Towards Time-Evolving Analytics: Online Learning for Time-Dependent Evolving Data Streams

Emanuele Della Valle [a,*], Giacomo Ziffer [a], Alessio Bernardo [a], Vitor Cerqueira [b] and Albert Bifet [c]

[a] *DEIB, Politecnico di Milano, Milano, Italy*
E-mail: emanuele.dellavalle@polimi.it; ORCID: https://orcid.org/0000-0002-5176-5885; E-mail: giacomo.ziffer@polimi.it; ORCID: https://orcid.org/0000-0002-2768-3580; E-mail: alessio.bernardo@polimi.it; ORCID: https://orcid.org/0000-0002-3492-0345

[b] *Dalhousie University, Halifax, Canada*
E-mail: vitor.cerqueira@dal.ca; ORCID: https://orcid.org/0000-0002-9694-8423

[c] *University of Waikato, Hamilton, New Zealand*
E-mail: abifet@waikato.ac.nz; ORCID: https://orcid.org/0000-0002-8339-7773

Abstract. Traditional historical data analytics is at risk in a world where volatility, uncertainty, complexity, and ambiguity are the new normal. While Streaming Machine Learning (SML) and Time-series Analytics (TSA) attack some aspects of the problem, we are far from a comprehensive solution. SML trains models using fewer data and in a continuous/adaptive way relaxing the assumption that data points are identically distributed. TSA considers temporal dependence among data points, but it assumes identical distribution. Every Data Scientist fights this battle with ad-hoc solutions. In this paper, we claim that, due to the temporal dependence on the data, the existing solutions do not represent robust solutions to efficiently and automatically keep models relevant even when changes occur, and real-time processing is a must. We propose a novel and solid scientific foundation for Time-Evolving Analytics in this perspective. Such a framework aims to develop the logical, methodological, and algorithmic foundations for fast, scalable, and resilient analytics.

Keywords: Time Evolving Analytics, Streaming Machine Learning, Time Series Analysis, Temporal Dependence, Concept Drift

1. Introduction

From drilling in an oil ring to managing traffic, from how doctors diagnose a disease to performing financial operations, the growing ability to collect, integrate, store, and analyze massive data fuels scientific breakthroughs and technological innovations. Recently, Machine Learning models exhibited or even surpassed human-level performance on individual tasks, e.g., Atari games [1] or object recognition [2]. Although these results are impressive, they are obtained from static models incapable of adapting their behaviour over time. As such, this requires restarting the training process each time new data becomes available (a.k.a., stateless retraining). In our dynamic world, this practice quickly becomes intractable for

*Corresponding author. E-mail: emanuele.dellavalle@polimi.it.

data streams or data, more generally, that may only be available temporarily due to storage constraints or privacy issues. Data streams call for novel stateful systems that adapt continuously and keep on learning over time. The urgency of this capacity is all the more remarkable if we consider that, despite the continuous advances in storage technology, already in 2025, the demand for storage will outstrip storage production by one order of magnitude. Seagate predicts that this exponential inflation will require analyzing almost 30% of global data in real-time [3].

Table 1

Difference among Offline Machine Learning, Time Series Analysis, Incremental Learning, Streaming Machine Learning, and Time-Evolving Analytics framework.

	i.i.d. dataset	i.i.d. data stream	time-dependent time series	evolving data stream	not i.i.d. data stream
Offline Machine Learning	✔	✘	✘	✘	✘
Incremental Learning	✔	✔	✘	✘	✘
Time Series Analysis	✔	✘	✔	✘	✘
Streaming Machine Learning	✔	✔	✘	✔	✘
Time-Evolving Analytics	✔	✔	✔	✔	✔

Moreover, in a growing number of markets where the demand for products was stable or even linear, the period in which the products are saleable is now short and seasonal. Finally, when changes hit, organizations that employ traditional stateless analytics techniques that rely heavily on historical data discover that their models are no longer relevant, resulting in a performance decrease. COVID-19 is one of those changes. As a consequence of the pandemic, many datasets became useless, causing several Pre-COVID-19 models to be no longer valid (src. Gartner [4] and MIT Tech Review [5]). Indeed, the typical assumption that data points are independent and identically distributed (i.i.d.) holds no longer. Historical data analytics, which counts on i.i.d., is at high risk.

A shared need is emerging for a novel type of Time-Evolving Analytics that:

R1. is not limited to a specific setting, i.e., it does not focus on a specific problem (e.g., only classification), but includes supervised, semi-supervised and unsupervised problems;

R2. makes high-order predictions[1], i.e., exploiting dependence in the sequences that spans multiple time steps and multiple scales from seconds/minutes to months/years;

R3. predicts *multiple possible future outcomes*., i.e., in a complex and uncertain world, the most probable outcome may be insufficient, and the ability to predict and evaluate the likelihood of each prediction becomes crucial;

R4. *continuously adapts to changes*, i.e., in a volatile world where data often have statistics that evolve, forgetting or ignoring past data become crucial;

R5. *learns statefully*, i.e., when changes are abrupt, the ability to detect patterns on the fly without the need to store all the data and pass through them multiple times is crucial; ideally, algorithms should learn from one data point at a time before discarding it; and

R6. does not structure the model a priori but allows the number and nature of the parameters to evolve over time.

As Table 1 shows, there are clear distinctions about each existing models' type of data. The difference between time series and data streams may be questioned. Indeed, time series may commonly arrive in

[1]The term order refers to Markov order.

the form of online data, and they thus can be treated as a data stream. Another way of seeing it: data streams may often involve temporal dependence and thus be considered as time series. Undoubtedly, the most general scenario is to handle not i.i.d. data stream. Offline Machine Learning deals only with independent and identically distributed data, not addressing neither R4 nor R5. Incremental learning does not adapt to changes (R4), despite addressing R5. Moreover, while Streaming Machine Learning (SML) and Time-series Analytics (TSA) attack some aspects of these challenges, we are far from a comprehensive solution. SML trains models using fewer data (R4) and in a continuous/adaptive and stateful way [6] (R5), i.e., relaxing the assumption that data points are identically distributed but still assuming that there is no temporal dependence. Hence, it does not address R2. On the contrary, TSA considers temporal dependence among data points [7] (R2), but it assumes identical distribution. Thus, it does not address R4. Indeed, every Data Scientist fights these challenges with ad-hoc solutions.

