A Benchmark for Early Time-Series Classification

Petro - Foti Kamberi, Evgenios Kladis, Charilaos Akasiadis

Institute of Informatics & Telecommunications
NCSR Demokritos
Early Time-Series Classification

- Train on labeled full-length time-series data.
- Predict class labels of unseen time-series data by observing only their prefixes.
- Balance trade-off between:
  - Accuracy / F1-Score: Predictive Performance,
  - Earliness: Required prefix length (Lower better).
- Example of application domains (among others):
  Life-sciences, Drug Discovery, Maritime, Energy, etc.
Early Time-Series Classification

- Train on labeled full-length time-series data.
- Predict class labels of unseen time-series data by observing only their prefixes.
- Balance trade-off between:
  → Accuracy / F1-Score: Predictive Performance,
  → Earliness: Required prefix length (Lower better).
- Example of application domains (among others):
  Life-sciences, Drug Discovery, Maritime, Energy, etc.

Interesting:

Not interesting:
Early Time-Series Classification

- Train on labeled full-length time-series data.
- Predict class labels of unseen time-series data by observing only their prefixes.
- Balance trade-off between:
  - Accuracy / F1-Score: Predictive Performance,
  - Earliness: Required prefix length (Lower better).
- Example of application domains (among others):
  Life-sciences, Drug Discovery, Maritime, Energy, etc.

Interesting:

Not interesting:
Early Time-Series Classification

- Train on labeled full-length time-series data.
- Predict class labels of unseen time-series data by observing only their prefixes.
- Balance trade-off between:
  - Accuracy / F1-Score: Predictive Performance,
  - Earliness: Required prefix length (Lower better).
- Example of application domains (among others):
  Life-sciences, Drug Discovery, Maritime, Energy, etc.
Early Time-Series Classification

- Train on labeled full-length time-series data.
- Predict class labels of unseen time-series data by observing only their prefixes.
- Balance trade-off between:
  → **Accuracy / F1-Score**: Predictive Performance,
  → **Earliness**: Required prefix length (Lower better).
- Example of application domains (among others):
  Life-sciences, Drug Discovery, Maritime, Energy, etc.
The need for an ETSC Benchmark

- Lack of concise evaluation of ETSC methods in the literature:
  - Evaluation and comparison against a limited set of alternative algorithms.

- Selection of Datasets meaningful for ETSC, which meet the online operation requirements:
  - Time interval between consecutive measurements > Prediction Time,
  - No Z-normalization \[^{[1]}\],
  - Temporal dimension.

[^1]\ R. Wu, A. Der, E. Keogh, When is early classification of time series meaningful, IEEE Transactions on Knowledge and Data Engineering (2021) 1–1, doi:10.1109/TKDE.2021.3108580.
Key Features of this Work

- Open-Source and extensible framework for ETSC evaluation.
- Incorporation of 5 existing algorithms + 1 developed by our team (and some variations).
- Evaluation on 12 meaningful for ETSC datasets (10 from UEA & UCR repository + 2 novel datasets).
- Results and Analysis from an Empirical Comparison.
Effective Confidence-based Early Classification (ECEC)\textsuperscript{[2]}:

- Train \( N \) probabilistic classifiers (on \( N \) prefixes), and compute the posterior probability of each being correct.
- Rank classifiers and determine confidence thresholds.
- Accept a prediction if its confidence surpasses the threshold.
- Cubic complexity in terms of time-series length and linear in terms of dataset size.

\[\begin{array}{cccccccc}
1137 & 1229 & 1213 & 1091 & 896 & 744 & 681 & 661 & 656 & 855
\end{array}\]

Incorporated Existing Algorithms - ECEC

Incorporated Existing Algorithms - ECO-K

Economy-K (ECO-K)\(^3\):

- Clustering of full time-series.
- A Classifier for each time-point \( t \).
- New observations are assigned to each cluster with a membership probability.
- Generate prediction at the time-point with the minimum \( f_\tau \):
  \[
  f_\tau(t_{s_t}) = \sum_{c_k} P(c_k \mid t_{s_t}) \sum_y \sum_{\hat{y}} P_{t+\tau}(\hat{y} \mid y, c_k)(\hat{y} \mid y) + C(t + \tau)
  \]
- Linear complexity in terms of both time-series length and dataset size.

<table>
<thead>
<tr>
<th>ts(_1)</th>
<th>ts(_2)</th>
<th>ts(_3)</th>
<th>ts(_4)</th>
<th>ts(_5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1137</td>
<td>1229</td>
<td>1213</td>
<td>1091</td>
<td>896</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Membership Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster(_1)</td>
<td>0.6</td>
</tr>
<tr>
<td>cluster(_2)</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Incorporated Existing Algorithms - ECTS

Early Classification of Time Series (ECTS)\cite{4}:

**Training:**
- Find 1-NN sets for all prefixes and all time-series.
- Determine Reverse NN sets.
- Check from which time-point onward the RNN sets remain unchanged (Minimum prediction length - MPL).
- Group into clusters based on Euclidean distance.
- 1-NN and RNN sets should belong to the same cluster.
- Linear complexity in terms of time-series length and cubic in terms of dataset size.

