Journal of Engineering and Technology for Industrial Applications

ITEGAM-JETIA

Manaus, v.10 n.49, p. 18-25. September/October., 2024.
DOI: https://doi.org/10.5935/jetia.v10i49.1132

ISSN ONLINE: 2447-0228

RESEARCH ARTICLE OPEN ACCESS

# AN ONLINE INCREMENTAL ADAPTATION MECHANISM TO SUBDUE THE EFFECT OF DRIFT IN STREAMING DATA

**Ushashree P[1], R B V Subramanyam[2]**

[1] Department of Computer Science and Engineering, National Institute of Technology, Warangal, India
[2] Professor, Department of Computer Science and Engineering, National Institute of Technology, Warangal, India.

[1] http://orcid.org/0000-0001-5154-9632, [2] http://orcid.org/0009-0005-8907-1984

Email: up720075@student.nitw.ac.in , rbvs66@nitw.ac.in

**ARTICLE INFO**

***Article History***
Received: May 24th, 2024
Revised: September 06th, 2024
Accepted: September 16th, 2024
Published: October 04th,2024.

***Keywords:***
Adaptive learning,
Concept drift,
Model retraining,
Online learning,
Streaming data.

**ABSTRACT**

Concept drift detection and adaptation is one of the crucial components of a resilient machine learning pipeline in production. The Adaboost is an ensemble approach that incorporates incremental learning, that is widely used for concept drift adaptation in streaming data. It is generally combined with other methods such as ARF classifiers and Bagging Classifiers. This study presents a collection of online incremental learning algorithms for streaming data to adjust machine learning categorization when there is concept drift. Better results are obtained on the Australian power dataset, demonstrating the efficacy of our approach in comparison to the current benchmark.



## I. INTRODUCTION

Concept drift in machine learning system, is a statistical property of the models where the target variable distribution which the machine learning model tries to predict changes over time, thereby making the model's effectiveness limited or severely impacted. The predictions from such models become less accurate with time. Almost all machine learning models face this challenge in production and thus it's a usual part and parcel of a machine learning lifecycle. It is also probably the most common reason why ML model needs to be refreshed and retrained periodically. Concept drift arises because usually the incoming data distribution changes over time and thus it drifts away from the historical data that was used during training, it may so happen that the relationships and correlations among features may also change. Thus, a shift in the distribution P(y|X), where y is the real target label and X is the input feature vector, could be used to characterize the phenomenon of concept drift.

Concept drift can further be understood with an example. Suppose a classification model is trained to detect any unusual network access request to a server in the field of cyber security. When this model was trained,1,000 requests a minute was an extremely large number of requests that could indicate a malicious attack. But later due to business expansion or marketing campaign, the website became very popular and now receiving 1000s of request is no big deal. Thus, an update to the model is required to take into account the change in data distribution and then only it can perform at an optimal level. Detecting concept drift as soon as possible is essential for maintaining models' performance in production.

To understand concept drift, it is essential to understand the data drift and model decay. The shift in the distribution of data over time is known as data drift. Concept drift can have many types based on its pattern.

Depending on the pattern, drifts can be divided into several types:

Gradual Concept Drift is the most prevalent kind of concept drift that can happen as a result of change in nature of business, launch of new line of business, tools or data collection approaches or deprecating legacy systems and schema with new ones. Such drifts are hard to notice as its impact is only visible after a long-time gap unless the systems are specially trained to notice small changes in the dataset every now and then.

Recurring Concept Drift occurs due to seasonal change in business processes. But not always such changes can be picked up

by a time series model even when seasonality is considered during model building. Customer churn, new competitors, market fluctuation etc. can cause this kind of drift.

Instantaneous Concept Drift are caused by abrupt events that could be outliers for example COVID-19 pandemic impacted all businesses and models were tuned to consider the new reality. When the performance of the model such as accuracy, F1-Score, R squared etc. starts deteriorating as time is passing, it is called model decay. When it goes down below a threshold, the model needs to be re-trained on a re-labeled dataset. Model decay occurs for one of the following reasons.

Covariate Shift: when a shift occurs in the independent feature variables.

Prior Probability Shift: when shift occurs in the dependent target variable.

Concept Drift: when shift occurs in the relationship itself between the independent feature vector and the dependent target variable.

A novel approach to handle the issue of concept drift in machine learning system is proposed in this paper which demonstrates the superior performance over concept drift detection on benchmark dataset. The developed method is a unique ensemble of incremental learning algorithms to adapt concept drift in streaming data applications. The obtained result surpasses the current best result on the Australian Electricity dataset, which is commonly used as the benchmark dataset for studying the concept drift problem in literature.

