International Journal of Mathematical, Engineering and Management Sciences
Vol. 10, No. 3, 777-796, 2025
https://doi.org/10.33889/IJMEMS.2025.10.3.039

Ram Arti Publishers

A Hybrid Framework for Real-Time Data Drift and Anomaly Identification Using Hierarchical Temporal Memory and Statistical Tests

Subhadip Bandyopadhyay
Global AI Accelerator,
Ericsson, Bangalore, Karnataka, India.
Corresponding author: subhadip.bandyopadhyay@ericsson.com

Joy Bose
Global AI Accelerator,
Ericsson, Bangalore, Karnataka, India.
E-mail: joy.bose@ericsson.com

Sujoy Roy Chowdhury
Global AI Accelerator,
Ericsson, Bangalore, Karnataka, India.
E-mail: sujoy.roychowdhury@ericsson.com

(Received on July 15, 2024; Revised on February 5, 2025 & February 19, 2025; Accepted on February 25, 2025)

Abstract
Data Drift refers to the phenomenon where the generating model behind the data changes over time. Due to data drift, any model built on the past training data becomes less relevant and inaccurate over time. Thus, detecting and controlling for data drift is critical in machine learning models. Hierarchical Temporal Memory (HTM) is a machine learning model developed by Jeff Hawkins, inspired by how the human brain processes information. It is a biologically inspired model of memory similar in structure to the neocortex and whose performance is claimed to be comparable to state of the art models in detecting anomalies in time series data. Another unique benefit of HTMs is their independence from training and testing cycles; all the learning takes place online with streaming data, and no separate training and testing cycle is required. In the sequential learning paradigm, the Sequential Probability Ratio Test (SPRT) offers unique benefits for online learning and inference. This paper proposes a novel hybrid framework combining HTM and SPRT for real-time data drift detection and anomaly identification. Unlike existing data drift methods, our approach eliminates frequent retraining and ensures low false positive rates. HTMs currently work with one dimensional or univariate data. In a second study, we also propose an application of HTM in a multidimensional supervised scenario for anomaly detection by combining the outputs of multiple HTM columns, one for each data dimension, through a neural network. Experimental evaluations demonstrate that the proposed method outperforms conventional drift detection techniques like the Kolmogorov-Smirnov (KS) test, Wasserstein distance, and Population Stability Index (PSI) in terms of accuracy, adaptability, and computational efficiency. Our experiments also provide insights into optimizing hyperparameters for real-time deployment in domains such as Telecom.

Keywords- Hierarchical temporal memory (HTM), Sequential probability ratio test (SPRT), Time series, Real-time anomaly detection, Data drift detection, Streaming data analysis, Hybrid machine learning models, Telecom network monitoring, AI powered data drift detection.

1. Introduction

Data drift is a phenomenon where the generating model of some data changes over time. In presence of data drift, any model trained on older data becomes less relevant, and its performance degrades with respect to the recently available data. If a machine learning (ML) model is constructed based on some training data, it is necessary to periodically retrain the ML model based on recent data to maintain a consistent performance, as the training data distribution may vary over time due to data drift.

Data drift detection is a long studied problem (Wang and Abraham, 2015; Wadewale and Desai, 2015; Kadam, 2019; Gemaque et al., 2020). It has historically appeared in various forms in statistical science literature including change point detection, statistical process control, and statistical inference of two or multiple sample tests for population homogeneity. Recently, with the rise of artificial neural network based approaches, deep learning techniques for data drift detection have gained prominence in the literature. However, with increasing demand for time sensitive applications, there is a growing emphasis on near real-time performance in data drift detection methodologies, particularly in use cases within the telecom and finance domain.

Some examples of applications where data drift or anomaly detection is useful include fraud detection in banking and finance by detecting anomalies in transactions, anomaly detection in the healthcare industry by identifying anomalies in patients' vital signs, Internet of Things (IoT) based sensor anomaly detection, and network intrusion detection in cybersecurity.

Hierarchical Temporal Memory (HTM) (Hawkins et al., 2019; Numenta, 2019a) is a neural network model that is more biologically plausible than deep learning. It is an unsupervised method with no separation between training and testing phases. Consequently, it performs well with streaming data. The network architecture in HTMs closely mirrors the cortical columns of the neocortex of the mammalian brain. The nature of HTM methodology falls under the sequential learning paradigm, which is highly relevant in use cases involving time series data.

Sequential learning is a well-established field in statistics that is currently being applied in multiple domains due to its relevance and general applicability (Sutskever et al., 2014). The Sequential Probability Ratio Test (SPRT) (Piegorsch and Padgett, 2011) is a fundamental statistical tool primarily used for sequential statistical inference. Although a confluence of sequential learning approaches and HTM-based techniques appears to be a natural fit, existing literature does not provide much evidence of such integration. We consider an unsupervised scenario for data drift detection, which is a more realistic scenario, as obtaining labelled data in a drifted scenario is often impractical since identification of drift is required at the outset. HTM and SPRT have been applied in various use cases involving time series data, such as time series anomaly detection, statistical process control and forecasting, to name a few. Specific applications can be found in Wu et al. (2018) for HTM and in Schoonewelle et al. (1995) for SPRT.

In this paper, we explore building an online solution for data drift detection (in an unsupervised scenario) and anomaly detection (in a supervised scenario) by combining HTM and SPRT. The idea of integrating these two sparsely explored approaches is driven by the need for an online anomaly and data drift detector with minimum or no retraining burden. Since HTM does not require retraining and both HTM and SPRT are suitable for near real-time adoption, an intuitive solution is to combine them. The advantage of this approach is that no frequent retraining of the drift detection algorithm is needed for near-real time applications.

It can be noticed that HTM and SPRT serve two different purposes. Where HTM basically predicts the similarity of the next time point data with the most recent data, SPRT compares the current data pattern with the historical data, mostly under some assumptions related to the data distribution. In a real-life scenario, the distribution of the data is not known, and we bypass this problem by leveraging HTM output and formulating it as a binary decision, namely, if the current data point is coming from a historical data distribution. With this formulation, irrespective of the real data distribution, the reformulated sequence of the data can be modelled under sequence of Bernoulli random variable since the current data point is either from the historical distribution or not from the same distribution. Hence, the rich field of SPRT can be

applied subsequently to infer a data drift, namely, if the current data point is not from the historical data distribution, then a drift can be detected to be in force.

There are existing works on concept drift detection on streaming data (Pesaranghader et al., 2018; Souza et al., 2020). There are also existing techniques for testing the homogeneity of a population based on two samples, e.g., Kolmogorov-Smirnov (KS) test, Wasserstein distance (Vallender, 1974), Population stability index or Jensen Shannon divergence (Endres and Schindelin, 2003; Cover and Thomas, 2006; Yurdakul, 2018) which are suitable for a rolling window based formulation for online data drift detection. However, many of these suffer from a high false positive rate, as we illustrate in our simulation experiments in Section 4.

In the unsupervised scenario, the proposed approach is divided into two phases. In the first phase, HTM is utilized to continuously model the probability of similarity of incoming sample data with respect to observed data points from recent past. The first phase output is consumed in the second phase to build an SPRT based drift detection algorithm.

In the supervised scenario, we assume that either knowledge from subject matter experts (SMEs) or any other source is available for outlier labelling. The HTM layer, as in supervised scenario, is considered for each data dimension. We propose a HTM output combiner via a Neural Network (NN) construction for anomaly detection. We have used a combination of existing models to generate the ground truth for the supervised scenario, since the experts' data labelling was not available readily.

The key contributions of this work are as follows:
(a) A hybrid methodology combining HTM and SPRT for efficient drift detection in streaming data.
(b) Extension of HTM for multivariate anomaly detection using a neural network combiner.
(c) An evaluation of the proposed framework against other methods, highlighting its better performance and adaptability.
(d) Insights into hyperparameter tuning for optimizing the framework's performance.

The rest of the paper is structured as follows. Section 2 reviews related work in the areas of HTMs, SPRT and data drift detection. Section 3 introduces the motivation and describes the application of combined HTM and SPRT for drift detection in unsupervised scenarios. Section 4 extends the approach of combining HTM using a neural network combiner to multivariate and supervised scenarios. Section 5 concludes the paper with a discussion on limitations and future directions.