In particular, the contributions of this paper are the following:

- We provide empirical evidence using the well-known Electricity dataset [8] that the contemporaneous presence of changes in the data distribution and temporal dependences puts ML, SML, and TSA at risk of being ineffective. In particular, we introduce a new experimental methodology that extends previous analysis [9, 10] adding ML, analyzing models' accuracy over time, and tracking their resource usage.
- We envision a new framework (namely, Time-Evolving Analytics) that keeps models (R1) relevant even when changes occur (R4), and real-time processing is a must (R5).
- We explain why such a framework has the potential to pave the way for the development of a new generation of sequential models that embed high-order temporal dependence (R2) and predict multiple possible futures (R3) without a fixed model structure (R6).
- We indicate the characteristics of a unifying model for Time-Evolving Analytics that does not assume i.i.d. and a unifying methodology that can guide practitioners in systematically addressing the requirements listed above.

The remainder of this paper is organized as follows. Section 2 discusses the need for a new type of Time-Evolving Analytics, overcoming the currently existing limitations. In Section 3 we give an overview of the main works about the temporal dependence in time series and data streams. In Section 4, we detail the experiments, comparing ML, SML, and TSA models on the Electricity dataset showing the reason why we need a new analytics. Section 5 presents the main challenges and benefits that may arise with Time-Evolving Analytics. Section 6 outlines the desiderata of the envisioned model and methodology. Finally, Section 7 discusses the main takeaways regarding research opportunities and challenges explored throughout the paper.

2. Time for a principled solution

For Computer Science, the 2000s have seen the Big Data industry's come-of-age and the corresponding scientific field at the intersection of High-Performance-Computing, Databases, and Machine Learning. New technology concepts are formulated and quickly transferred into products. The scientific community struggles in keeping the industry's pace, and it is lingering considerably.

For instance, take the case of nowcasting. The term is a contraction of "now" and "forecasting ." It originates in meteorology, referring to the need for weather forecasts that are time and space specific for periods less than a few hours [11]. However, it is an active research field also in economics [12], epidemiology [13, 14], and energy [15], just to cite a few. These forecasts are particularly hard because the i.i.d. assumption hardly holds, multiple temporal dependencies are present, and changes (both in data distribution and the order of the temporal dependency) are frequent. As a result, past data are often of little practical interest (i.e., forgetting them is valid), but a method can be effective only by selectively remembering recurring patterns and seasonal effects. Moreover, even if the ideal forecast should be crisp, the uncertainty is so high that the only viable solution is to generate alternative predictions with an associated confidence level. Last but not least, nowcasting also means massive flows of observations that we had better process incrementally than in large batches.

We claim that the rapid and ad-hoc solutions appearing on the market[2] cannot provide long-term foundations and will fail to adapt models fast enough in the volatile, complex, and uncertain future that awaits us [3]. We argue that even more than for Big Data, academia has a crucial role in setting and developing the new course. Even if designed by phenomenal data engineers and data scientists, Industrial Solutions are prone to fail due to the complexity of the problem and the lack of basic principles. We, therefore, propose to develop the foundations of Time-Evolving Analytics.

Time-Evolving Analytics relies on formalizing a unifying framework that pivots from TSA and SML to a class of analytics operating with time-dependent data streams and online adaptive techniques. There is a need to objectively reassess the theoretical framework to ensure learning conditions in these online scenarios. The Empirical Risk Minimization Principle (ERMP) [17] represents one of the most important formal steps for the area of ML because it ensures the learning conditions for supervised algorithms. Given the Law of Large Numbers [18], a set of assumptions must hold to guarantee learning; otherwise, the ERMP becomes inconsistent. Notably, the underlying distribution is fixed, so it does not change with the data sampling; otherwise, the convergence could not be ensured given samples would follow a different probability distribution. Second, all data points must be independent and identically distributed. Both assumptions limit learning in online scenarios, where the joint probability distribution can change over time, and data observations will most certainly present some degree of dependence. It is thus crucial to formalize a time-centered framework to make the ERMP consistent so that learning can be theoretically ensured.

3. Related Work

Learning in a non-stationary environment is a big challenge for current Machine Learning methods. Algorithms that learn to optimize sequentially predictive models over a stream of data have been extensively investigated in the online learning setting [19]. Nevertheless, online learning assumes an i.i.d data sampling procedure and considers a single task domain, which sets it apart from what is necessary for these scenarios. Indeed, the above requirements call for stateful systems that continually learn over time [20]. Notably, the evolving nature of data streams in many real-world applications, such as transportation, smart home, computer security, and finance [21] motivates the relaxing of the assumption of identically distributed data, defining the concept drift [22]. Most existing SML works focused on drift detection and adaptation, proposing several drift detectors and stateful adaptive algorithms [23]. Despite their efficiency in detecting changes in data streams, the underlying assumption of the above methods is that data points are generated independently from other data points in the same stream.

Empirical experiments [9, 10] have shown that there are significant temporal dependencies among data points in a stream of data, thus making it crucial for further studies. The main issue resultant from

[2]For instance, DeepMind claims to be able to forecast the next hour of rain using a deep generative model [16].

such time dependence is that supervised learning has no guarantee according to the Statistical Learning Theory [17]. This means that any supervised model trained on top of those dependent data produce some inconclusive model since data are not sampled in an i.i.d. fashion. Only a few works directly addressed the problem of temporal dependence under these circumstances. Some works suggested handling temporal dependence by using adaptive estimation with a so-called forgetting factor, i.e., the importance of a data point in a stream is inversely proportional to its age [24]. However, these approaches are partial solutions that do not adequately consider how data points relate to each other over time. The effects of time upon streams are thus relatively unexplored in this scenario. At the time of this writing, there are very few investigations in assessing temporal dependence in SML literature [9, 10, 25–27]. Finally, to ensure data independence, dynamical system tools can reconstruct the input space to represent all dependencies in terms of a new set of dimensions [28]. Notably, the Takens' embedding theorem [29] guarantees to obtain a multidimensional space of the data stream which respects the i.i.d. assumption so learning can be theoretically ensured.

TSA provides a fundamental background to support new developments in these scenarios. Indeed, it deals with sequences of data points having a temporal order, where previous signal values present the primary, sometimes the only, source of predictive information. Effectively, according to auto-regressive models, the observations of a time series are regressed on their past lags. Another way TSA approaches incorporate past information is by using moving average models, which use past forecast errors as predictors in a multiple regression model. Auto-correlation functions assess the presence or the absence of temporal dependence. The preprocessing of the time series is usually done through transformations such as the Box-Cox method [30] for stabilizing the variance or differencing operations to remove trend or seasonality to turn data i.i.d., or, in TSA terms, to make the time series stationary.

At the same time, TSA models assume that data are stationary. The analysis is assumed to be offline with batch data without any requirement (inherent to streams) for low memory and low processing time per data point. For instance, existing algorithms for estimating ARIMA parameters, such as least squares and maximum likelihood-based methods, require access to the entire dataset in advance, violating a data stream principle and making it impossible to deal with concept drifts. So far, only a few investigations adapted the original algorithm to learn incrementally [31, 32]. Nevertheless, simple incremental learning is typically insufficient in a streaming context; it does not satisfy the time and memory requirements of real-time analytics nor implements any adaptive technique to tackle data streams' evolving nature.