The paper is segmented into the subsequent sections. Section II provides an overview of the different tasks that were performed towards identification and handling of concept drift, Section III describes the dataset used in this study and focus on our methodology and algorithms to handle concept drift, Section IV shows result and comparison with current benchmarks and finally Section V provides a conclusion.

## II. THEORETICAL REFERENCE

In this section, a comprehensive review on the various work done in concept drift detection as well as adaptation is provided along with the summary of advantages and disadvantages for each method implemented.

For [1] provides a comprehensive survey on concept drift, its types, detection and adaption. It describes the problem of drift specifically in the machine learning environment and its challenges, especially classification of streaming data. The authors argue how in classification problems, a variation in the concept or distribution of dataset over any time period is defined as concept drift and how the performance of such models degrade even in stationary data. They thus point out that handling this problem is more challenging in data streams particularly. They begin with categorization of current streaming data classification algorithms along with benchmark results and their capability to solve the issue of concept drift. They also provide a comparison of various tools available. They further list down the benchmark datasets and performance metrics used in literature. It therefore serves a roadmap for any new researcher working in the domain of concept drift for streaming data classification.

Accurate Concept Drift Detection Method (ACDDM) is a new framework proposed by [2] for concept drift detection. Abilities to identify conceptual deviations in changing data streams. The status of the previously calculated error rate was initially evaluated by calculating a term called Hoeffding's inequality. It measures the probability of error. If the current error of the learner base differs from the calculated error of the Hoeffding inequality, concept drift can occur in many cases, leading to its occurrence. However, this method only detects drift without adaptation.

According to [3] discuss an Incremental learning framework that helps it to learn the correct classification for the future/further unseen data points from the historic streaming data. In this paper, the authors proposed a well-known idea of an ensemble learning method based on incremental learning to handle both class imbalance and concept drift too. Their work focuses only on the concept drift as class imbalance is not of much interest to them. They handle concept drift by a dynamic cost-sensitive weighting scheme which helps the classifier weight each data point according to their classification model's performance and sensitivity. Authors apply the proposed method on Australia's electricity pricing to predict if the price will go up or down compared to that of another city Victorias in a given 24-hour time period. The authors argue that their method beat the current benchmark on the given dataset.

Machine learning algorithms [4] are used to extract knowledge from real-time data, which is typically stored in a static database and processed in batches. They handle the changing patterns of data. Additionally, the ideas may alter as time passes. In the streaming environment, these elements should be given priority. According to [5] argues that previous methods quoted so far for concept drift detection first detect the time and positions of the drift occurrence and then only tires to adapt it by modelling the change in distribution of the data. They mention that such an approach is unlikely to work when underlying factors for change are predictable, thereby making the model miss any future concept drift trend of the streaming data. Authors say that such cases have not been fully explored in previous works which they have included in their novel method called DDG-DA1. The authors contend that the novel approach can efficiently and methodically predict the source of data distribution and enhance the efficacy of models that are susceptible to concept drift. They made it possible by first training a predictor which can estimate the future data distribution, then once the data distribution is estimated they generate a training sample and train a new model on the generated data. They test their methods on real world datasets such as electricity data or stock price data. Electricity load data and solar irradiance data are common and obtain benchmark result on all these.

For [6] handles real world data and existing concept drift. The major focus is on decision-based problems in real world environment. It mainly deals with the problem of the concept drift, specifically about the time, type and pattern of the drift in a non-stationary environment. According to [7] present a comprehensive survey that classifies different concept drift detectors based on their main features, drawbacks, and benefits. They concluded by proposing areas for further investigation, difficulties encountered in research, and the direction of future studies.

Provides a tool for quantitative measurement and description of concept drift by computing marginal distributions of variables. Such quantitative drift analysis techniques lay the foundation of communicating the drift in terms of Bayesian and marginal probabilities. Authors provide their results on three benchmark datasets and thus demonstrate the effectiveness of quantitative drift measurement techniques on real-world learning tasks [8]. [9] Handles concept drift by relying on computation and analysis of the empirical loss of online learning systems or algorithms. Their method is developed based on obtaining statistical parameters from data distribution of loss by shuffling and

re utilizing the data several times through resampling. Additionally, they provide a theoretical guarantee for the designed procedure—an upper bound—based on the performance and stability of the underlying learning algorithms. According to the results of their experiments, their method performed well even in the presence of gaussian noise and had very high recall and precision values [9].