2. A Brief Introduction to HTM and Related Work

In this section, we introduce HTMs and its application in anomaly detection.

2.1 Hierarchical Temporal Memory (HTM)

Hierarchical temporal memory was first introduced by Jeff Hawkins, based on the groundwork in the book "On intelligence" and further developed through Numenta's research efforts (Hawkins et al., 2019; Numenta, 2019a). HTMs are biologically inspired memory models that mimic the structure of the neocortex, functioning as associative memory neural networks. They are considered more biologically plausible than the traditional deep learning models such as Convolutional Neural Networks (CNNs).

HTM's architecture closely resembles the cortical columns of the neocortex of the mammalian brain. It incorporates properties of the neocortex such as hierarchical structure, sparse distributions, Hebbian learning (following Hebb's law), noise tolerance, and adaptability to changing data distributions or data

drift. HTMs operate in an unsupervised manner, eliminating the need for separate training and testing phases, making them well-suited for streaming data applications.

HTM's computational mechanisms include encoding input data as binary vectors called sparse distributed representations (SDRs), implementing associative memory, enabling online and continuous learning, predicting the future inputs, and generating an anomaly score that quantifies deviations from learned patterns and estimates the likelihood of the current input being anomalous.

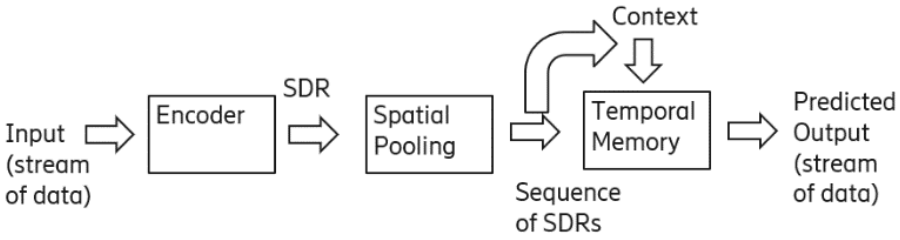


Figure 1. High level architecture of HTMs.

HTMs have the following components:

- **Encoder:** this is a layer of neurons that takes the input stream of data and encodes it as a binary vector. All types of input data, whether it be numbers, text, images, audio or video, are encoded as binary vectors.

I_t = input data at time t
Output of encoder = x_t = Binary representation of I_t [1 0 0 1 0 0 ….. 1].

- **Spatial Pooler:** This is a layer that converts the binary vector using an encoding scheme called sparse distributed representation or SDR. Sparse refers to the fact that most of the bits are 0s and a few (around 2%) are 1s. Multiple neighboring HTM cells are activated for each bit of data, and the overlap between two SDRs is the measure of similarity. It takes the binary vectors as input and activates (sets to a predictive state) columns which have the highest overlap with the input vector.

Output of spatial pooler = $a(x_t)$ = SDR representing x_t.

- **Temporal memory:** This layer incorporates the temporal aspects of the data, i.e., it remembers the relation between the current data and previous data, by having an explicit memory to represent the state or context. All the data is encoded as SDRs as mentioned earlier. For each new data that comes in, it predicts what the next data should be. It implements Hebbian learning (Hebb, 2005) which is a biologically plausible learning algorithm based on Hebb's rule: cells that fire together, wire together. Since the SDRs are binary vectors, only some bits are 1s, which activate the HTM neurons connected to those bits and set the connection weight to 1.

Output of temporal memory = $\pi(x_t)$ = SDR predicting $a(x_{t+1})$ where x_{t+1} means the input at time (t+1).

- **Anomaly Detector:** This module outputs the log likelihood of the current input being an anomaly. It takes two inputs: the SDR predicting $a(x_{t+1})$ and the output of the spatial pooler $a(x_t)$ and outputs the log likelihood of anomaly at time t.

Output of anomaly detector = l_t = Log likelihood of anomaly.
Figure 1 shows the high-level architecture of the HTMs.

There are a number of available implementations of HTMs, such as Numenta, which is proprietary and HTM-core, which is open-source. These provide practical frameworks for applying HTMs to real-world anomaly detection and data drift detection tasks. In this paper, we have used the open-source HTM-core (Otahal et al., 2019) library to implement HTMs.

2.2 Anomaly Detection Aspect of HTMs

The basic principle of using HTMs for anomaly detection is as follows: the HTM learns a sequence of data and predicts what the next data should be at any given time, based on what it has learnt in the past. If the prediction does not match the actual new data that is coming in, that means a potential anomaly (Numenta, 2019b).

The system for anomaly detection thus compares the predicted data at time t with the actual data, and gets a prediction error, from which it decides the likelihood of an anomaly. It can work in real time without any separation between test data and training data.

There have been a few studies in the literature (Ahmad and Purdy, 2016; Wu et al., 2018; Anandharaj and Sivakumar, 2019; Numenta, 2019a; Barua et al., 2020) that show how HTMs perform comparably or better for anomaly detection than other conventional ML techniques. Moreover, HTMs have the added advantage of one-shot learning (only one instance of a data is sufficient for the HTM to learn the associations), which makes them suitable for use with streaming data.

Figure 2 shows the overview of an anomaly detection system using HTMs.

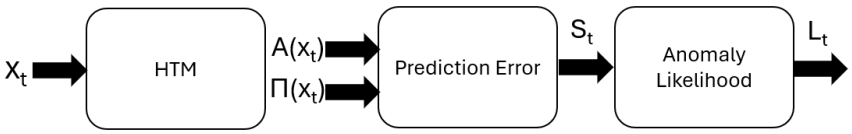


Figure 2. Overview of architecture for anomaly detection system using HTMs.

Figure 3 shows the architecture of a single HTM for anomaly detection, where the input stream is in a single dimension, with an additional module called drift detector that detects the drift in the input stream, if any.

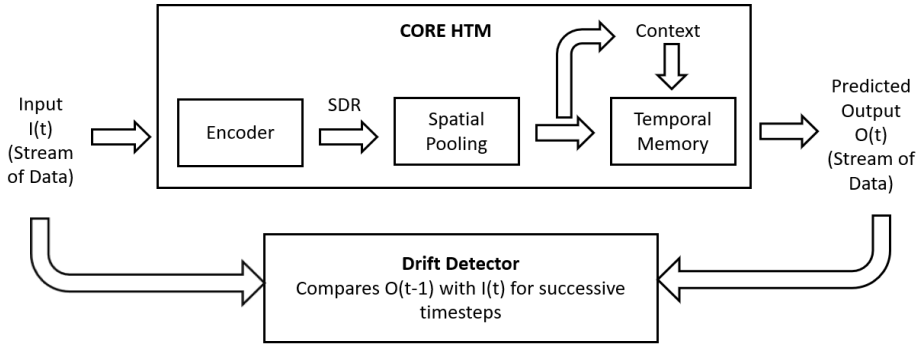


Figure 3. Architecture of using HTM for anomaly detection in a single dimensional scenario.

Figure 4 shows the architecture of the anomaly detector where the input stream is in multi-dimensional.

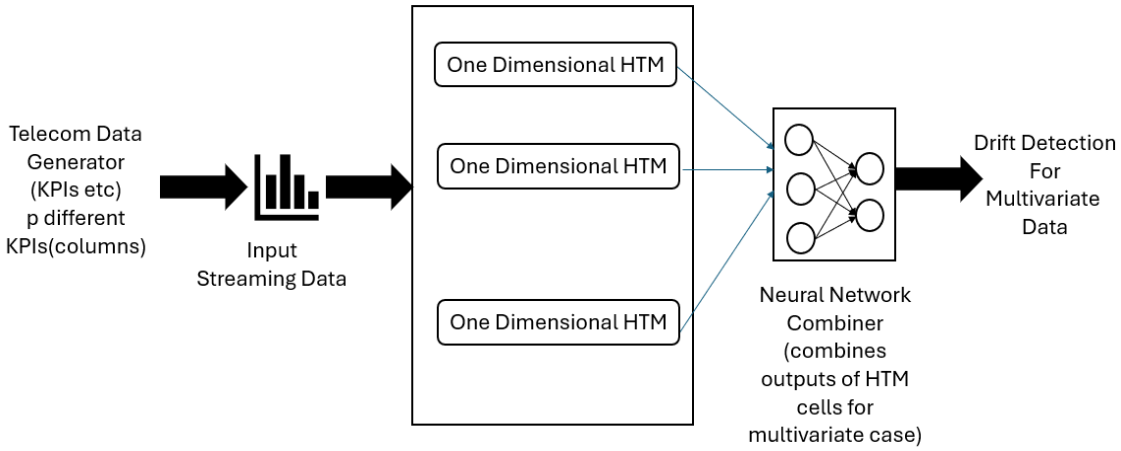


Figure 4. Architecture of using HTM for anomaly detection in a multi-dimensional scenario.