4. Experiments

We performed several experiments comparing traditional Machine Learning, Time Series Analytics, and Streaming Machine Learning techniques to support our thesis. We look for a popular benchmark that resembles a nowcasting problem, and we selected the Electricity dataset [8]. The dataset comes from the Australian New South Wales Electricity Market. It contains $45,312$ instances that record electricity prices at 30-minute intervals. In origin, it was a time series, but the authors of the dataset turned it into a binary classification problem where the class label identifies the price change (UP or DOWN) related to a moving average of the last 24 hours. Due to changing consumption habits, unexpected events, and seasonality, the data is subject to concept drift. From [33], two observations can be made about this dataset. Firstly, the data are not independently distributed over time; they have a temporal dependence. If the price goes UP now, it is more likely than by chance to go UP again, and vice versa. Data are heavily auto-correlated with prominent cyclical peaks at every 48 instances (24 hours) due to

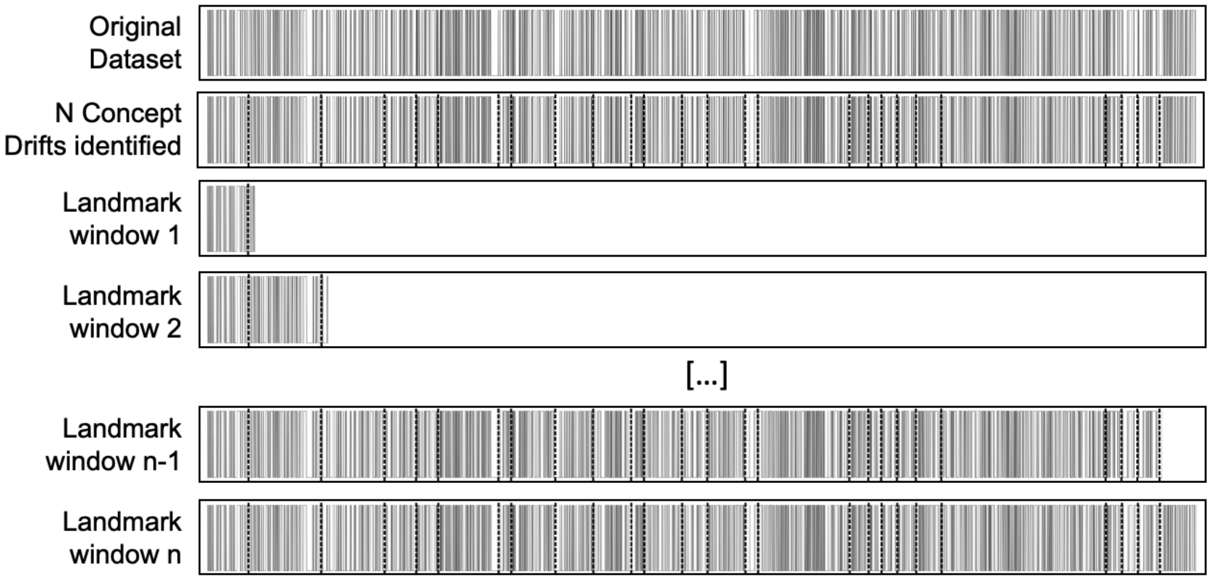


Fig. 1. The figure illustrates our method to collect experimental evidence to support our claims. The graph at the top shows the distribution of the label over time in the original Electricity dataset: the denser the bars, the more often the label changes. We used ADWIN to identify 131 concept drifts then sliced the dataset at each concept drift. We train and test models using a landmark window that starts with the first slice and grows broader a slice at a time.

electricity consumption habits. Secondly, the prior distribution of classes in this data stream is evolving. Moreover, [10] tested several models with different orders and showed that the models with an order greater than 2 did not improve the results.

In our tests, as Fig. 1 shows, we use ADWIN to detect where the concept drift occurs and divide the stream into segments referring to the same concept. In the case of the traditional Machine Learning and Time Series Analytics methods, we simulate the practice to monitor the performance of the deployed models and retrain them as soon as a concept drift occurs. For this reason, we use a landmark window that incorporates a new segment of the stream after every drift. At the first iteration, the window contains the first segment. We hold out as testing set the last 48 samples (i.e., 24 hours) and use the remaining part as the training set. In this way, we preserve the temporal correlation among data. Then, we add to the window the second segment, we retrain the model from scratch using the entire window (two segments) except the last 48 samples of the new segment added, and we test it using the 48 samples kept apart. We repeat this process for every segment.

Instead, in the case of Streaming Machine Learning models, we use the 5-fold distributed prequential cross-validation [34] to incrementally update them with the new segment instead of retraining the models from scratch. However, to make the comparison fair, we reset the metrics for each new segment. We applied this approach to simulate a common practice among the practitioners, i.e., trying to understand how a model performs only on new data, possibly representing a new concept, without considering the older data. For all the models, we used the standard hyperparameter values proposed by the libraries used (Scikit-learn[3] for ML models and River[4] for the SML and TSA ones).

[3] https://scikit-learn.org/0.22/
[4] https://riverml.xyz/0.9.0/

We use the No Change classifier (NC), a.k.a. naive or persistent method in the TSA terminology, as a baseline. As ML methods, we test the K-Nearest Neighbours (KNN), Naïve Bayes (NB), Decision Tree (DT), Random Forest (RF), Gradient Boosting, Bagging, and AdaBoost classifiers. As TSA methods, we test the Simple Exponential Smoothing with $\alpha = 0.5$ (SES_{05}), ARMA having both auto-regressive and moving average equal to 2 ($ARMA_{22}$), ARIMA having both auto-regressive and moving average equal to 2 and 1 differencing order ($ARIMA_{212}$), and SARIMA having both auto-regressive and moving average equal to 2, 1 differencing order, and a seasonality of 48 samples, 24 hours ($SARIMA_{212}$) classifiers. Knowing from [10] that the maximum order of significance on temporal dependence was 2, we used the parameters that maximized the ability of the TSA models to capture all of the information within the data. Our experiments also tested LSTM, whose performance is in line with the other tested models. Being extremely memory and time-consuming, we focus on the approaches most suitable for streaming in this work.