Provides a detailed survey on adaptive learning process for supervised learning. Authors first create a well-defined category of existing strategies implemented for concept drift detection and adaptation, then they provide an overview of the most well-known and popular techniques as well as algorithms. They also discuss the evaluation strategy for such adaptive algorithms under the presence of concept drift and present several illustrative examples and applications. Their survey covers the different facets of concept drift along with types, algorithms examples, applications, advantages, and disadvantages in an integrative way to reflect on the existing work done in this direction [10].

Most research work in [11-13] has the limitations of data streams—such as their infinite length, concept change, concept evolution, and concept recurrence—are the focus of most research endeavors. Concept drift detectors are used in many different applications, such as churn prediction for mobile companies and theft detection in the energy distribution system. Despite this, these algorithms are unreliable when handling dirty data.

According to [14] proposed a novel approach titled Optimum-path Forest classifier which is used for handling concept drift based on the decision of the OPF classifiers committee. For [15] experimentally assesses the prequential methodology by examining its three commonly employed ways for updating the prediction model: Basic Window, Sliding Window, and Fading Factors. The main objective is to determine the most precise variant for experimentally evaluating prior results in situations when idea drifts occur. The focus is mostly on the accuracy observed within the entire data flow.

For [16] discussed the performance of detection and introduced an algorithm that integrates both False positive rate and the error rate. It is called Drift Detection Method with False Positive rate for multi-label classification (DDM-FP-M). It initially calculates false positive rate and then interlaced with the Drift Detection method. Method integrates the disagreement measure, a diversity measure commonly used in static learning, with the Page-Hinkley test to detect drifts in streaming scenarios [17]. Through the analysis of both artificial and real-life situations [18] have seen that each data stream may necessitate a distinct measurement function in order to detect changes in concepts, Considering the particular features of the respective application field.

The online sequential extreme learning machines method is validated using two synthetic case studies that involve various types of idea drift. For [19] method is applied to two publicly available real-world datasets. According to [20] discussed an ensemble methodology to identify a collection of highly reliable predictions using clustering algorithms and classifier predictions. Then they used the Kullback-Leibler (KL) divergence method to quantify the disparities in distribution between consecutive segments, with the aim of identifying any changes in the underlying notion. Assessed the effectiveness of single-variable change detection methods. These techniques are applied to ensembles, in which every member scans a certain feature in the input space of an unsupervised problem detecting changes. An extensive evaluation of the ensemble combinations was given [21].

## III. MATERIALS AND METHODS

### III.1 DATASET

The Electricity dataset [22] used in this study was obtained from the Australia's New South Wales Electricity Market. The dataset consists of 45,312 records and 10 attributes. The target variable in this dataset represents the price change of electricity compared to the moving average of its demand over the previous 24-hour period. The dataset exhibits a significant complexity and has been extensively studied in the literature, with numerous benchmark results given. Consequently, this dataset is an ideal choice for implementing novel concept drift models, including our own.

The variables of this dataset are nswdemand, nswprice, vicdemand, vicprice, and transfer price It gives the electricity demand and price for Victoria and New South Wales, two Australian states. There is a measurement of the power transferred between these two states. The transfer price change as compared to a moving average of the electricity demand during the preceding 24 hours is shown by the target label. The datapoints are generated at an interval of 30 minutes. The raw values are normalized after data collection in the interval of [0,1]. Two columns namely ID and date are dropped as those are not helpful towards our goal and therefore, 8 attributes are left in the training and test datasets.

### III.2 PREPROCESSING

Standard data preprocessing steps are implemented to measure the quality of data. There are no missing values, and all numeric columns are already scaled in the range of [0,1]. Correlation among all variables is computed to determine whether there is a strong correlation between them. The correlation plot is shown in Figure 1.

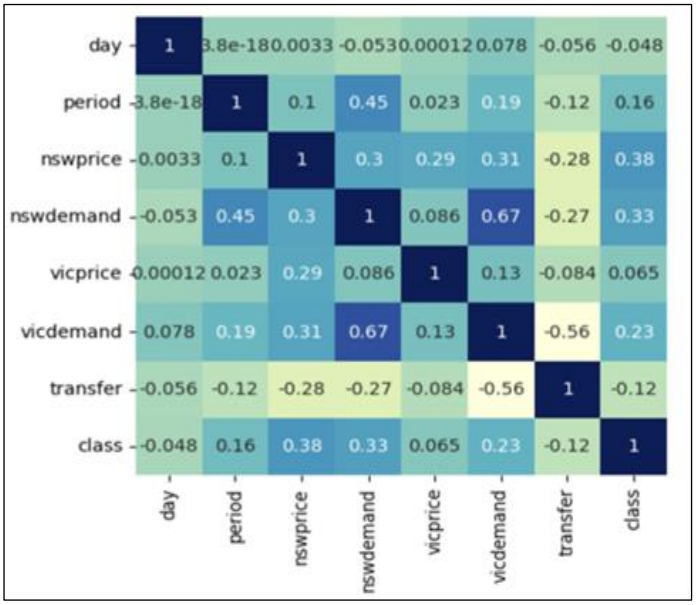


Figure 1: Correlation among variables in electricity dataset.
Source: Authors, (2024).

