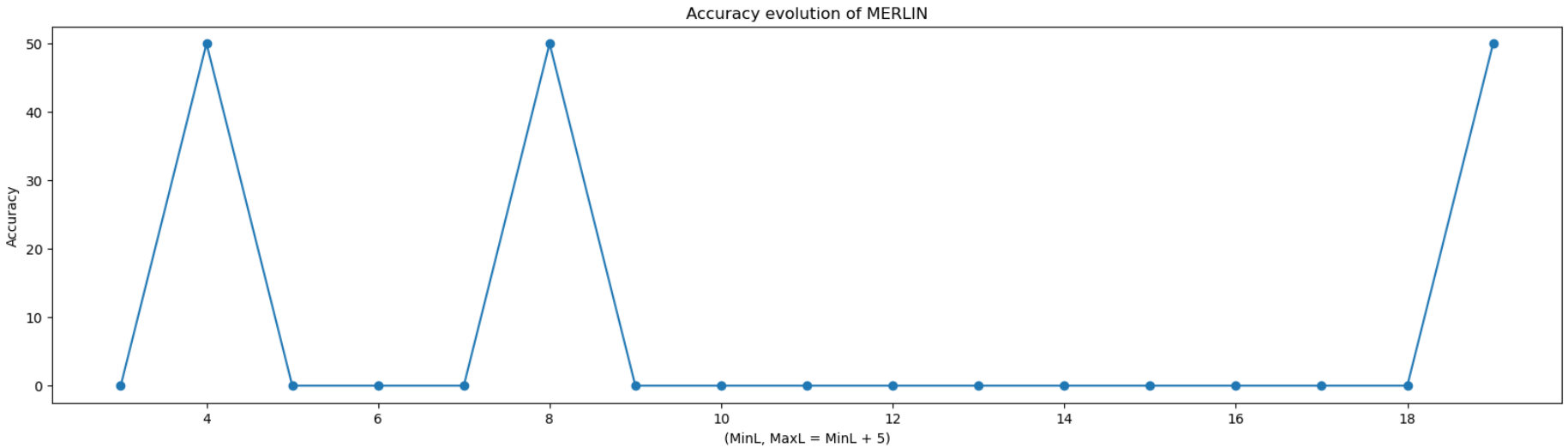

Figure – Accuracy on NYC-Taxi Dataset for different values of MinL/MaxL

Observation

MERLIN either detects one anomaly or none of them. The execution time for the whole range of is quite huge (more than hour). Tn the article they may have "cherry-picked" values of *MinL* and *MaxL* in order to detect both anomalies.