3. Data Drift Detection through Hybridization of HTM and SPRT in an Unsupervised Scenario

3.1 Motivation and Overview of the Solution

Data drift is a critical issue that is needed to be addressed to make any data based algorithmic task, as a part of any application pipeline, consistent and reliable in field performance (Gemaque et al., 2020). A generic problem for any data-based drift detection algorithm is the frequent need of retraining. Each time a drift is detected, the underlying data distribution changes and hence re-calibration of algorithm is needed with respect to the new data distribution. For non-parametric approaches like Chebyshev's inequality (Marshall and Olkin, 1960) driven k-σ limit-based drift detection or z-statistic based drift detection, the mean and variance re-computation is necessary when the data distribution changes after a data drift. Any automated pipeline must address this critical requirement through re-training, which ultimately increases the cost and complexity of maintaining an online, real-time, or near-real-time autonomous drift detector.

To address this problem, we propose to break this requirement into two hierarchical parts. In the first phase of our approach, a HTM layer is utilized to continuously model and output the probability of similarity of incoming sample data with the observed data points from the recent past. HTM, by its formulation, continuously adapts to changes in the recent data distribution within a reasonable time lag. Hence the probability output from the HTM layer consistently re-adjusts itself with respect to the new environment, also known as the drift in the data distribution.

The first stage outputs feed an SPRT based drift detection algorithm in the second stage. We re-formulate the problem, through formulation of an appropriate hypothesis, into a Bernoulli distribution based SPRT for Bernoulli parameters. The SPRT in the second stage can continuously operate within the lagged time window to assess the similarity of incoming data with recent past observations and detect drift. When a drift is detected, after the lagged time, the HTM re-adjusts itself with the change in data distribution. Subsequently, the SPRT module continues the drift detection seamlessly, now with respect to the new data, without any break. Thus, the whole flow continues seamlessly.

3.2 Re-Formulation of the Problem

We set the assumptions, definitions and notations below before formal algorithmic steps.

Assumption 1: We assume that the 'd' most important dimensions are identified through exploratory data analysis.

For a specific dimension, we re-formulate the problem of data drift detection in the following way:

Define p_i = Probability that the i^{th} dimension follows a new distribution due to data drift.

It can be noted that p_i is estimated and outputted by the HTM layer consistently corresponding to the current data point as a quantification of the chance that the current observation is significantly different to the recent past data pattern. Hence a sequence of p_i is obtained consistently from the HTM layer.

Proposition 1: Following the definition of p_i stated above and a suitable p_{null} and p_{alt} in (0,1), a data drift can be associated with the hypothesis $p_i \geq p_{alt}$, which represents that the data pattern has changed significantly (e.g., for a $p_{alt} = 0.65$). On the other hand, the hypothesis $p_i \leq p_{null}$ (e.g., $p_{null} = 0.45$) can be associated with the hypothesis that there is no drift. In the similar line of thought, the hypothesis $p_{null} < p_i < p_{alt}$ corresponds to the lack of data support scenario to decide that a data drift has happened and hence an inference on data drift occurrence needs more data points. Thus, a data drift detection problem can be formulated as a sequential hypothesis testing problem and SPRT is a natural candidate as the apt tool for testing data drift.

Proposition 2: p_{alt} and p_{null} can be looked upon as two tuning parameters which can either be obtained as users' input depending on the conservativeness of the user's perception, or based on subject matter experts' (SME) knowledge or set suitably based on historical data driven analysis in the specific use case context.

Note that setting the value of p_{null} below 0.5 safeguards a higher chance of a false positive in drift detection.

Next, we re-formulate the HTM layer output as Bernoulli random variables distribution through a re-parametrization.

Definition 1: Define htm_t = output of the HTM (for a specific data dimension) at time point t. This represents the likelihood that the new observation at time t deviates from the historical data.

Definition 2: Let us define $c_t = 1$ if $htm_t >$ bin_threshold where bin_threshold in range (0,1),
$= 0$ otherwise.

Lemma 1: Under the assumption of independent sample data, $c_t \sim$ Bernoulli $(0, p_i^*)$, $t = 1,2,..$ where p_i^* represent the probability of data drift event, for the i-th data dimension, corresponding to the data drift hypothesis proposed in Proposition 1.

Proof: The proof follows directly from the definition of Bernoulli distribution once we note the following points.

(i) p_{alt} mentioned in the hypothesis in Proposition 1 and bin_threshold in the definition of c_t has one to one mapping.
(ii) The event $c_t = 1$ and onset of data drift hypothesis in Proposition 1 corresponds to each other.
(iii) $P(c_t = 1) = p_i^*$.
(iv) The sequence $\{c_t, t=1,2,..\}$ is a sequence of independent events.

Remark 1: Implication of Lemma 1 is that the random event, 'i-th data dimension's distribution remained unchanged at time point t, t = 1,2,..', can be considered approximately following Bernoulli (p_i^*) where p_i^* quantifies the similarity likelihood of t-th observation and recent past data points. Note that under no drift hypothesis, $p_i^* \leq p_{null}$, for i = 1,..,d.

Remark 2: We can extend Proposition 1 for each of the data dimensions and conceptualize p_{null} and p_{alt} varying accordingly. This generic approach can accommodate more flexibility in the dimension wise drift detection where the definition of a drift can be tuned as per the corresponding data dimensional feature and importance. For more important data dimensions (typically determined by the use case or an exploratory data analysis based on historical data) we can adapt a conservative view to reduce the false discovery rate.

3.3 Constructing SPRT for Checking Data Drift in a Single Dimension

First, we state the relevant assumptions and definitions.

Assumption 2: We assume that the observed data has a time stamp so that we can perceive it as a data stream.

For a single data dimension, we want to derive a SPRT pipeline based on the HTM output to test, in continuum, the null hypothesis that there is no data drift, against the alternative that a data drift has happened.

To construct a SPRT, we need the upper thresholds of two types of error, namely type 1 and type 2 error.

Definition 3:
(i) Let us define a = probability of type 1 error = P (we infer drift has happened whereas it has not).
(ii) Define b = probability of type 2 error = P (we infer no drift where actually there is a drift).

It can be re-emphasized that these choices should be governed by the specific business use case. Where retraining is less costly, we can relax a and increase recall.

In our simulation experiment, we have considered a = 0.05 and b = 0.005 that gave a reasonably good detection.

Remark 3: We introduce a rescaling of htm_t and work with the rescaled values to construct c_t (refer Definition 1 in previous subsection). The re-scaled score from HTM encodes the dissimilarity between the observation at time point t and the prediction of the observation from HTM pipeline in a same way as the original HTM output. This re-scaling is a window based smoothing type approach to regularize the abrupt fluctuations that may result in computational instability.

We describe this re-scaling process in the next subsection 3.4.

Following Lemma 1 in previous subsection, we can consider the sequence $\{c_t ; t=1,2,.. \}$ is coming from a Bernoulli distribution with parameter 'p' where p quantified the probability that the observation at time point 't' is a sample from a new distribution. Thus, testing for a low value of 'p' will lead to test for a data drift as stated in Theorem 1 below.

Definition 4: Define Cm_t = cumulative sum of c_t up to time point t.