Feature engineering is implemented to compute new derived features from the original 8 features of this dataset. Polynomial feature selection with degree 2 and 3 are done to generate 55 and 120 features respectively. During the modelling phase, different versions of models are trained with original

features as well as derived feature sets of 55 and 120 and results are compared. The dataset is also described in Table 1.

Table 1: Dimensions of the electricity dataset.

| | Dataset shape | Training set shape | Test set shape |
|---|---|---|---|
| Original features | (45312,8) | (31718,8) | (13594,8) |
| derived features (Degree 2) | (45312,55) | (31718,55) | (13594,55) |
| derived features (Degree 3) | (45312,120) | (31718,120) | (13594,120) |

Source: Authors, (2024).

## III. 3 MODELS

There are multiple ways to handle concept drift in machine learning systems. For batch processing, models can be retrained with new instances and relabeled target variable, when concept drift is observed. For streaming data, incremental learning is usually the most preferred solution. The ideal solution for handling concept drift is thus to quickly adapt to concept drift. As most real-life production level machine learning applications run on streaming data, online learning to handle concept drift is ideal and the same has been explored by various work done so far. Our work develops a unique mixture of online learning algorithms which beats the current best obtained result on Austrian electricity dataset for concept drift adaption. Concept drift can be formally described as follows.

When two-time instances t and t + 1 experience concept drift, it is described as

$$X: Pt(X, y) \neq Pt+1(X, y) \quad (1)$$

In Eq.1 Pt denotes the joint probability distribution of data at time t and Pt+1 denotes the distribution at time t+1. Concept drift has occurred if the two distributions are not equal.

In the realm of Bayesian Probability [2], the classification of data point is done using the class label posterior probabilities, where each class y's posterior probability can be expressed in terms of X as in Eq.2

$$P(y|X) = P(y)\, P(X|y) / P(X) \quad (2)$$

where P(y) is the known prior probability of the class y, P(X|y) is the marginal probability of X given class y and

$$P(X) = \sum_{i=1}^{n} P(y)P(X|y) \quad (3)$$

Bayesian theory shows that the concept drifts can have two types:

Real concept drift: This kind of drift indicates a shift in the model's performance. i.e change in the class labels posterior probabilities.

Virtual drift: This kind of drift refers to a shift in the underlying distribution of the features while the performance of the model remains unchanged. i.e., without changing the conditional probability, the P(X) input probability distribution changes.

### III.3.1 Incremental Learning

Incremental learning is an approach where each data point is sent successively to train the machine learning model. It is the standard approach for streaming data where the entire batch of data cannot be used in one shot for training, so model is made to learn in an incremental fashion with model weights getting updated as and when new data samples arrive for training. At any given time, step t, the historical data can be described as:

$$XH = (X1, X2, X3 \ldots. Xt) \quad (4)$$

So, for next instance t+1, to predict a label yt+1 using data till Xt+1 a learner Lt is trained using either all the data points or a sample from the given data XH then the learner Lt is used to predict the label for the data point Xt+1. The identical procedure is recurred for the next data point where predicted yt+1 from the classification model is used as input along with Xt+1. So, Xt+1 and Yt+1 becomes a part of historical data. Figure 2 shows a pictorial representation of incremental learning. The incremental learning algorithms explored and adapted in this work are described here:

Adaboost classifier [23] is a boosting ensemble method also popular for batch modelling. In case of AdaBoost, when a new observation arrives, the model learns from it k times. Initially weights are randomly initialized and updated based on misclassification error. The value of k is calculated from a Poisson distribution of model parameters. The parameters or weights are updated when the weak learners fit on the data successively. AdaBoost classifiers are known to perform well in case of concept drift under streaming data due to its nature of learning the weights from misclassification from individual data points. It also handles biasness which is a common pattern in incremental learning.