As SML methods, we test the Online Naïve Bayes, Online K-Nearest Neighbours, Online K-Nearest Neighbours with ADWIN, Very Fast Decision Tree (VFDT), Hoeffding Adaptive Tree (HAT), Extremely Fast Decision Tree (HATT), Adaptive Random Forest (ARF), Online Bagging (OB), Online Bagging with ADWIN (OB_{ADWIN}), Leveraging Bagging (LB), Online AdaBoost, and Adaptive XGBoost with both the push ($AXGBoost_{push}$) and the replace ($AXGBoost_{replace}$) strategy classifiers.

Lastly, we apply also two different temporal augmentations. In the first one, we add 1 or 2 past labels to the Online K-Nearest Neighbours with ADWIN ($SWT10_{KNN_{ADWIN}}$, $SWT20_{KNN_{ADWIN}}$), Very Fast Decision Tree ($SWT10_{VFDT}$, $SWT20_{VFDT}$), Hoeffding Adaptive Tree ($SWT10_{HAT}$, $SWT20_{HAT}$), and Adaptive Random Forest ($SWT10_{ARF}$, $SWT10_{ARF}$) classifiers. In the second one we also add 1 or 2 past predicted label as recommended in [35], e.g. algorithms starting with SWT12 means that uses 1 past label and 2 past predicted labels. We applied it to the Hoeffding Adaptive Tree

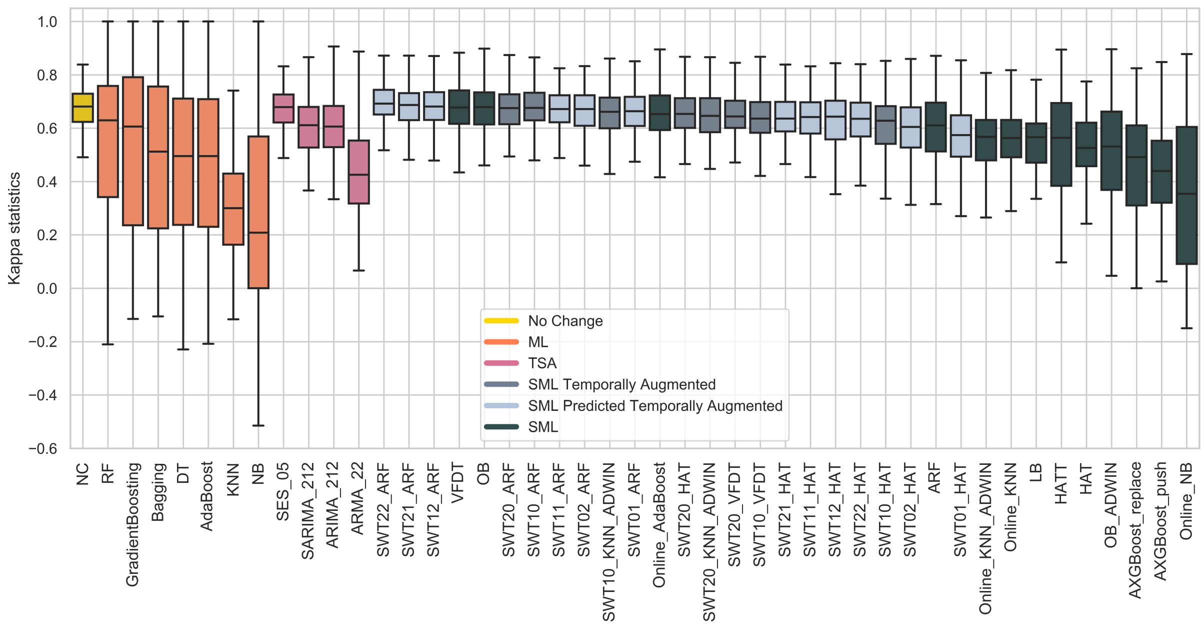


Fig. 2. Box plot of the Kappa statistics results achieved during the experiments grouped by types of algorithms (Baseline, ML, TSA, SML, SML Temporally Augmented, and SML Predicted Temporally Augmented).

($SWT01_{HAT}$, $SWT11_{HAT}$, $SWT21_{HAT}$, $SWT02_{HAT}$, $SWT12_{HAT}$, $SWT22_{HAT}$), and Adaptive Random Forest ($SWT01_{ARF}$, $SWT11_{ARF}$, $SWT21_{ARF}$, $SWT02_{ARF}$, $SWT12_{ARF}$, $SWT22_{ARF}$) algorithms.

Figure 2 shows the box plot of the Kappa statistics results achieved by all the evaluated methods, grouped by types of algorithms (Baseline, ML, TSA, SML, SML Temporally Augmented, and SML Predicted Temporally Augmented). It is noteworthy that the No Change classifier, here used as the baseline, is one of the best approaches. Nevertheless, some methods have whiskers larger than the baseline, resulting in a high variability. This behaviour is highly evident for ML methods that obtain the best maxima and worst minima. It is also interesting that the Augmented SML algorithms outperform their respective base classifier. Notably, the methods augmented with the past predictions showed the best improvements, resulting in high mean and small interquartile ranges.

For this reason, for each group, in Fig. 3 we compared the value over time of the Kappa statistics of the best and worst algorithm w.r.t. the No Change classifier. Comparing the baseline with the TSA models, we notice that the Simple Exponential Smoothing algorithm (best TSA) performs as the baseline, while the ARMA model (worst TSA) is much worse.

Undoubtedly, using only the label to apply TSA methods is not enough, and we have to consider the exogenous part of the data. To this purpose, there are the ML methods. The Random Forest (best ML) model, in some segments, is better than the baseline, while the Naïve Bayes (worst ML) is almost always worse. Moreover, both models show high and low peaks, a sign of a lack of stability. In fact, with datasets affected by continuous changes over time, as the one tested, there is the need to monitor the performance and, when they decrease, to retrain the model from scratch using more data.

We used the SML models to avoid this problem, both with and without temporal augmentation. We can observe that, in some segments, the VFDT (best SML), $SWT20_{ARF}$ (best SML Temporally Augmented), and $SWT22_{ARF}$ (best SML Predicted Temporally Augmented) classifiers achieve better results than the

Fig. 3. Kappa statistics results achieved over all the segments. The first column compares the best TSA, ML, SML, SML Temporally Augmented, and SML Predicted Temporally Augmented models w.r.t. the Baseline (No Change classifier), while the second column compares the worst TSA, ML, SML, SML Temporally Augmented, and SML Predicted Temporally Augmented models w.r.t. the Baseline.