Theorem 1: Following the SPRT rule for Bernoulli sequence $\{c_t\}$, we identify a drift has set in if:

$$Cm_t > Upper_{\_limit} = \frac{\log(\frac{1-b}{a}) + t*[\log(\frac{1-p\_nul}{1-p\_alt})]}{\log(\frac{p\_alt}{p\_null}) - \log(\frac{1-p\_alt}{1-p\_nul})} \quad (1)$$

We identify no change of data distribution if:

$$Cm_t < Lower_{\_limit} = \frac{\log(\frac{b}{1-a}) + t*[\log(\frac{1-p\_nul}{1-p\_alt})]}{\log(\frac{p\_alt}{p\_null}) - \log(\frac{1-p\_alt}{1-p\_nul})} \quad (2)$$

Proof: Following Proposition 1, 2 and Lemma 1, since we have framed the data drift detection problem as testing of hypothesis based on Bernoulli formulation of the HTM output data SPRT construction, the proof follows directly from the theory of SPRT (e.g., Piegorsch and Padgett, 2011). More explanation around Equations (1) and (2) can be explored in Piegorsch and Padgett (2011).

Remark 4: For the multidimensional scenario, SPRT based control chart can be run simultaneously for all the dimensions and a final data drift detection rule can be flexibly formulated combining dimension wise drift detection inference. For example, a conservative drift detection rule can be formulated as, if drift is detected for at least one of dimensions, a data drift is inferred as final output. This additional layer gives us a flexibility of use case specific drift detection formulation.

We continue monitoring the process and identify onset of a data drift when the process is out of control or no data drift when the process is within control. Whenever a data drift onset decision is reached, the SPRT is started afresh from the next time point. Note that the hyperparameters a, b, p_{alt} and p_{null} remains unaltered.

The advantage that HTM brings is that when the drift is completed, the estimated HTM probability will stabilize in a lower range such as (0,0.5) automatically and hence the process will again show as under control, or as in our case, no data drift as all the new data is coming from the new distribution after a drift.

We will be able to identify the drift window directly from observing the charts (e.g., **Figures 5 to 7** in univariate scenario). A data drift window begins when at least one data dimension exceeds the control threshold based on Cm_t comparison.

Remark 5: After a drift starts, monitoring each data dimension continues with SPRT being re-started and we can identify when the drift ends by identifying a time point when a new drift is detected in any of the single dimensions. During this time duration, the SPRT check for all the dimensions remain below the lower threshold, as the data points in this duration are generated by the new data distributions.

It can be noted that a more appropriate way to accommodate a data drift detection in a multidimensional scenario is to consider both HTM and SPRT together in multidimension. Research in the multidimensional HTM is yet to be addressed in the literature. Therefore, we consider unidimensional HTM output. Though multidimensional SPRT is well addressed in literature, we however considered a rule-based approach on unidimensional SPRT to illustrate the concept.

3.4 Rescaling of HTM Output for Anomaly Score Generation in Unsupervised Scenario

The output of core HTM is a similarity score of the observation given all history. From our simulation experiment, we observed that working with direct HTM output is not suitable for creation of the binary sequence $\{c_t, t = 1,2,..\}$. We create an anomaly score with the same in the SPRT phase discussed as follows: Let at time t_0, we have run HTM and we have output from HTM core as htm_value_t_0. Let corresponding

observed value is obs_val_t. Now consider a historical window of size w_{sz}. This window size is again a user driven or a data driven input which will depend on the business use case, SME and/or historic data. We compute the rolling standard deviation σ_{roll} from this data. For the t-th observation, anomaly score is defined as:

$$\text{anoml\_score} = \frac{\text{absolute(htm\_value\_t} - \text{obs\_val\_t)}}{k * \sigma\_\text{roll}} \quad (3)$$

Thus, the final HTM output is obtained from anoml_score as follows:

$$\begin{aligned} htm_t &= \text{anoml\_score if anoml\_score} < 1 \\ &= 1 \text{ if new\_anoml\_score} > 1 \end{aligned} \quad (4)$$

We repeat this for all the data points in the time series and create the sequence $\{c_t, t=1,2..\}$ as Definition 2.

3.5 Algorithmic Steps for Data Drift Detection

Consolidating the discussion in section 3.1 and 3.2, we formally state the algorithm steps below:

Algorithmic steps for data drift identification

Input: Data on relevant dimension streamed as input, L (window length for htm output regularization), a, b, p_{null}, $p_{alt,}$ bin_threshold (to construct c_t), anomaly limit k.
Output: Starting time point of a possible data drift.

Execute the following step for the incoming data stream in continuum.

(i) Input data stream to HTM layer and collect output htm_t for t = 1,2,.. upto current time point.
(ii) Re-scale HTM output as per Equation (3) and (4) in section 3.4.
(iii) Compute c_t following Definition 2 and Cm_t following Definition 4.
(iv) Compute $Upper_{\_limit}$ and $Lower_{\_limit}$ following Equation (1) and (2) in Theorem 1.
(v) Compare Cm_t with $Upper_{\_limit}$ and $Lower_{\_limit}$ to either detect data drift or continue monitoring as per Theorem 1.

3.6 Experiment on Drift Detection with Simulated Data in the Unsupervised Scenario

To illustrate the concept, we consider a univariate scenario where data is generated from a normal distribution and the drift is present as a dynamically changing mean over time. We consider three basic patterns of data drift, namely, periodically change, slowly but monotonical change and abrupt changing in the mean. After running the drift detection algorithm, vertical lines in the data plot represent onset of a drift phase. The parameter sets corresponding to these experiments are mentioned in **Table 1** later.

Periodically time varying mean: The data is drawn from a normal distribution with a mean function with period 250 and amplitude 0.5. It is shown in **Figure 5**.

Monotonically time varying mean: In the below example, samples are generated from a normal distribution with monotonically changing mean over time. The changing mean pattern is determined by a 3-rd degree polynomial with randomly generated coefficient from a standard normal distribution. It is shown in **Figure 6**.

Abrupt mean shift: The third pattern considered is an abrupt data drift through a one-time increase in the gaussian mean by 2 units. This change is imparted at time point t = 250 and we can notice the distinctness in the data distribution before and after t = 250. This is shown in **Figure 7**.