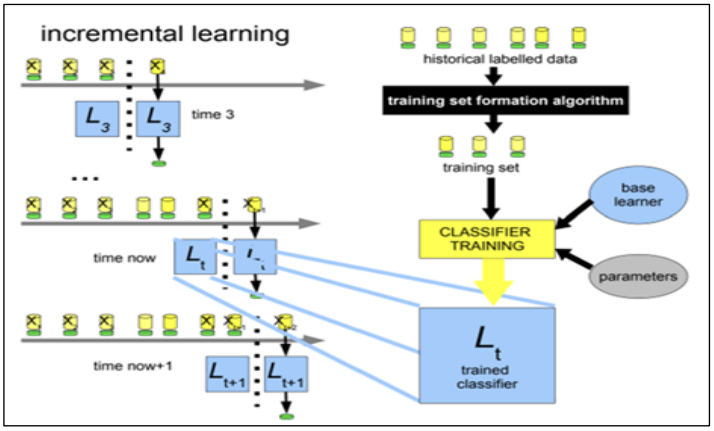


Figure 2: Incremental Learning process.
Source: [24].

Leveraging Bagging [25] is a bagging technique to handle high variance in incoming data streams. It is an improvement over the classic Oza Bagging classification model. The performance of bagging is leveraged or improved by increasing its sampling frequency or the resampling rate. It also uses a Poisson probability distribution mechanism to arrive at the re-sampling process. A higher weight value of the Poisson distribution considers the high variance in training data and thus different range of weights are updated accordingly to accurately classify the data samples.

The ADWIN algorithm is used by Leveraging Bagging techniques to manage concept drift. It keeps track of each classifier's performance within the ensemble and replaces underperforming classifiers with new dummy classifiers when

concept drift is identified. In the next stage, the dummy classifier's weights are changed once more. Recent efforts have achieved state-of-the-art performance on numerous standard data sets by leveraging bagging classifiers, as detailed in the results section.

Logistic regression [26] is a classical method for batch learning as well as online learning. This also supports learning with mini batches of data. For many ensemble methods in incremental learning such as Leveraging bagging classifier or AdaBoost classifier, it is used as a base learner. The working principle remains the same even for the case of single instance learning mechanism. Adaptive Random Forest classifier [27] is another very important method for incremental learning. It is popular for its ability to induct variance with replacement, and randomly selecting feature subset based on entropy of splits of nodes and its ability to do drift detection for base trees.

Adaptive random forest work on the principle of training background trees when a drift detection warning is generated, and it replaces the active tree in case warning escalates to a drift i.e. trees are generated on the fly when drift is observed and thus model is updated with new trees which learns from the pattern of data points which caused the drift. Thus, it adapts the drift detection mechanism. ADWIN Bagging [28] another bagging method based on Oza bagging classification model. ADWIN can be used as drift detector as well as drift adaptation methods. Once a concept drift is observed, the worst classifier of the ensemble which is calculated using the error estimated by ADWIN optimizer is replaced with a new classifier which learns the weight to handle the drift.

Hoeffding trees [29] are a class of tree-based models for incremental learning which has different flavors for different use cases. For example: Hoeffding Any Time Tree (HATT), vanilla Hoeffding Tree, EFDT etc. Each variant has its own advantages and disadvantages. Some are slower, others are faster but less accurate and so on. These methods work by splitting re-evaluation based on node purity and provides a theoretical upper bound for converges. Such methods continually revisit the nodes of the trees and update its internal structure. It handles non-stationarity better than many incremental learning algorithms.

### III.3.2 Ensemble Online Learning

The multiple combinations of methods described in the above section are implemented with different parameters to train an online learning algorithm. More than 15 combinations are trained with either different feature sets or model parameter sets and then those are compared with each other and the available literature on best performing models.

The schematic diagram of the best performing ensemble is shown in Figure 3. It shows a model pipeline consisting of three different classifiers namely AdaBoost classifier, Leveraging bagging classifier and logistic regression. The data is also scaled in the range (0,1) before model training using a standard scaler. The number of classifiers within the boosting ensemble is 3. Random seed is set as well so that the model results can be reproduced. In comparison to other algorithms, the execution time of the developed solution is also much better with 192 seconds whereas some of the other classifiers such as ARF classifier takes as much as 1800 seconds with up to 10 internal boosting classifiers.

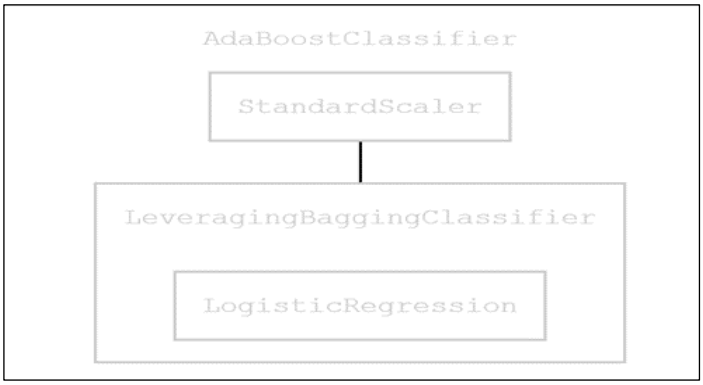


Figure 3: The best performing ensemble of incremental methods.
Source: Authors, (2024).