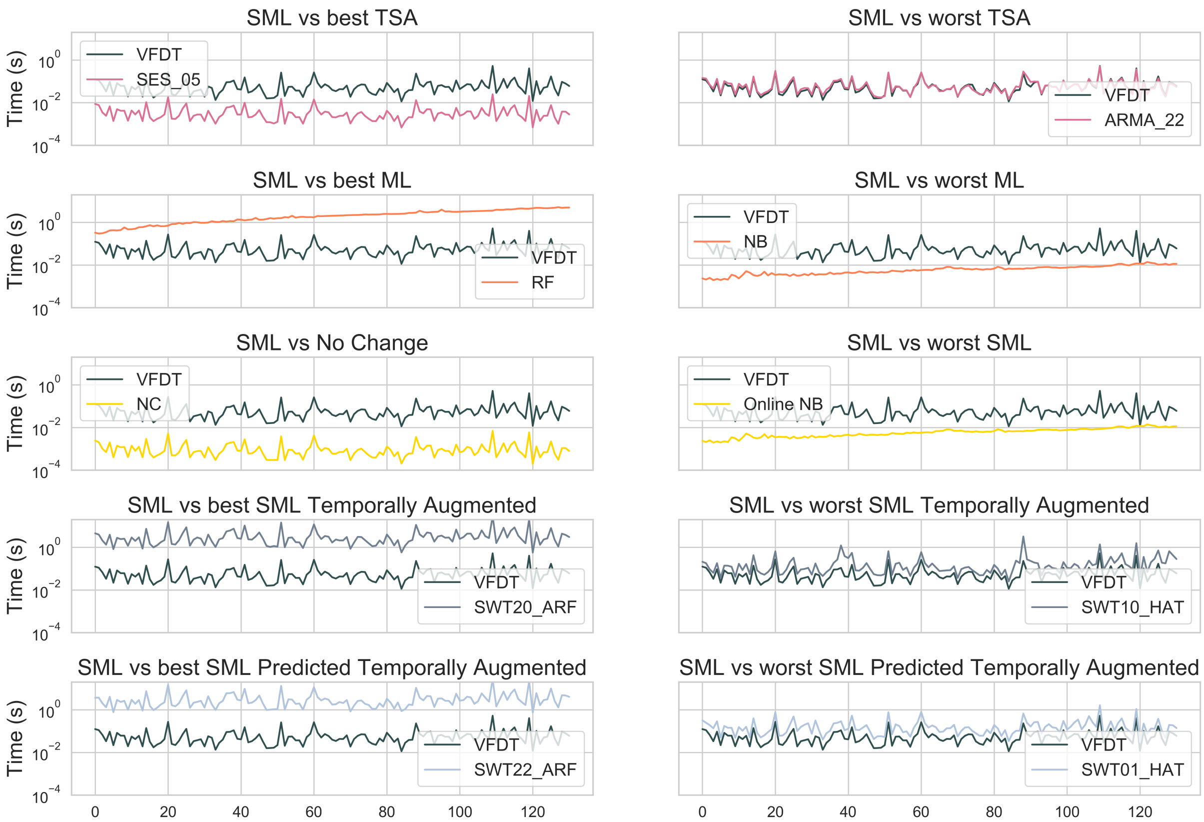
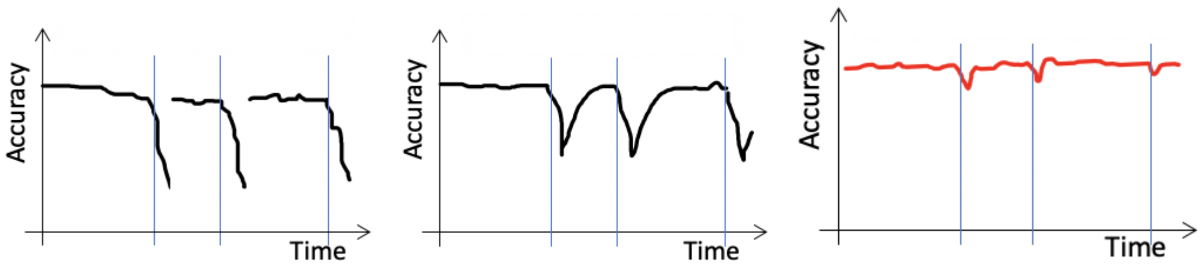


Fig. 4. Time results achieved over all the segments. The first column compares the best TSA, ML, SML, SML Temporally Augmented, and SML Predicted Temporally Augmented models w.r.t. the Baseline (No Change classifier), while the second column compares the worst TSA, ML, SML, SML Temporally Augmented, and SML Predicted Temporally Augmented models w.r.t. the Baseline.

baseline. However, they are not still the best for the entire experiment. Nevertheless, they are more stable w.r.t. the ML models, meaning that they can autonomously adapt to the concept drift occurrences.

In conclusion, it is not enough to detect when a concept drift occurs, as SML does, but there is the need to detect the changes of the temporal dependence over time, too. SML cannot do so since, once fixed, the number of temporal augmentations cannot be changed over time.

A further important requirement to consider when analyzing algorithms for data streams is the execution time, which discriminates the models that use resources most efficiently. Figure 4 shows the execution time over the segments. It is worth noticing that as ML models accumulate data (the landmark window widens), the time and resources consumed increase. This behaviour is not present in the SML and TSA online models, which, being incremental approaches and analyzing one element at a time, consume constant time and resources, except for sporadic peaks that are due to strong "model adjustments" that every so often the model does to adapt to changes. In general, this analysis gives us strong indications that the right way to manage incoming data flows is one of the incremental models.



(a) Time-series analytics **(b)** Streaming analytics **(c)** Time-evolving analytics

Fig. 5. Type of analytics. Each horizontal line represents a concept drift occurrence.

5. Challenges and Benefits

To succeed in this considerable potential, we must address a mix of outstanding conceptual and technical challenges. The first challenge is to identify a method to ignore past data when a change occurs while retaining temporal dependence that still holds. Traditional TSA techniques offer conceptual and technical solutions to measure auto-correlation [36]. Still, they all assume that the data points are identically distributed, or in TSA terms, the time-series are stationary. Thus, practitioners periodically look for changes and refresh models accordingly, leading to a performance like the one shown in Fig. 5a. On the other way round, although SML offers a wide collection of drift detectors to forget past data when changes occur and start learning in parallel a new model on the data points that follow the change [23, 37] leading to a performance like the one shown in Fig. 5b, a more sophisticated detector is required to monitor temporal dependence and selectively remember past data.

This leads to the following intriguing questions: given a data stream in which data points present temporal dependence and changes can occur at any time, can we continuously learn and fast adapt a model so that it remains relevant? Would different types of changes (abrupt, gradual, and recurrent) make the problem harder or simpler? What would be the Markov-order impact on the model (i.e., the number of the past points to consider [38])? To what extent can we harness existing SML and TSA techniques for the task? What new methods are required? A second complementary challenge is incrementally learning those models with a novel generation of stateful algorithms that process data points as they arrive and then discard them [39]. This challenge stems from the unbounded nature of data streams and the consequent impossibility of storing them. We need to design single-pass [40] and low-time-complexity [41] algorithms analogously to what is done today in stream processing, where constant-time-, and polylog-space-complexity algorithms find a vast application [42]. Moreover, those models must come with a module that explains/justifies why the predictions are considered correct and how much the users trust them. This is especially true for ambiguous situations in which multiple possible future outcomes of a given sequence are predicted.