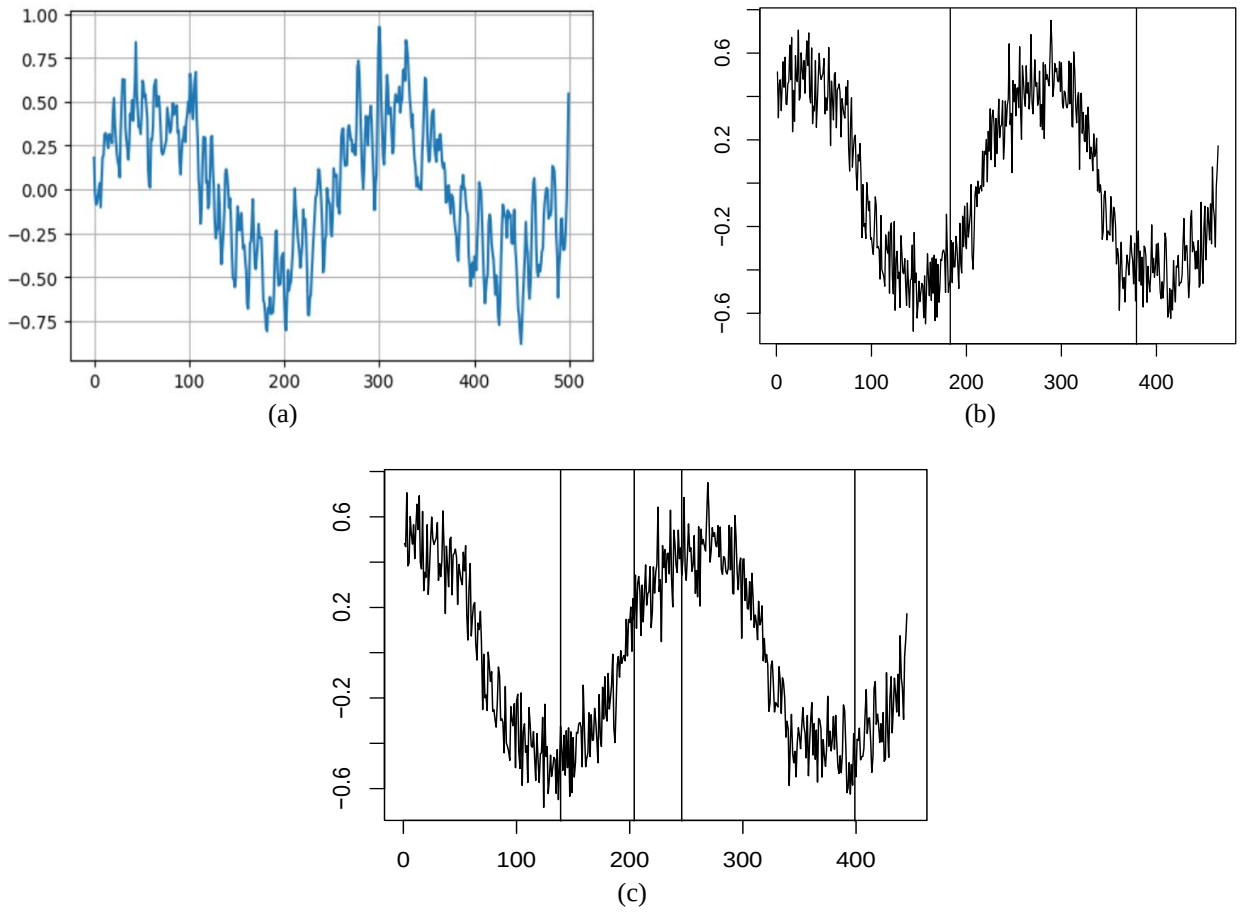


Figure 5. Time (X-axis) – Vs – value (Y-axis) Plot (a) Gaussian with periodically changing mean (b) Drift start time (vertical lines) with historical window size 25 (c) Drift start time (vertical lines) with historical window size 10.

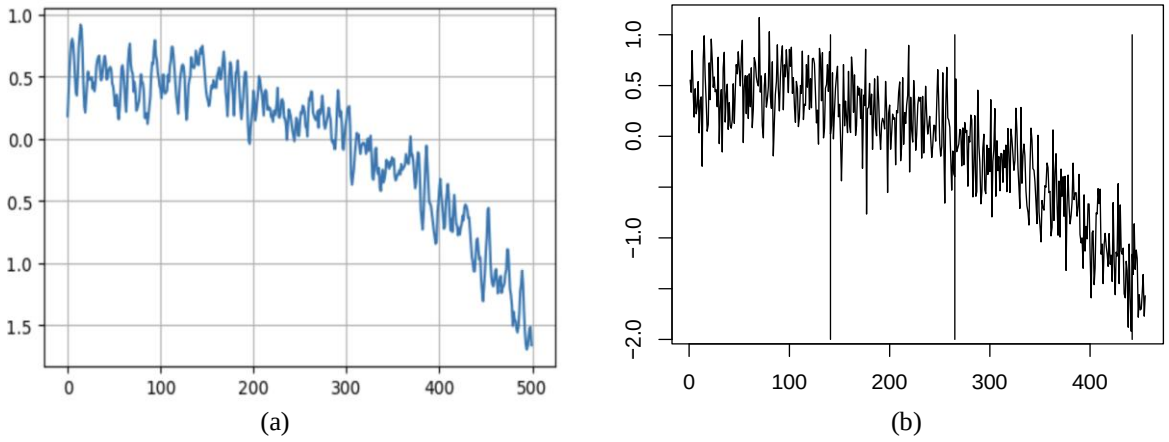


Figure 6. Time(X-axis) – Vs – value(Y-axis) plot (a) Plot of rescaled data with monotonically changing mean (b) Detected drift start time point (Vertical lines).

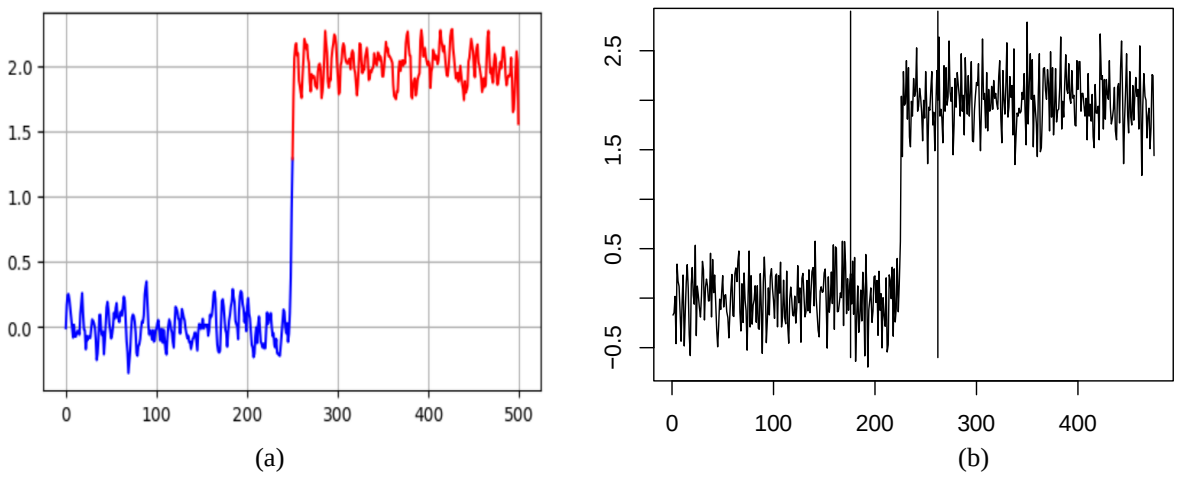


Figure 7. Time(X-axis) – Vs – value(Y-axis) plot (a) Plot of rescaled data with abrupt mean change (b) Detected drift start time point (Vertical lines).

3.7 Discussion of the Experimental Result of the Proposed Approach

The hyperparameters for identification of the three data drift scenario (Periodically, and Abrupt mean shift over time) following algorithm stated in section 3.5 are given in **Table 1** later.

It can be noted that apart from the abrupt data drift scenario, in the other two scenarios where the mean shifts more smoothly in a periodic or monotonic way, the definition of a data drift itself is very much subjective. Naturally, the drift detection aspect in these two cases is also relative to the definition of drift. In fact, it is quite intuitive that the definition of drift will depend on the variability of performance of the underlying model in two different data regimes. For example, if we consider **Figure 5(b)** and **5(c)**, the difference lies only in the historical window size for anomaly score calculation, which reflects quite two different scenarios of data drift. In **Figure 5(b)**, the data drift regions are visibly separated in the two half periods whereas in **Figure 5(c)**, the data drift is considered more locally as the window length is more and the change pattern in mean is comparatively slow. Hence, the underlying behavior of the model will determine which definition of drift is more likely.

A similar discussion can be anchored around the monotonic mean change or abrupt mean shift scenario, for which drift onset time periods are illustrated with the parameter values mentioned in **Table 1**. For data drift identification in the abruptly changing mean scenario, it can be understood that a comparatively lesser window length will not be able to identify the drift scenario, and a similar observation was made during the simulation experiment. However, this increases false drift discovery as we observed from **Figure 7**.

We also experimented with a scenario when there was no data drift and it actually did not detect any drift but the sensitivity is sharp around the historical window size. We could identify no drift situation by increasing the window size, starting from 15.

The proposed HTM-SPRT did work in a mixed way and to a good extent in an expected way. But some of the tuning parameters seems a bit more important than anticipated. For example, setting the null and alternative hypothesis values differently has lesser impact of drift detection than different historical window size. Another notable observation is the possible need of setting a high bar for type II error (smaller value

of b). When the cost of retraining applications is comparatively more than the cost of a drift detection failure, we would like to keep such type II error bound so as to limit multiple local drift detection within a short time span.

The parameters used in our experiments are listed in **Table 1**, as mentioned earlier.

Table 1. Parameters corresponding to different drift detection scenarios.