**Pseudocode for AdaBoost Optimized:**

Input: Define the training data set T1 for classifier, Number of basic supervised classifiers M and streaming incremental dataset D
Output: Class values for each sample of Testing set T2

BEGIN
01: Create the first ensemble leveraging bagging classifier model on the training data set T1 by Bagging
02: t=0
03: Repeat:
04: t=t+1
05: Fetch the new incoming sample from incremental set Di
06: Classify the sample Di by base classifier logistic regression on the previous ensemble model
07: Update parameters of each leveraging classifier
08: Calculate error to find the classifier performing below threshold by base classifier
09: if the sum of error < 0.5
10: Store the label of Di
11: A new base classifier to be trained on the labeled Di
12: Bagging tree to be pruned, and the new classifier to be added to the ensemble bagging model to replace the worst-performing one
13: Assign the new classifier a weight based on misclassification error
14: else
15: pass
16: End if condition
17: Update the Leveraging bagging ensemble model parameters to the Ada-boost classifier
18: continue until the end of the data stream from D.
19. end for loop
20. END

### III.3.3 Training Process

The model process is shown in Figure 4. The original dataset is used for model training as well as derived features generated using polynomial feature engineering. Multiple different models are trained, results of which are shown in result section. Accuracy, Recall, Precision and F1-score are used to measure the performance of trained models. Effective data processing steps such as NA removal, duplicate checking, scaling,

normalization, outlier detection and train test split are performed before model training.

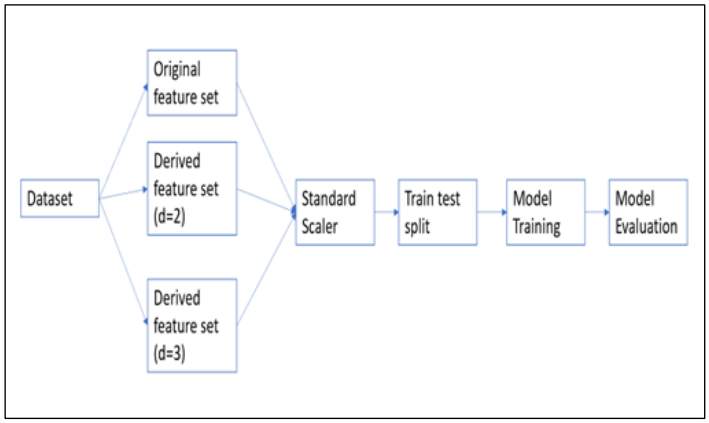


Figure 4: Training process for concept drift adaptation.
Source: Authors, (2024).

## IV. RESULTS AND DISCUSSIONS

Model experiments and results are documented in this section. Table 2 shows the various methods implemented on the electricity dataset and the result obtained on the same. Model column provides the name of the algorithm, description column provides the information on model parameters and accuracy and f1-score is shown in third and fourth column. The result is sorted on increasing order of f1-score. The very last row of the table shows the best model. The precision and recall values are not shown for the sake of brevity but the same can be understood from the f1-score values. For the best model, precision 89.14% is and recall is 87.97%. the best model is a combination of AdaBoost classifier used with the pipeline of leveraging bagging classifier and logistic regression with number of models as 3. It provides an accuracy of 90.32% and f1-score of 88.55%. The developed algorithm is named AdaBoost Optimized as highlighted in Table 2 with a star mark.

Table 2: Model Comparison

| **Model** | **Description** | **Accuracy (%)** | **F1-Score (%)** |
|---|---|---|---|
| ADWIN Bagging Classifier | Number of models: 3 | 67.31 | 53.69 |
| ADWIN Boosting Classifier | Number of models: 3 | 68.06 | 58.8 |
| Hoeffding Tree Classifier | Default parameters | 74.02 | 65.24 |
| Extremely Fast Decision | | | |
| Tree Classifier | Default parameters | 74.28 | 65.58 |
| ARF Classifier | Hyper parameter updated | 75.6 | 67.73 |
| FFM Classifier | Nfactors 10, intercept=.5, | 75.12 | 67.87 |
| AMF Classifier | Default parameters | 76.03 | 68.34 |
| Leveraging Bagging Classifier | ARF Classifier as base, No. of models: 7 | 75.25 | 68.45 |
| AdaBoost Classifier | Number of models: 5 | 75.52 | 68.6 |
| AdaBoost Classifier | Hoeffding Adaptive Tree Classifier as base | 74.67 | 68.95 |
| ARF Classifier | Leaf_prediction=mc, Number of models: 3 | 76.74 | 70.02 |
| AdaBoost Classifier | ARF Classifier as base, No, of models:5 | 79.58 | 75.34 |
| AdaBoost Classifier | ARF Classifier as base, No. of models: 10 | 80.22 | 76.25 |
| **Ada Boost Classifier*** | **Leveraging Bagging Classifier and Logistic Regression as base.** | **90.32** | **88.55** |

Source: Authors, (2024).