A significant benefit of Time-Evolving Analytics would be continuously and fast adapting models to changes. The availability of predictive models in situations where historical data analytics fails to keep models relevant increases the potential for identifying new insights, taking the right decision, and acting in time. An added reason is the sequential nature of the models Time-Evolving Analytics would adaptively learn mining the short and long-lasting temporal dependence in the data streams, performing as shown in Fig. 5c. Practitioners would get a tool to explain fluctuations/trends and simultaneously predict the likelihood of possible alternative futures. Time-Evolving Analytics has the potential to drastically simplify what-if analyses in selecting among complex alternatives in a volatile and uncertain world. Last, developing stateful incremental algorithms that learn one sample at a time is very important. The smaller are the data to process, the less sophisticated and the more scalable the tools are [41].

6. The envisioned framework

In both time-series and data stream abstractions, the temporal aspect carries meaningful information; it thus requires a novel principled solution that redefines the theoretical guarantees to make learning from time-dependent data streams and evolving time series consistent.

6.1. Framework's Desiderata

Starting from TSA and SML's cornerstones, our envision framework formalizes Time-Evolving Analytics's foundations. Desiderata of the ideal framework include:

Problem agnostic. This framework is not limited to a specific setting (e.g., only classification). Instead, it includes unsupervised and semi-supervised methodologies. These two domains are particularly favourable since many real-world data streams do not necessarily provide class labels or, when they do, labels may be late enough to jeopardize the model update.

No task boundaries. Learning from the input data without requiring clear task divisions makes this framework applicable to any never-ending data stream. Moreover, this large sample size opens the opportunity to avoid structuring the model a priori and determining it from data with non-parametric models where the number and nature of the parameters can evolve over time.

Stateful learning. Acknowledging data's dynamic nature with its volume, variety, and velocity is at the framework's core. The learning and prediction rates are significant with real-time data streams, and it is most valuable when they come. In such a fast time-changing scenario, the ability to statefully detect patterns on the fly without the need to store all the data and pass through them multiple times is crucial. Ideally, models should efficiently handle a possible unlimited stream of high-dimensional ever-changing data within bounded computational and memory resources while stateful learning from one data point at a time before discarding it.

Graceful forgetting. Given the unbounded nature of data streams, graceful forgetting of trivial information is an important mechanism to balance stability and plasticity. Furthermore, it will also be possible to revisit previously seen tasks to enhance the corresponding task knowledge by discarding useless information.

Selective remembering. A further challenge is integrating new knowledge while preserving past information and retaining the temporal dependence that still holds, with the final objective of a greater generalization over time. Forward transfer or zero-shot learning are important concepts here, highlighting previously acquired knowledge to aid the learning of new tasks by increased data efficiency. The retained previous knowledge will also benefit from backward transfer, i.e., continue improving while learning future related tasks.

Adaptive learning. The framework targets dynamic environments that may change unexpectedly. Methodologies must continuously adapt to the evolution of the data distribution over time. Adjusting to such concept drift represents a natural extension for their learning systems.

Learning Sequences. Real-world data streams often contain dependence that spans multiple time steps and scales from seconds/minutes to months/years. Under those circumstances, the ability to recognize and predict this dependence in the temporal sequences becomes essential to the model's capability to make high-order predictions[5]. Models should thus learn many variable-order temporal sequences to correlate important patterns that happened many time steps in the past. Finally, an ideal algorithm should learn the order automatically and efficiently.

Forecasting alternatives. In a complex and ambiguous world, the single best prediction may not be sufficient in several situations. Indeed, there may be multiple possible future outcomes for a given temporal scenario. Therefore, instead of being limited to using maximum likelihood prediction, algorithms should output a distribution of possible simultaneous future predictions and the associated confidence levels. The model can then evaluate the likelihood of each prediction online. In case of substantial uncertainty, the ability to predict simultaneously and assess the probability of each prediction becomes crucial.

6.2. A Unifying Model

There is no consensus on which statistical properties properly hold in a real-time scenario. Practitioners apply algorithms to data streams with unwarranted assumptions that could invalidate the results obtained. It is thus necessary to converge on a comprehensive Time-Evolving Analytics model. To this end, the framework must reassess the theoretical setting, formalize management strategies for both concept drift and temporal dependence, and conceive a unifying model that incorporates temporal dependence.

In literature, most drift detection methods use statistical tests that assume independent data. Temporal dependence often violates the assumptions of the statistical tests, hence incorrectly applying them and making irrelevant the detected changes [9, 10]. There is a need to reassess the whole concept drift detection phase and the entire design decisions for change detectors. As for the former, it would be interesting to analyze the relationships between seasonality, trend, residual error, and concept drift. We should also explore the relationship between concept drift detectors and anomaly detectors, which find rare points that do not conform to the distribution. Similarities between these areas are evident, and

[5] The term order refers to Markov order.

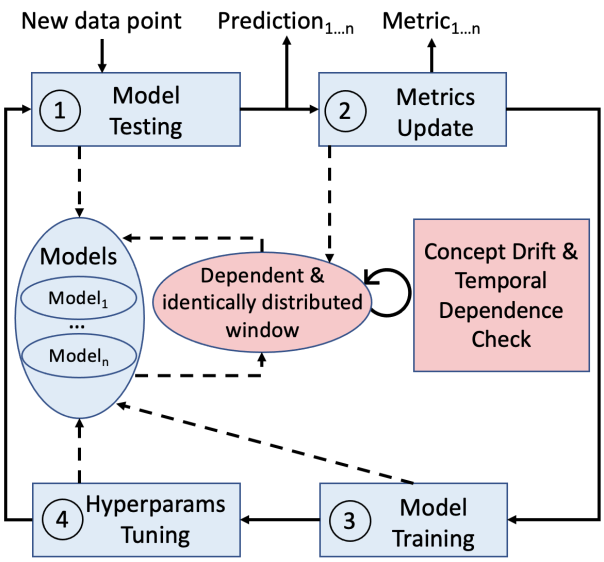


Fig. 6. Proposed architecture to manage both Concept Drifts and Temporal Dependencies.

concepts such as zero-positive learning [43] can be crucial in the presence of rare events for both concept drift and temporal dependence. As for the latter, Fig. 6 shows a proposal for a new generation of change detectors that, in parallel to the concept drift detection, discover i) if there is temporal dependence, ii) how long this dependence is, and iii) in case of temporal dependence changes when one stops, and another starts. Determining the number of lags in which a temporal dependence exists is crucial to adapt the model [26]. An idea to explore is how to apply, in an incremental way, the Granger causality test [36] or the (Partial) AutoCorrelation functions — for example, using a window containing both identically distributed and dependant data to train a model. When a new drift occurs, or data into the window stops being temporal dependent, the window should slide, and the model should adapt consequently to it.