Simulation scene	Window length	Bin threshold	p_null	p_alt	a	b	Anomaly limit k (anomaly is beyond $k*\sigma_{roll}$)
Shock (**Figure 7**)	15	0.65	0.45	0.5	0.05	0.005	k= 1
Slow changing mean (**Figure 6**)	35	0.65	0.45	0.5	0.05	0.005	k= 1
Periodically changing mean (**Figure 5**)	25,45	0.65	0.45	0.5	0.05	0.005	k= 1

Comparison with a few competing approaches:
To conceive some meaningful comparison of the proposed approach with some relevant existing approaches, we have conducted an experiment. Kolmogorov-Smirnov (KS) test, Wasserstein distance and Population Stability Index (PSI) are three popular approaches of comparing two sets of data points with respect to the data distribution. However, to build an online drift detector we need to formulate an appropriate execution of the above algorithms. One such formulation, keeping a logical similarity with the proposed approach of HTM and SPRT ensemble, is given below.

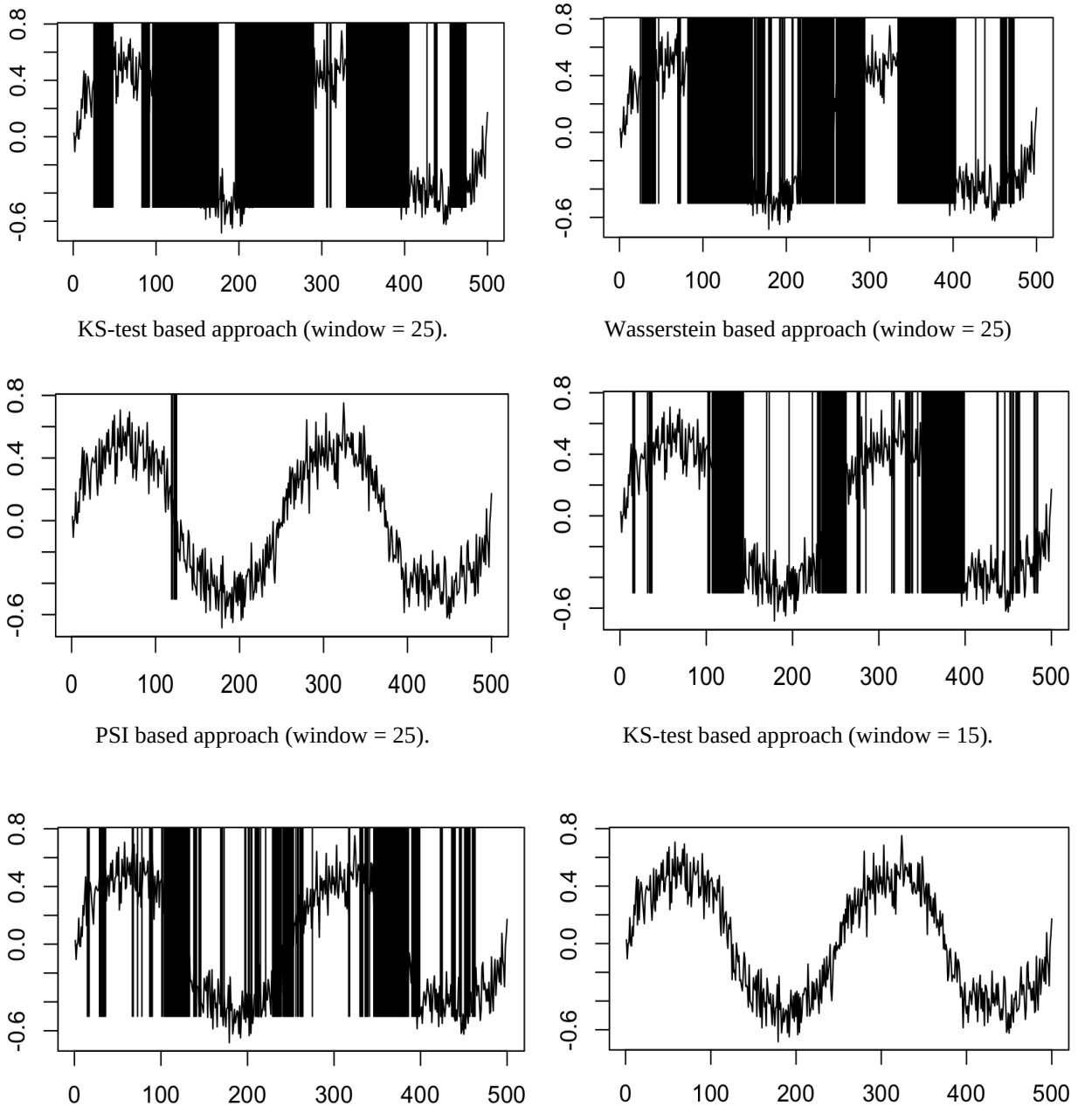
(i) Start with data from an initial window size [1, w] as the reference dataset. Consider data from the time window [w+1,2w] as the target data set.
(ii) Test the hypothesis that the reference and the target data set are from same probability distribution with the three algorithms stated above at the time point w+1.
(iii) Change the reference window to [1, w+1] and the target window [w+2, 2w+1] and conduct the same testing as in 2.
(iv) Continue till the end of the available data horizon, till the reference window is [500-2w,500-w] and the target data widow is [500-w+1, 500]

To compare with the proposed approach, we considered w = 15, 25 and plot identified drift start time points as a vertical line overlaid with the data (similarly as in the proposed approach). For testing no drift null hypothesis in step 2 mentioned immediately above, we use a two sided Kolmogorov-Smirnov (KS) test (ks.test() in base R package). For PSI, we compare the PSI score psi** with the Z-score obtained from function psi() from R package PDtoolkit (PDtoolkit, 2023) and reject the no drift null hypothesis if psi** > Z-score. For Wasserstein distance-based approach, we simulate the 95% percentile point from the distribution of Wasserstein distance computed from 500 iteration. In each iteration, Wasserstein distance was computed from two independent random sample of size 50 drawn from N(0,1) distribution. Note that under no data drift null hypothesis, this 95% percentile point corresponds to a threshold for testing the null hypothesis at level $\alpha = 0.05$.

A sample illustration is given in **Figure 8** below for the scenario with periodically varying mean, and in **Figure 9** below for abrupt mean shift.

It can be noticed that for the periodically varying mean scenario, the Kolmogorov-Smirnov (KS) and Wasserstein based approach raises alarmingly high alert of drift detection whereas PSI based approach raises very few drift detection alerts. In comparison, our proposed approach is much more balanced. For the abrupt mean shift scenario, the performance of the alternatives shows similar behavior of high number

of alerts in consecutive time points across a time window. A similar behavior of the competing approach is observed in the slowly moving mean change scenario. From the above discussions, we can note that the alternative approaches are practically not applicable due to very high false positive alarm whereas our proposed approach shows a much better promise in terms of performance and practical usability.



KS-test based approach (window = 25).

Wasserstein based approach (window = 25)

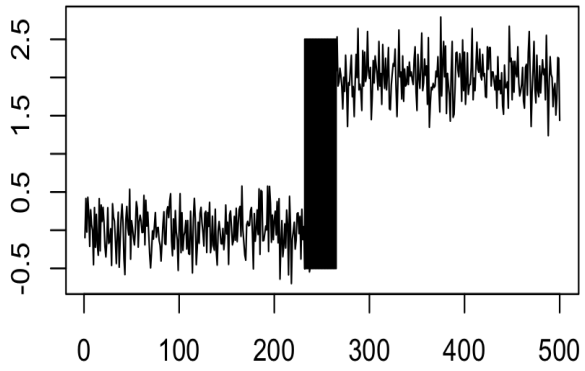
PSI based approach (window = 25).

KS-test based approach (window = 15).

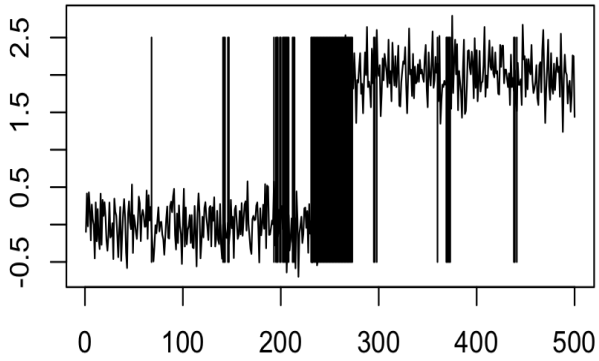
Wasserstein based approach (window = 15).