Model results on derived features with degree 2 and 3. From Table 3, it is evident that increasing number of features is not helping the model. When the same ARF classifier is trained using original set of features, result is slightly better with accuracy of 76.74% and f1-score of 70.02% in comparison to derived features of 75.31% as accuracy and 67.67% as f1-score.

Table 3: Models on derived features.

| Model | Derived features degree | Accuracy | F1-Score |
|---|---|---|---|
| ARF Classifier | D=2 | 73.79 | 64.09 |
| ARF Classifier | D=3 | 75.31 | 67.67 |

Source: Authors, (2024).

The result obtained is compared with the result obtained by other researchers on the same dataset as shown in Table 4. The best results obtained so far are DDG-DA [5], DDM [2] and leveraging bagging classifier [30] with the accuracy of 84.98%, 85.41% and 88.12% respectively. In comparison to above, AdaBoost Optimized obtains an accuracy of 90.32% which is several steps ahead. The developed model has a better f1-score as well, which shows that the model is quite stable even with class imbalance.

Table 4: Model Performance Comparison.

| | Accuracy (%) | F1-Score (%) |
|---|---|---|
| DDG-DA [5] | 84.98 | |
| DDM [2] | 85.41 | |
| Leveraging Bagging classifier [30] | 88.12 | 86.45 |
| AdaBoost Optimized | 90.32 | 88.55 |

Source: Authors, (2024).

In order to further show the advantage of online learning, A classification model is trained using batch learning on the same data and the compare the result with online learning. A number of classification models are trained, but only the best performing model is shown in Table 5. Random forest model performs the best with an accuracy of 84.07% and F1-score as 80.62 % which is much lower compared to the online algorithm AdaBoost Optimized.

Table 5: Batch learning vs online learning.

| | Model Name | Accuracy | F1-Score |
|---|---|---|---|
| Batch | Random Forest | 84.07 | 80.62 |
| Online | AdaBoost Optimized | 90.32 | 88.55 |

Source: Authors, (2024).

## V. CONCLUSIONS

Concept drift is a prevalent issue observed in operational Machine Learning models. It is critical to incorporate into the ML pipeline the tools and techniques required to detect and address concept drift; failure to do so will result in a gradual degradation of model performance. and results from it will not be useful. Incorporating some basic steps in ML production pipeline can help in detecting potential errors early and help keep models updated and relevant. Methods developed in this paper do a great job handling concept drift. It can further be improved for unstructured data scenarios.

## VI. AUTHOR'S CONTRIBUTION

**Conceptualization:** Ushashree P.
**Methodology:** Ushashree P.
**Investigation:** Ushashree P.
**Discussion of results:** Ushashree P, R B V Subramanyam
**Writing – Original Draft:** Ushashree P.
**Writing – Review and Editing:** Ushashree P
**Supervision:** R B V Subramanyam
**Approval of the final text:** Ushashree P

## VII. ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Professor R B V Subramanyam for his constant guidance and support. I would also like to thank my friends and colleagues for their valuable input and suggestions. I am also grateful to my parents without their support nothing would have been possible.

## VIII. REFERENCES

[1] Mehta, S. (2017). Concept drift in streaming data classification: algorithms, platforms and issues. Procedia computer science, 122, 804-811.

[2] Yan, M. M. W. (2020). Accurate detecting concept drift in evolving data streams. ICT Express, 6(4), 332-338.

[3] Ng, W. W., Zhang, J., Lai, C. S., Pedrycz, W., Lai, L. L., & Wang, X. (2018). Cost-sensitive weighting and imbalance-reversed bagging for streaming imbalanced and concept drifting in electricity pricing classification. IEEE Transactions on Industrial Informatics, 15(3), 1588-1597.

[4] Ramírez-Gallego, S., Krawczyk, B., García, S., Woźniak, M., & Herrera, F. (2017). A survey on data preprocessing for data stream mining: Current status and future directions. Neurocomputing, 239, 39-57.

[5] Li, W., Yang, X., Liu, W., Xia, Y., & Bian, J. (2022, June). DDG-DA: Data Distribution Generation for Predictable Concept Drift Adaptation. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 36, No. 4, pp. 4092-4100).