6.3. A Unifying Methodology

There is a lack of methodologies to consider temporal dependence in the data stream while learning sequences to predict multiple alternative outcomes. The envisioned model alone is useless without developing the methodology necessary to use its algorithms. Such a methodology must include the capabilities for training and testing models (but without distinguishing the two tasks), choosing both the best hyperparameter values and model components, and selecting the most appropriate metrics.

Prequential [44] and cross-validated prequential [45] evaluation approaches include a fading factor or a windowed evaluation to "forget" old predictions performance. This practice works well to handle the concept drift occurrences, but it is at risk in the presence of temporal dependence. Indeed, they consider short-term dependence, but they fail to consider long-term ones. The same type of window proposed in Fig. 6 may solve temporal dependence' training and testing problems. Moreover, as noticed by Krawczyk et al. [40], there is an extreme need for a new framework for evaluating analytics solutions for data streams; in particular, we need new benchmark streams containing several drifts of a variety of types that ease the parallel training, testing, and comparison of different models. The starting point for the synthetic data generator is a set of Polya's Urns [46]. Each urn contains a different number of black and white balls, depending on the colour. At each timestamp, the generator extracts a ball from one of the urns without replacing them. An empty urn gets refilled in the initial status unless a concept drift occurs. In this way, the generator creates sequences for which the exchangeability assumption holds, but the i.i.d. does not.

In the hyperparameter tuning scenario, AutoML [47] is gaining popularity, but there are only a few online autoML solutions in the literature [48–50]; none of them considers temporal dependence. A valid starting point might be autoML Lifelong ML challenge[6], to then define a precise theoretical formalization that, taking into account both i.i.d. and not i.i.d. data, allowing to find the best model for a given scenario. The exploration shall not be limited to hyperparameters but should include choosing the continuous analysis pipeline's best components. If successful, these methodologies will train in parallel and resource-wise efficient ways different versions of various models and use the window envisioned in Fig. 6 to continuously adapt the hyperparameter values to stay consistent with the concept drifts occurrences and temporal dependence.

The last point is about metrics selection. During the training phase, model validation is the only way to control how good the model learning progress is. Therefore, the correct selection and computation of performance metrics is a fundamental task. On the one hand, it would be worth exploring the combination of the K-Temporal statistic [10] with the (cross-validated) prequential evaluation approach since the

[6]https://www.4paradigm.com/competition/nips2018

combination could give a valuable baseline metric to address not i.i.d. data. On the other hand, the window proposed in Fig. 6 may represent a suitable option to the (cross-validated) prequential evaluation approach combined with any chosen stand-alone metric. This combination could also give a valuable way to evaluate the model when the i.i.d. assumption does not hold. Moreover, we intend to identify new stand-alone metrics for monitoring if the model correctly considers temporal dependence during the learning phase. They are missing, and it is of fundamental importance to develop them.

7. Conclusions

We discussed several challenges that pertain online learning for time-evolving data streams. We stressed the need for unifying theory between Time-series Analytics and Streaming Machine Learning. These novel theoretical foundations will serve as a solid basis for forecasting multiple possible outcomes of a high-order sequence. Models learned with Time-Evolving Analytics will remain relevant even when changes like COVID-19 hit, thus allowing to compete in volatile and uncertain markets.

We also discussed that the No Change detector can outperform the state-of-the-art ML/TSA/SML methods used in a classification streaming evaluation. Moreover, SML approaches with temporal augmentation represent only a first reasonable solution to incorporate the temporal aspect into the learning process. All these results strengthen our claim that when combined with concept drifts, the temporal dependence might have a high impact on the learning process and the evaluation. Thus temporal dependence should be considered during the learning process of the models. We hope that this paper will open several directions for future research. The lack of such a theory is, indeed, the root cause of the current inability to adapt predictive models fast, continuously, and incrementally.

References

[1] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel et al., A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play, *Science* **362**(6419) (2018), 1140–1144.
[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., Imagenet large scale visual recognition challenge, *International journal of computer vision* **115**(3) (2015), 211–252.
[3] Seagate, Dataage 2025 - the digitization of the world, 2020.
[4] K. Panetta, Gartner Top 10 Data and Analytics Trends for 2021, 2021, (Online; accessed).
[5] W.D. Heaven, Our weird behavior during the pandemic is messing with AI models, *MIT Technology Review* (2020).
[6] A. Bifet, R. Gavaldà, G. Holmes and B. Pfahringer, *Machine learning for data streams: with practical examples in MOA*, MIT press, 2018.
[7] G.E.P. Box and G.M. Jenkins, Time Series Analysis: Forecasting and Control, Prentice Hall PTR, 1994.
[8] M. Harries and N.S. Wales, SPLICE-2 Comparative Evaluation: Electricity Pricing, 1999.
[9] A. Bifet, Classifier Concept Drift Detection and the Illusion of Progress, in: *ICAISC (2)*, LNCS, Vol. 10246, Springer, 2017, pp. 715–725.
[10] I. Zliobaite, A. Bifet, J. Read, B. Pfahringer and G. Holmes, Evaluation methods and decision theory for classification of streaming data with temporal dependence, *Mach. Learn.* **98**(3) (2015), 455–482.
[11] J.W. Wilson, N.A. Crook, C.K. Mueller, J. Sun and M. Dixon, Nowcasting thunderstorms: A status report, *Bulletin of the American Meteorological Society* **79**(10) (1998), 2079–2100.
[12] D. Giannone, L. Reichlin and D. Small, Nowcasting: The real-time informational content of macroeconomic data, *Journal of Monetary Economics* **55**(4) (2008), 665–676.
[13] M.A. Johansson, A.M. Powers, N. Pesik, N.J. Cohen and J.E. Staples, Nowcasting the spread of chikungunya virus in the Americas, *PloS one* **9**(8) (2014), e104915.
[14] J.T. Wu, K. Leung and G.M. Leung, Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: a modelling study, *The Lancet* **395**(10225) (2020), 689–697.

[15] J. Zhang, R. Verschae, S. Nobuhara and J.-F. Lalonde, Deep photovoltaic nowcasting, *Solar Energy* **176** (2018), 267–276.