PSI based approach (window = 15).

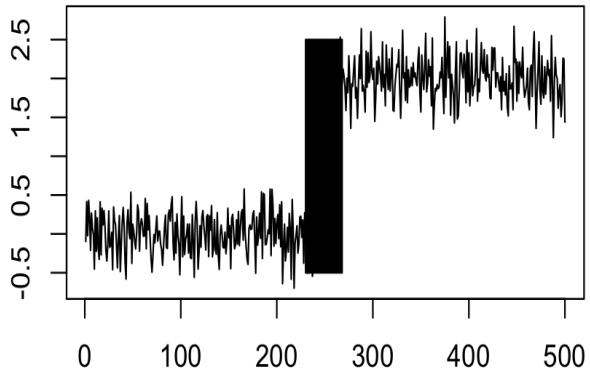
Figure 8. Time (X-axis) – Vs – value (Y-axis) Plot from three competing approaches in the periodically varying mean scenario.



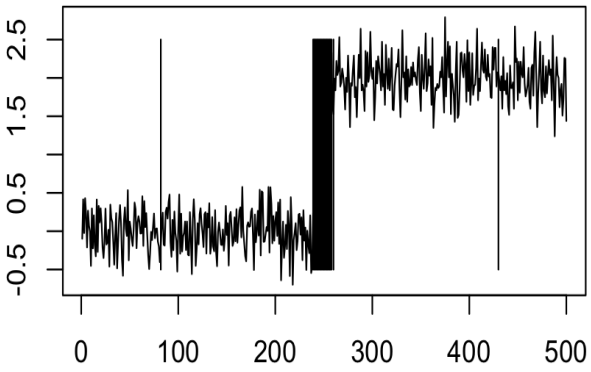
KS-test based approach (window = 25).



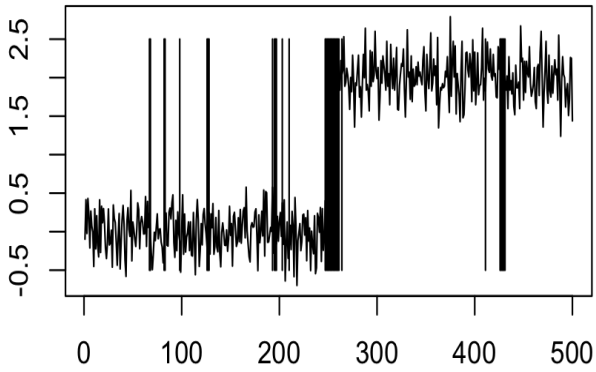
Wasserstein based approach (window = 25).



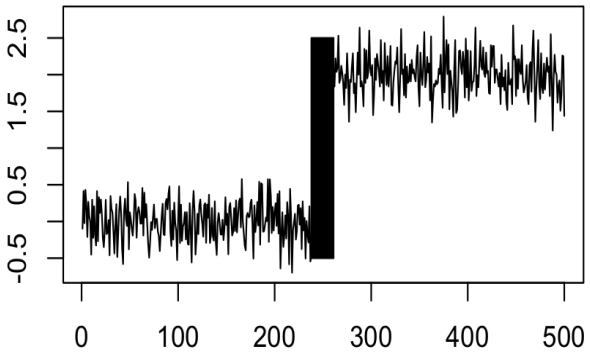
PSI based approach (window = 25).



KS-test based approach (window = 15).



Wasserstein based approach (window = 15).



PSI based approach (window = 15).

Figure 9. Time (X-axis) – Vs – value (Y-axis) Plot from three competing approaches in the abrupt mean shift scenario.

4. Data Drift Detection in Supervised Scenario: Approach for Multivariate Data

Currently, HTMs work mainly with univariate data. However, in the real-world use cases, we may have labelled multidimensional data, which the current HTM cannot handle. To extend HTM application into multivariate data setup, we propose to leverage neural network as explained in the following subsections.
In a supervised scenario when labelled data (e.g., data with outlier tag) is available, we propose an extension of univariate HTM based approached to multivariate setup through a neural network combiner. In this approach we first fit an individual HTM model for marginal dimensions. Next, a neural network is trained to classify observations as outlier where the input layer of the network consumes the output of the fitted HTM model. For loss computation, labels of the original data are utilized.

4.1 Application of HTM in the Multivariate Scenario

As a motivating example, let us consider telecom domain data where different features like latency, throughput and user load etc. from a telecom network are available at the cell level (specific locality, for example). These features typically represent different aspects of the network behavior. The anomalous behavior of a network can originate typically from a combination of multiple such aspects. Hence, anomaly from a multidimensional view should be the natural way to detect anomalous behavior of the network at a given time point. Independent feature level view may not correctly identify the anomalous state. For example, the latency and throughput marginally may not be anomalous but put together, it can detect an unusual scenario where for the specific throughput, the observed latency may not be a normal behavior of the network.

To extend the efficient anomaly detection capability of HTM from univariate setup into a multivariate scenario, we propose to utilize a neural network (NN) to combine the univariate HTM output and detect anomaly from a multivariate perspective. In the following subsection we discuss the detailed methodology.

4.2 Neural Network to Combine the Output of Multiple HTM Cells for Multivariate Data

The inputs and outputs of the neural network are given as follows:

- Corresponding to each data dimension, we fit one HTM cell. Output of HTM cells put together for a specific time point is inputted to the NN as one data point with same time label. Note that data other than HTM output can be combined directly with the HTM output form the input data to the NN.
- The output of the neural network is a binary number representing whether the combination of inputs at a given time represents an actual anomaly. We train the neural network using the ground truth labelled (normal/anomalous) data.
- A trained neural network can predict anomalous observation from test data based on similar input features mentioned above.

4.3 Anomaly Detection using Existing Models as the Ground Truth of the Anomalies

The primary challenge for us was to create the outlier tags, as limited telecom domain expert support was available. We handled this by applying the multiple outlier detection (OD) algorithm on the data and tagging an observation as outlier if 80% agreement is observed from the multiple algorithms. We used PyOD package (Zhao et al., 2019) for this benchmarking, where twelve OD algorithms were chosen suitably. The data preprocessing, however, was a lengthier step where linear, first and second order interaction was considered for feature generation. Subsequent PCA for dimension reduction was executed. In the end we worked with 54 PCA dimensions where the initial data was of 295 dimensions.

A brief overview of these algorithms from PyOD package that were used for anomaly benchmarking are given below:

- Linear Models for Outlier Detection: This includes the following.
  - PCA: Principal Component Analysis use the sum of weighted projected distances to the eigenvector hyperplane as the outlier scores)
  - MCD: Minimum Covariance Determinant (uses the Mahalanobis distances as the outlier scores)
- OCSVM: One-Class Support Vector Machines
- Proximity-Based Outlier Detection Models: This includes the following.
  - LOF: Local Outlier Factor
  - CBLOF: Clustering-Based Local Outlier Factor
  - kNN: k Nearest Neighbors (use the distance to the kth nearest neighbor as the outlier score)
  - Median kNN Outlier Detection (use the median distance to k nearest neighbors as the outlier score)
  - HBOS: Histogram-based Outlier Score
- Probabilistic Models for Outlier Detection:
  - ABOD: Angle-Based Outlier Detection
- Outlier Ensembles and Combination Frameworks: This includes the following.
  - Isolation Forest
  - Feature Bagging
  - LSCP

4.4 Application of HTMs to Identify Anomalies in a Sample Dataset of KPIs

The applications include the following steps.

(i) Generating HTM output: Once the 54 columns had been generated, we used HTM on each of the columns to identify the anomalies in each. We treated each of the columns as a HTM input stream. After applying the HTM algorithm to identify the anomalies in each column, we resulted in a list of anomalies (0 or 1).

(ii) Training the neural network (NN): We then trained a neural network of two hidden layers, one input and one output layer and optimized it with Keras Optimizer with 120 as epoch as the stopping rule. The HTM outputs of the 54 columns was the input to the neural network. The output was a binary value or either 0 (indicating absence of anomaly) or 1 (denoting presence of an anomaly).

(iii) Performance of the NN combiner: The NN combiner did a decent job of 90% outlier detection.