**Testing:**
- Match new instances with their NN.
- If testing time-series length is larger than NN’s MPL, then generate a prediction.

<table>
<thead>
<tr>
<th>Time series</th>
<th>MPL (NN)</th>
<th>MPL (Clustering)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ts\textsubscript{1}</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>ts\textsubscript{2}</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>ts\textsubscript{3}</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>ts\textsubscript{4}</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>ts\textsubscript{5}</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

\cite{4} Z. Xing, J. Pei, P. S. Yu, Early classification on time series, Knowledge and information systems 31 (1) (2012) 105–127.
Incorporated Existing Algorithms - EDSC

Early Distinctive Shapelet Classification (EDSC)\cite{5}:

- One of the first ETSC methods.
- User defines a range of subseries lengths.
- Create shapelets that are maximally representative of a class (\textit{subseries, threshold, class label}).
- Threshold: minimum distance that another time-series should have to assign to the same class label \(\rightarrow\) find minimum distances of time-series with different class labels.
- Determine most discriminative shapelets for each class by checking their utility (F1-score with weighted Recall).
- Cubic complexity in terms of time-series length and quadratic complexity in terms of dataset size.

Truncate the dataset to $N$ overlapping prefixes.

For each prefix train a WEASEL-Logistic Regression pair.

For each prefix length train an One Class SVM that accepts correct predictions and rejects false ones.

Check for consecutive and consistent predictions before terminating.

Quadratic complexity in terms of time-series length.

Incorporated Existing Algorithms - TEASER

Two-tier Early and Accurate Series Classifier (TEASER)\(^6\):

\[1137 \quad 1229 \quad 1213 \quad 1091 \quad 896 \quad \cdots \rightarrow \text{Weasel} \rightarrow \text{OneClass SVM} \]

\[\begin{align*}
\text{Weasel} & \rightarrow \text{OneClass SVM} \\
\text{No} & \rightarrow v \leq \text{consistency} \\
\text{No} & \rightarrow \text{Prediction} \\
\text{Yes} & \\
\text{Yes} & \\
\end{align*}\]
A new, simpler benchmark method: - STRUT

Selective Truncation of Time-series (STRUT):

- Split the training dataset to all possible prefixes.
- Train and validate full time-series classification algorithms (Minirocket\textsuperscript{[7]}, MLSTM\textsuperscript{[8]}, WEASEL\textsuperscript{[9]}).
- Select minimum $t$ where a user-defined metric $S(.)$ is optimized

$$STRUT(X_{test}) = \min(\arg \max S(h^t(X_{test}^{1:t} | D_{train})))$$

- Faster Approximation Variant: Decide which time-points $t$ to evaluate using bisection (essentially an iterative binary search approach).
- Linear complexity in terms of dataset size. For S-WEASEL quadratic complexity in terms of time-series length and linear for S-MINI. The complexity of S-MLSTM is affected by the inherent complexity of LSTMs.
Datasets

- Ten meaningful for ETSC datasets from the UCR/UEA repository.
- Two new datasets:
  1. Life-sciences domain: large-scale simulations of drug treatments for cancer,
  2. Maritime domain, vessel AIS messages.
- Categorization according to characteristics.

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th>No. of Variables</th>
<th>No. of Instances (Height)</th>
<th>No. of Time-points (Length)</th>
<th>No. of Classes</th>
<th>Class Imbalance Ratio</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>BasicMotions</td>
<td>6</td>
<td>80</td>
<td>100</td>
<td>4</td>
<td>1</td>
<td>3.02</td>
</tr>
<tr>
<td>Biological</td>
<td>3</td>
<td>644</td>
<td>49</td>
<td>2</td>
<td>5.37</td>
<td>222.93</td>
</tr>
<tr>
<td>DodgerLoopDay</td>
<td>1</td>
<td>158</td>
<td>288</td>
<td>7</td>
<td>1.25</td>
<td>12.77</td>
</tr>
<tr>
<td>DodgerLoopGame</td>
<td>1</td>
<td>158</td>
<td>288</td>
<td>2</td>
<td>1.07</td>
<td>12.77</td>
</tr>
<tr>
<td>DodgerLoopWeekend</td>
<td>1</td>
<td>158</td>
<td>288</td>
<td>2</td>
<td>2.43</td>
<td>12.77</td>
</tr>
<tr>
<td>HouseTwenty</td>
<td>1</td>
<td>159</td>
<td>2,000</td>
<td>2</td>
<td>1.27</td>
<td>779.19</td>
</tr>
<tr>
<td>LSST</td>
<td>6</td>
<td>4,925</td>
<td>36</td>
<td>14</td>
<td>111</td>
<td>167.26</td>
</tr>
<tr>
<td>Maritime</td>
<td>7</td>
<td>80,591</td>
<td>30</td>
<td>2</td>
<td>4.21</td>
<td>37.98</td>
</tr>
<tr>
<td>PickupGestureWiimoteZ</td>
<td>1</td>
<td>100</td>
<td>361</td>
<td>10</td>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>PLAID</td>
<td>1</td>
<td>1,074</td>
<td>1,345</td>
<td>11</td>
<td>6.73</td>
<td>3.41</td>
</tr>
<tr>
<td>PowerCons</td>
<td>1</td>
<td>360</td>
<td>144</td>
<td>2</td>
<td>1</td>
<td>0.9</td>
</tr>
<tr>
<td>SharePriceIncrease</td>
<td>1</td>
<td>1,931</td>
<td>60</td>
<td>2</td>
<td>2.19</td>
<td>1.64</td>
</tr>
</tbody>
</table>