Yahoo Dataset

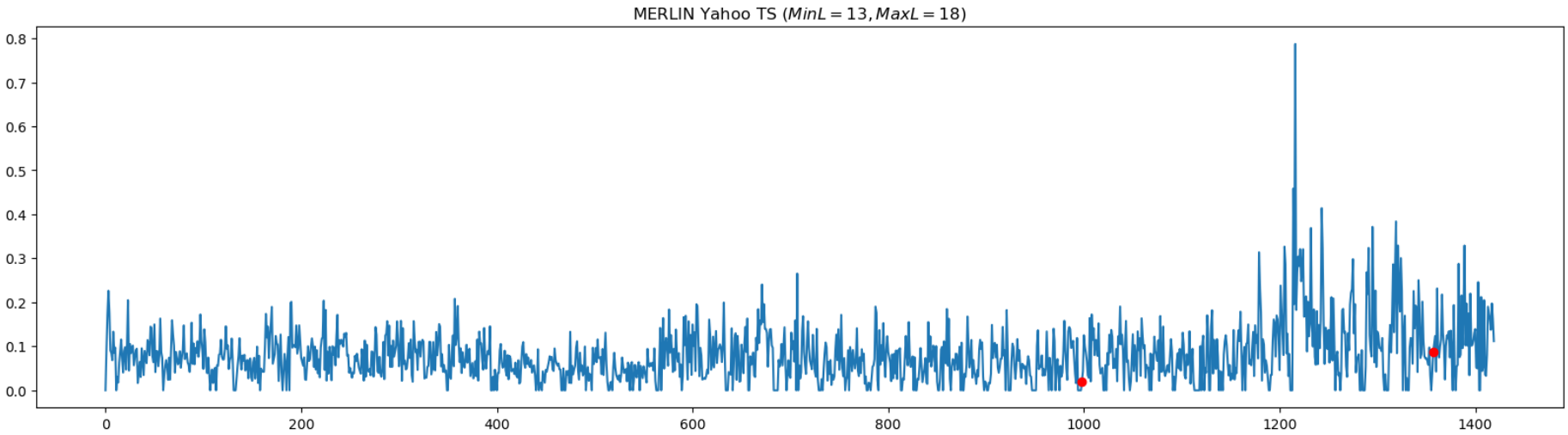

Figure – MERLIN on Yahoo ! Dataset

Observation

MERLIN detects discords that are not those that can be seen by a human eye, for example the peak around position 1200 representing the 50th day.

Yahoo Dataset

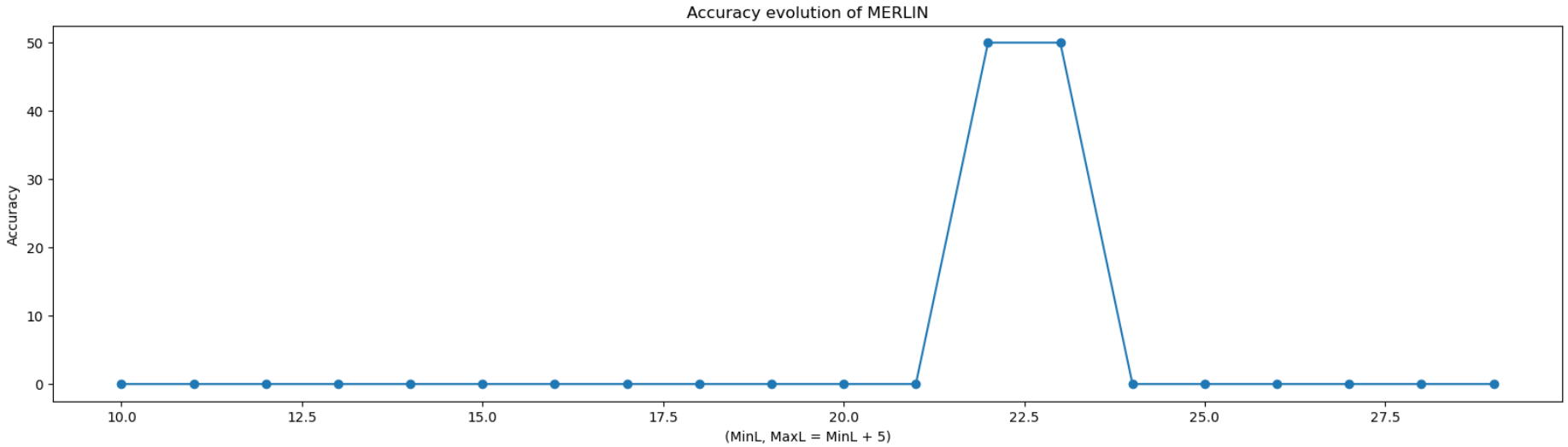

Figure – Accuracy on Yahoo ! Dataset for different values of MinL/MaxL

Observation

Again, poor accuracies in general, performance of the algorithm seems very dependent of the choice of *MinL* and *MaxL*, but no specific frame-work was developed for that.