[6] Lu, J., Liu, A., Song, Y., & Zhang, G. (2020). Data-driven decision support under concept drift in streamed big data. Complex & intelligent systems, 6(1), 157-163.

[7] Agrahari, S., & Singh, A. K. (2022). Concept drift detection in data stream mining: A literature review. Journal of King Saud University-Computer and Information Sciences, 34(10), 9523-9540.

[8] Webb, G. I., Lee, L. K., Petitjean, F., & Goethals, B. (2017). Understanding concept drift. arXiv preprint arXiv:1704.00362.

[9] Harel, M., Mannor, S., El-Yaniv, R., & Crammer, K. (2014, June). Concept drift detection through resampling. In International conference on machine learning (pp. 1009-1017). PMLR.

[10] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM computing surveys (CSUR), 46(4), 1-37.

[11] Wang, S., Schlobach, S., & Klein, M. (2011). Concept drift and how to identify it. Journal of Web Semantics, 9(3), 247-265.

[12] Faria, E. R., Gonçalves, I. J., de Carvalho, A. C., & Gama, J. (2016). Novelty detection in data streams. Artificial Intelligence Review, 45, 235-269.

[13] Masud, M. M., Al-Khateeb, T. M., Khan, L., Aggarwal, C., Gao, J., Han, J., & Thuraisingham, B. (2011, December). Detecting recurring and novel classes in concept-drifting data streams. In 2011 IEEE 11th International Conference on Data Mining (pp. 1176-1181). IEEE.

[14] Iwashita, A. S., de Albuquerque, V. H. C., & Papa, J. P. (2019). Learning concept drift with ensembles of optimum-path forest-based classifiers. Future Generation Computer Systems, 95, 198-211.

[15] Hidalgo, J. I. G., Maciel, B. I., & Barros, R. S. (2019). Experimenting with prequential variations for data stream learning evaluation. Computational Intelligence, 35(4), 670-692.

[16] Wang, P., Jin, N., & Fehringer, G. (2020, August). Concept drift detection with false positive rate for multi-label classification in iot data stream. In 2020 International Conference on UK-China Emerging Technologies (UCET) (pp. 1-4). IEEE.

[17] Mahdi, O. A., Pardede, E., Ali, N., & Cao, J. (2020). Diversity measure as a new drift detection method in data streaming. Knowledge-Based Systems, 191, 105227.

[18] de Mello, R. F., Vaz, Y., Grossi, C. H., & Bifet, A. (2019). On learning guarantees to unsupervised concept drift detection on data streams. Expert Systems with Applications, 117, 90-102.

[19] Yang, Z., Al-Dahidi, S., Baraldi, P., Zio, E., & Montelatici, L. (2019). A novel concept drift detection method for incremental learning in nonstationary environments. IEEE transactions on neural networks and learning systems, 31(1), 309-320.

[20] Khezri, S., Tanha, J., Ahmadi, A., & Sharifi, A. (2020). STDS: self-training data streams for mining limited labeled data in non-stationary environment. Applied Intelligence, 50, 1448-1467.

[21] Faithfull, W. J., Rodríguez, J. J., & Kuncheva, L. I. (2019). Combining univariate approaches for ensemble change detection in multivariate data. Information Fusion, 45, 202-214.

[22] Harries, M., & Wales, N. S. (1999). Splice-2 comparative evaluation: Electricity pricing.

[24] Žliobaitė, I. (2010). Learning under concept drift: an overview. arXiv preprint arXiv:1010.4784.

[23] Oza, N. C., & Russell, S. J. (2001, January). Online bagging and boosting. In International Workshop on Artificial Intelligence and Statistics (pp. 229-236). PMLR.

[25] Bifet, A., Holmes, G., & Pfahringer, B. (2010). Leveraging bagging for evolving data streams. In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part I 21 (pp. 135-150). Springer Berlin Heidelberg.

[26] LaValley, M. P. (2008). Logistic regression. Circulation, 117(18), 2395-2399.

[27] Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Enembreck, F., Pfharinger, B., ... & Abdessalem, T. (2017). Adaptive random forests for evolving data stream classification. Machine Learning, 106, 1469-1495.

[28] Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., & Gavalda, R. (2009, June). New ensemble methods for evolving data streams. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 139-148).

[29] Manapragada, C., Webb, G. I., & Salehi, M. (2018, July). Extremely fast decision tree. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 1953-1962).

[30] Zliobaite, I. (2013). How good is the electricity benchmark for evaluating concept drift adaptation. arXiv preprint arXiv:1301.3524.