5. Conclusion

In this paper we have explored approaches to use HTMs in the unsupervised and supervised scenarios for anomaly detection in combination with SPRT and neural network respectively. For the supervised scenario, we used a method to transform univariate HTM based anomaly detection approach to multivariate anomaly detection via combining marginal HTM anomaly detection probability through a multilayered neural network. Our approach is experimented on a sample multivariate time series data consisting of Key Performance Indicators or KPIs in a telecom dataset. This simple combination shows a reasonable accuracy in detecting anomalous events. For the unsupervised scenario, we have used HTMs in combination with a statistical technique known as SPRT to detect data drift. The advantage of an HTM based approach lies in the instantaneous use where expensive training phase is not necessary, and anomaly detection can efficiently take place on streaming/real time/near time data scenario which is specifically suitable for many real-life use cases. We have observed that the proposed approach is capable to identify data drift in a streaming data with strongly significant lower number of false alarms than existing approaches based on familiar population homogeneity test (Kolmogorov-Smirnov (KS) test, Wasserstein distance and Population Stability Index based approaches). We can apply the proposed method as a standalone application as well as a part of a machine learning (ML) pipeline. Inclusion of this approach in a ML pipeline can make the ML application more efficient by raising an alarm of a data drift start period and data

drift end period. It can be noted that after a data drift start is identified, the data drift end period can be identified as the next data drift start signal.

6. Limitations and Future Work

The proposed approach is dependent on multiple hyperparameter specifications which are either to be supplied as a derivative of domain knowledge, or to be estimated through elaborate simulation based on enough historical data. Efficient choice of hyperparameters is important to balance a strike between recall and precision. Though we have proposed a few basic parameter combinations from a primary simulation experiment, this should come ideally as a part of tuning stage. For the supervised scenario, the objective function is quite trivial, but in the unsupervised scenario, specifically in a multidimensional set up, we need to pursue a deeper study to understand data driven optimality criteria and hence set a proper data driven hyper-parameter tuning.

There are a couple of gaps that we would like to fill in further continuation of this study as mentioned below. *Hyper-parameter tuning:* A possible direction may be to consider relative population stability in different drifted data regime while identifying most unstable transition phase as an indicator of data drift. In the one-dimensional scenario, an early experiment with data variance as a measure of stability seems to work reasonably. Corresponding multivariate extension is needed to be studied in detail. *Multivariate extension:* Multivariate extension of the HTM from first principles and combine it with existing multivariate SPRT is another interesting direction to explore. *Composite Hypothesis in SPRT:* It can be noted that data drift detection is mapped with simple hypothesis construction via HTM and testing via SPRT. Ideally, we should consider both null and alternative hypotheses in composite status through ranges of the parameter in Bernoulli distribution corresponding to a drifted or not drifted data scenario. Hence the right variation of SPRT can be explored for the testing exercise.

Conflict of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

AI Disclosure

During the preparation of this work the author(s) used generative AI in order to improve the language of the article. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors would like to thank the editor and anonymous reviewers for their comments that helped to improve the quality of this work.

References

Ahmad, S., & Purdy, S. (2016). Real-time anomaly detection for streaming analytics. *arXiv Preprint*, arXiv:1607.02480.

Anandharaj, A., & Sivakumar, P.B. (2019). Anomaly detection in time series data using hierarchical temporal memory model. In *2019 3rd International conference on Electronics, Communication and Aerospace Technology* (pp. 1287-1292). IEEE. Coimbatore, India. https://doi.org/10.1109/ICECA.2019.8821980.

Barua, A., Muthirayan, D., Khargonekar, P.P., & Al Faruque, M.A. (2020). Hierarchical temporal memory based machine learning for real-time, unsupervised anomaly detection in smart grid: WiP abstract. In *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems* (pp. 188-189). IEEE. Sydney, Australia.

Cover, T.M., & Thomas, J.A. (2006). *Elements of information theory* (2nd ed.). Wiley, New Jersey, USA.

Endres, M., & Schindelin, J.E. (2003). A new metric for probability distributions. *IEEE Transactions on Information Theory*, *49*(3), 1858-1860. https://doi.org/10.1109/TIT.2003.811249.

Gemaque, R.N., Costa, A.F.J., Giusti, R., & Dos Santos, E.M. (2020). An overview of unsupervised drift detection methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *10*(6), e1381. https://doi.org/10.1002/widm.1381.

Hawkins, J., Lewis, M., Klukas, M., Purdy, S., & Ahmad, S. (2019). A framework for intelligence and cortical function based on grid cells in the neocortex. *Frontiers in Neural Circuits*, *12*, 431889. https://doi.org/10.3389/fncir.2018.00121.

Hebb, D.O. (2005). *The organization of behavior: a neuropsychological theory*. Psychology Press, New York.

Kadam, S.V. (2019). A survey on classification of concept drift with stream data. *HAL Archives*. https://hal.science/hal-02062610/file/A_survey_on_classification_of_concept_drift_with_stream_data.pdf.

Marshall, A.W., & Olkin, I. (1960). Multivariate Chebyshev inequalities. *The Annals of Mathematical Statistics*, *31*(4), 1001-1014.

Numenta (2019a). *HTM white paper*. Retrieved from https://numenta.com/neuroscience-research/research-publications/papers/hierarchical-temporal-memory-white-paper/.

Numenta (2019b). *The science of anomaly detection: How HTM enables anomaly detection in streaming data*. Retrieved from https://www.grokstream.com/wp-content/uploads/dae-uploads/Numenta-White-Paper-Science-of-Anomaly-Detection.pdf.

Otahal, M., Keeney, D., McDougall, D. (2019). *HTM-core, Github*. https://github.com/htm-community/htm.core.

PDtoolkit (2023) Collection of tools for pd rating model development and validation. https://cran.r-project.org/web/packages/PDtoolkit/index.html.

Pesaranghader, A., Viktor, H.L., & Paquet, E. (2018). McDiarmid drift detection methods for evolving data streams. In *2018 International Joint Conference on Neural Networks* (pp. 1-9). IEEE. Rio de Janeiro, Brazil.

Piegorsch, W.W., & Padgett, W.J. (2011). Sequential probability ratio test. In Lovric, M. (ed) *International Encyclopedia of Statistical Science* (pp. 510-515). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04898-2_510.

Schoonewelle, H., Van der Hagen, T.H.J.J., & Hoogenboom, J.E. (1995). Theoretical and numerical investigations into the SPRT method for anomaly detection. *Annals of Nuclear Energy*, *22*(11), 731-742. https://doi.org/10.1016/0306-4549(95)00005-Y.

Souza, V.M., Chowdhury, F.A., & Mueen, A. (2020). Unsupervised drift detection on high-speed data streams. In *2020 IEEE International Conference on Big Data* (pp. 102-111). IEEE. Atlanta, GA, USA. https://doi.org/10.1109/BigData50022.2020.9377880.

Sutskever, I., Vinyals, O., & Le, Q.V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, *27*, 3104-3112.

Vallender, S.S. (1974). Calculation of the Wasserstein distance between probability distributions on the line. *Theory of Probability & its Applications*, *18*(4), 784-786. https://doi.org/10.1137/1118101.

Wadewale, K., Desai, S. (2015). Survey on methods of drift detection and classification for time-varying data sets. *International Research Journal of Engineering and Technology*, *2*(9), 709-713.

Wang, H., & Abraham, Z. (2015). Concept drift detection for streaming data. In *2015 International Joint Conference on Neural Networks* (pp. 1-9). IEEE, Killarney. https://doi.org/10.1109/IJCNN.2015.7280722.

Wu, J., Zeng, W., & Yan, F. (2018). Hierarchical temporal memory method for time-series-based anomaly detection. *Neurocomputing*, *273*, 535-546. https://doi.org/10.1016/j.neucom.2017.08.049.

Yurdakul, B. (2018). Statistical properties of population stability index. *Western Michigan University Dissertations*. Retrieved from https://scholarworks.wmich.edu/dissertations/3208/.

Zhao, Y., Nasrullah, Z., & Li, Z. (2019). *PYOD, Github*. https://github.com/yzhao062/pyod.