Ultra Subtle

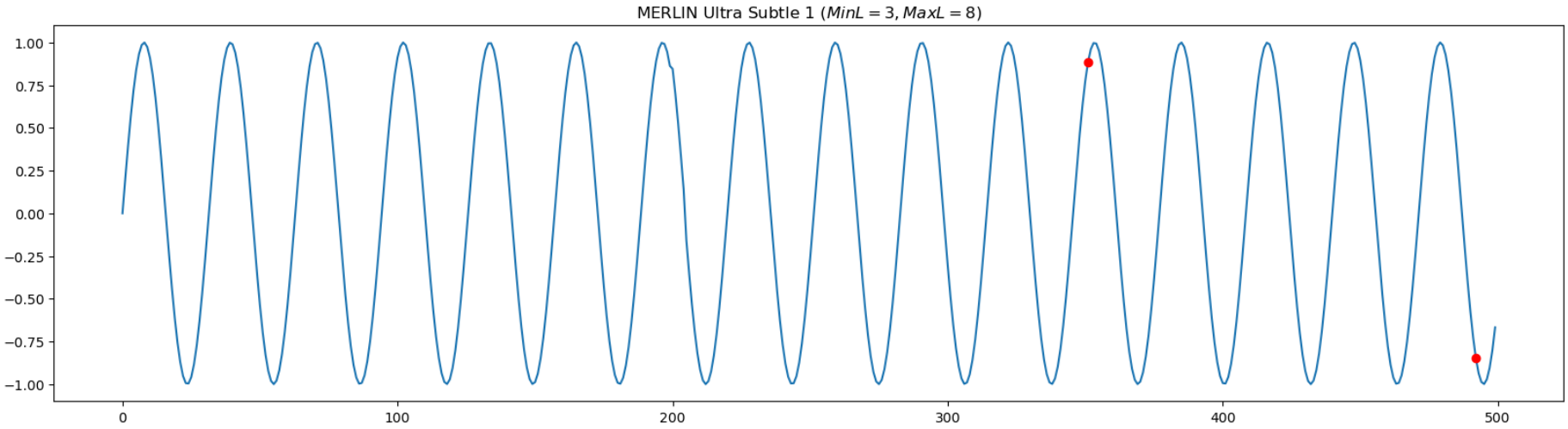

Figure – MERLIN on Ultra-subtle discord in a signal

Observation

MERLIN does not detect the true discord but he points out two false positive ones.

Twin Freak

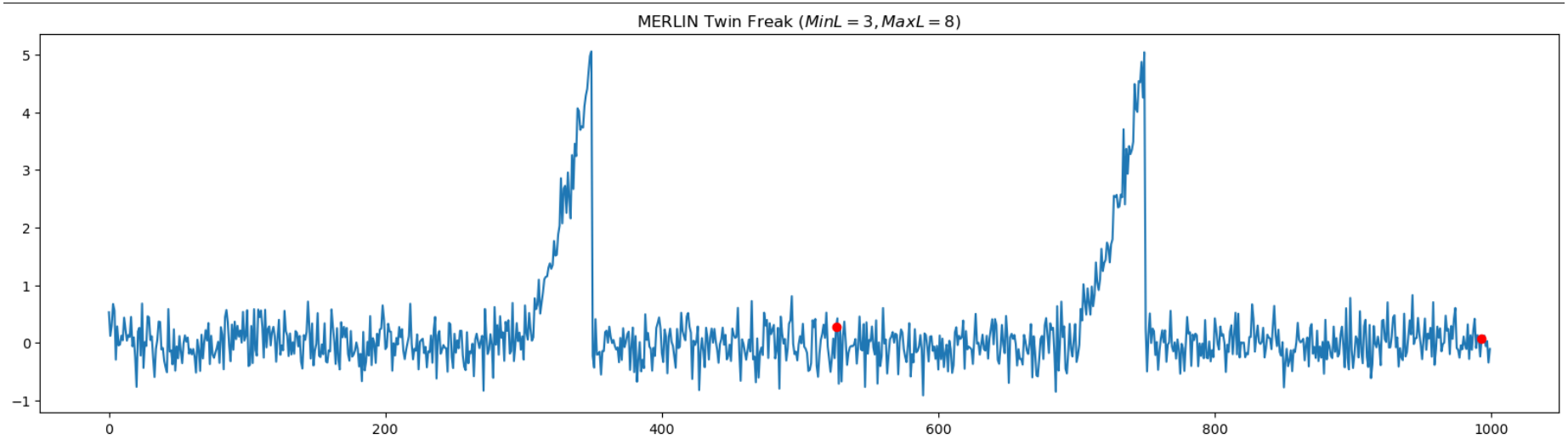

Figure – MERLIN on a Twin Freak constant signal

Observation

MERLIN does not detect any of the two discords here, as expected by the authors.

Extensions

- **Optimizing Execution Time :**
  - **MERLIN++ :** utilized Orchard indexing technique for efficient discord identification
  - **PALMAD :** introduced GPU-based parallelization to enhance MERLIN efficiency
- **Choice of MinL and MaxL :**
  - Seeking faster parameter selection methods than classical grid search
  - Seeking more generalizable parameter selection than using domain knowledge
  - Dynamic parameter adaptation based on discovered discords lengths
  - Dynamic parameter adaptation based on time series events such as spikes

(PALMAD = Parallel Arbitrary Length MERLIN-based Anomaly Discovery)