Group Specifications

<table>
<thead>
<tr>
<th>Group</th>
<th>Specifications</th>
<th>Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wide</td>
<td>Length &gt; 1,300</td>
<td>HouseTwenty, PLAID</td>
</tr>
<tr>
<td>Large</td>
<td>Height &gt; 1,000</td>
<td>LSST, Maritime, PLAID, SharePriceIncrease</td>
</tr>
<tr>
<td>Unstable</td>
<td>Coef. of Var. &gt; 1.08</td>
<td>BasicMotions, Biological, HouseTwenty, LSST, PLAID, SharePriceIncrease</td>
</tr>
<tr>
<td>Imbalanced</td>
<td>Class Imbalance Ratio &gt; 1.73</td>
<td>Biological, DodgerLoopWeekend, LSST, Maritime, PLAID, SharePriceIncrease</td>
</tr>
<tr>
<td>Multiclass</td>
<td>Number of Classes &gt; 2</td>
<td>BasicMotions, DodgerLoopDay, LSST, PickupGestureWiimoteZ, PLAID</td>
</tr>
<tr>
<td>Common</td>
<td>None of the above</td>
<td>BasicMotions, DodgerLoopGame, DodgerLoopWeekend, PickupGestureWiimoteZ, PowerCons</td>
</tr>
<tr>
<td>Univariate</td>
<td>One variable per instance</td>
<td>DodgerLoopDay, DodgerLoopGame, DodgerLoopWeekend, HouseTwenty, PickupGestureWiimoteZ, PLAID, PowerCons, SharePriceIncrease</td>
</tr>
<tr>
<td>Multivariate</td>
<td>More than one variable per instance</td>
<td>BasicMotions, Biological, LSST, Maritime</td>
</tr>
</tbody>
</table>

Top: Tested Datasets’ Characteristics, Bottom: Dataset Groups and Categorization Criteria
The Framework

- Main language: Python 3.7
  - ECEC and TEASER implemented in Java
  - EDSC in C++

- Performance Metrics:
  - Accuracy
  - F1-Score
  - Earliness
  - Harmonic Mean of Accuracy and Earliness
  - Training and Testing times

- CLI with various options that allows to specify, e.g.:
  - Input files format (.arff and .csv)
  - Number of variables in the multivariate cases
  - Method for applying univariate algorithms to multivariate datasets (e.g. voting)
  - Target class for F1-Score calculation
  - Option for cross-validation

Public GitHub Repository
https://github.com/xarakas/ETSC
Adding New Algorithms:

- Create a Python interface implementing the abstract class EarlyClassifier.
- Create a python script that implements the train and predict methods.
- Extend the file CLI to:
  (a) Import the new algorithm implementation,
  (b) Define input parameters, execution options, and algorithm invocation.
- Note: Algorithms can be implemented in any language, provided a Python wrapper is available.

Enriching Datasets

- Datasets should be in .csv file format.
- Each row should represent a time-series instance and the first value should represent the assigned class label.
(i) Wide
(ii) Large
(iii) Unstable
(iv) Imbalanced
(v) Multiclass
(vi) Common
(vii) Univariate
(viii) Multivariate
Key Takeaways

- Introduction of an **open-source framework** for evaluating and objectively comparing ETSC algorithms.

- The **framework** is highly **extensible**, allowing for the seamless incorporation of new **algorithms** and the enrichment with new **datasets** (that should be meaningful for ETSC).

- **Benchmark Method - STRUT**: We present a benchmark method called STRUT, which serves as an essential baseline for the monitoring of the ETSC performance without a dynamic component to declare earliness.
Ongoing and Future Work

We are currently extending the framework’s algorithms with:

- **T-SMOTE**[^10]: An algorithm that focuses on addressing class imbalance in time-series data to improve classification.
- **SDRE**[^11]: Incorporating Spectral Density Ratio Estimation for more robust ETSC solutions.
- **MOO-ETSC**[^12]: Multi-Objective Optimization in ETSC, exploring diverse performance criteria.

In the future, we plan to incorporate Automated Machine Learning (AutoML[^13]) tools for optimizing hyperparameters in ETSC algorithms.


Thank you!

Contact: cakasiadis@iit.demokritos.gr
         pkamperi@iit.demokritos.gr

More info: https://cer.iit.demokritos.gr/

Comments/Questions?