[16] S. Ravuri, K. Lenc, M. Willson, D. Kangin, R. Lam, P. Mirowski, M. Fitzsimons, M. Athanassiadou, S. Kashem, S. Madge et al., Skilful precipitation nowcasting using deep generative models of radar, *Nature* **597**(7878) (2021), 672–677.

[17] V. Vapnik, *The nature of statistical learning theory*, Springer science & business media, 1999.

[18] L. Devroye, L. Györfi and G. Lugosi, *A probabilistic theory of pattern recognition*, Vol. 31, Springer Science & Business Media, 2013.

[19] S. Shalev-Shwartz, Online Learning and Online Convex Optimization, *Found. Trends Mach. Learn.* **4**(2) (2012), 107–194.

[20] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh and T. Tuytelaars, A continual learning survey: Defying forgetting in classification tasks, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[21] I. Žliobaitė, M. Pechenizkiy and J. Gama, An overview of concept drift applications, *Big data analysis: new algorithms for a new society* (2016), 91–114.

[22] A. Tsymbal, The problem of concept drift: definitions and related work, *Computer Science Department, Trinity College Dublin* **106**(2) (2004), 58.

[23] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama and G. Zhang, Learning under Concept Drift: A Review, *IEEE Trans. Knowl. Data Eng.* **31**(12) (2019), 2346–2363.

[24] D.A. Bodenham and N.M. Adams, Continuous monitoring for changepoints in data streams using adaptive estimation, *Stat. Comput.* **27**(5) (2017), 1257–1270.

[25] J. Read, R.A. Rios, T. Nogueira and R.F. de Mello, Data Streams Are Time Series: Challenging Assumptions, in: *BRACIS (2)*, LNCS, Vol. 12320, Springer, 2020, pp. 529–543.

[26] Q. Duong, H. Ramampiaro and K. Nørvåg, Applying temporal dependence to detect changes in streaming data, *Appl. Intell.* **48**(12) (2018), 4805–4823.

[27] Y. Song, J. Lu, H. Lu and G. Zhang, Learning Data Streams With Changing Distributions and Temporal Dependency, *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[28] L. de Carvalho Pagliosa and R.F. de Mello, Applying a kernel function on time-dependent data to provide supervised-learning guarantees, *Expert Syst. Appl.* **71** (2017), 216–229.

[29] F. Takens, Detecting strange attractors in turbulence, in: *Dynamical systems and turbulence, Warwick 1980*, Springer, 1981, pp. 366–381.

[30] R.M. Sakia, The Box-Cox transformation technique: a review, *Journal of the Royal Statistical Society: Series D (The Statistician)* **41**(2) (1992), 169–178.

[31] O. Anava, E. Hazan, S. Mannor and O. Shamir, Online Learning for Time Series Prediction, in: *COLT*, JMLR Workshop and Conference Proceedings, Vol. 30, JMLR.org, 2013, pp. 172–184.

[32] C. Liu, S.C.H. Hoi, P. Zhao and J. Sun, Online ARIMA Algorithms for Time Series Prediction, in: *AAAI*, AAAI Press, 2016, pp. 1867–1873.

[33] A. Bifet, J. Read, I. Zliobaite, B. Pfahringer and G. Holmes, Pitfalls in Benchmarking Data Stream Classification and How to Avoid Them, in: *ECML/PKDD (1)*, LNCS, Vol. 8188, Springer, 2013, pp. 465–479.

[34] A. Bifet, G.D.F. Morales, J. Read, G. Holmes and B. Pfahringer, Efficient Online Evaluation of Big Data Stream Classifiers, in: *KDD*, ACM, 2015, pp. 59–68.

[35] T.G. Dietterich, Machine Learning for Sequential Data: A Review, in: *SSPR/SPR*, Lecture Notes in Computer Science, Vol. 2396, Springer, 2002, pp. 15–30.

[36] C.W. Granger, Investigating causal relations by econometric models and cross-spectral methods, *Econometrica: journal of the Econometric Society* (1969), 424–438.

[37] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy and A. Bouchachia, A survey on concept drift adaptation, *ACM Comput. Surv.* **46**(4) (2014), 44:1–44:37.

[38] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE* **77**(2) (1989), 257–286.

[39] B. Babcock, M. Datar, R. Motwani et al., Load shedding techniques for data stream systems, in: *Proceedings of the 2003 Workshop on Management and Processing of Data Streams*, Vol. 577, Citeseer, 2003.

[40] B. Krawczyk, L.L. Minku, J. Gama, J. Stefanowski and M. Wozniak, Ensemble learning for data stream analysis: A survey, *Inf. Fusion* **37** (2017), 132–156.

[41] A. McGregor, A. Pavan, S. Tirthapura and D.P. Woodruff, Space-efficient estimation of statistics over sub-sampled streams, *Algorithmica* **74**(2) (2016), 787–811.

[42] Z. Milosevic, W. Chen, A. Berry and F.A. Rabhi, Chapter 2 - Real-Time Analytics, in: *Big Data*, R. Buyya, R.N. Calheiros and A.V. Dastjerdi, eds, Morgan Kaufmann, 2016, pp. 39–61.

[43] T. Lee, J. Gottschlich, N. Tatbul, E. Metcalf and S. Zdonik, Greenhouse: A Zero-Positive Machine Learning System for Time-Series Anomaly Detection, 2018.

[44] J. Gama, R. Sebastião and P.P. Rodrigues, On evaluating stream learning algorithms, *Mach. Learn.* **90**(3) (2013), 317–346.

[45] A. Bifet, G.D.F. Morales, J. Read, G. Holmes and B. Pfahringer, Efficient Online Evaluation of Big Data Stream Classifiers, in: *KDD*, ACM, 2015, pp. 59–68.

[46] H. Mahmoud, *Pólya urn models*, CRC press, 2008.

[47] F. Hutter, L. Kotthoff and J. Vanschoren (eds), *Automated Machine Learning - Methods, Systems, Challenges*, The Springer Series on Challenges in Machine Learning, Springer, 2019.

[48] M. Bahri, B. Veloso, A. Bifet and J. Gama, AutoML for Stream k-Nearest Neighbors Classification, in: *IEEE BigData*, IEEE, 2020, pp. 597–602.

[49] J.G. Madrid, H.J. Escalante, E.F. Morales, W. Tu, Y. Yu, L. Sun-Hosoya, I. Guyon and M. Sebag, Towards AutoML in the presence of Drift: first results, *CoRR* **abs/1907.10772** (2019).

[50] J. Wilson, A.K. Meher, B.V. Bindu, S. Chaudhury, B. Lall, M. Sharma and V. Pareek, Automatically optimized gradient boosting trees for classifying large volume high cardinality data streams under concept drift, in: *The NeurIPS'18 Competition*, Springer, 2020, pp. 317–335.