2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Detection of Real Concept Drift Under Noisy Data Stream

1st Sirvan Parasteh
Computer Science Department
University of Regina, Regina, Canada
Sirvan.Parasteh@uregina.ca

2nd Samira Sadaoui
Computer Science Department
University of Regina, Regina, Canada
Samira.Sadaoui@uregina.ca

3rd Mohammad Sadegh Khosravani
Computer Science Department
University of Regina, Regina, Canada
mko041@uregina.ca

***Abstract*—Concept drift detection in noisy data streams is challenging yet essential. This paper introduces NPRDD, a new concept drift detection algorithm that is robust to noise and accurately identifies Real drifts. NPRDD operates on a moving window of recent data, utilizing predicted class probabilities and cross-entropy-based surprise measures to weigh real drift candidates. In line with the Bayesian definition of Real concept drift, NPRDD considers a sample as a drift candidate when the classifier makes an error but is highly confident in its judgment. We evaluate NPRDD on synthetic datasets by varying the noise levels and comparing its performance with other well-established methods. Our results show that NPRDD outperforms other methods regarding ROC-AUC and Accuracy metrics.**

***Index Terms*—Real concept drift, noisy data stream, concept drift detection, model adaptation, cross-entropy, surprise level.**

I. Introduction

In non-stationary environments, the data-generating sources may change over time, leading to discrepancies in the distributions of observations between the training and deployment of machine learning (ML) models. This phenomenon, known as Concept Drift (CD), can significantly lower the prediction quality because the probabilistic relationships between the input and output variables evolved [7], [18]. Drifted data, if not adequately accounted for, usually leads to incorrect classifications and negatively impacts the performance of ML models in real-world scenarios. For example, unforeseen events, such as the COVID-19 pandemic or the introduction of new products and services in the market, can alter buyers' spending behavior, causing shifts in the performance of ML models. As data-generating sources evolve, continuous monitoring of ML models and frequent updates become essential to accommodate new emerging concepts [13]. For instance, adapting existing ML models using recent samples is crucial for sudden and significant decreases in predictive performance. Understanding and detecting these unpredictable data changes is vital for developing robust model adaptation mechanisms.

The CD is a complex notion that can manifest in various forms, including its types, Real, Virtual and Mixed, and transition speeds, such as Abrupt, Gradual and Incremental. This paper focuses on the Real CD, which represents changes in the statistical properties of the target class variable [13]. The presence of noisy data in the data stream poses challenges for traditional CD detection techniques.

Several research studies have explored the performance of various CD detection algorithms [8] and [3]. These investigations reveal that no single approach consistently outperforms others across all scenarios. The choice of detection method is closely linked to the specific needs of each application, including the dataset's characteristics and the ML models in use. Furthermore, these past studies identified several limitations in existing CD detectors, including the sensitivity to parameters' tuning, a high running time, and difficulty tackling complex data streams. Additionally, despite that noise negatively influences data interpretation and machine learning performance, the majority of drift detection techniques ignore noise presence, even though they're often sensitive to it [17]. Such oversight complicates data stream mining and demonstrates the need for efficient CD detectors to deal with noisy data and diverse data types.

Our study aims to address the above limitations by focusing specifically on the Real CD and identifying noise in the class label. Our supervised method, called NPRDD (Noise-aware Probabilistic Real Drift Detection), leverages any base classifier that is capable of providing class probability estimates. Our method employs a moving window of recent samples, maintaining statistics, such as probabilities, surprise levels and the real drift ratio. NPRDD balances the detection of genuine drifts with the minimization of false alarms due to noise by utilizing a combination of predicted class probabilities and cross-entropy-based surprise measures. The relative surprises, calculated within the current window, are used to weigh the real drift candidates, facilitating differentiation between noise and actual changes in the data distribution. NPRDD offers a quick adaptation of the base classifier to the detected concept changes, ensuring reliable performance in dynamic environments. We extensively evaluate the performance of NPRDD and four well-established CD detection methods on several synthetic Abrupt datasets possessing different levels of noise (5%, 10%, 15% and 20%), with a total of 12 datasets.

The remainder of this paper is organized as follows. Section II provides a formal definition of Real concept drift and highlights Abrupt drifts. Section III describes the CD detection methods used in the experimental comparison.

Section IV introduces our method, NPRDD, and describes its algorithmic details and key features. Section V presents the extensive experimental study and performance results. Finally, Section VI concludes the paper, summarizing the contributions of our research and suggesting directions for future work.

II. Definitions of Real Concept Drift

This paper utilizes the formal, probabilistic definitions of CD presented in [7], [10], [19]. In a data stream setting, data arrives continuously and often quickly, which is modeled as a sequence of samples: $\{(x_1, y_1), (x_2, y_2), \cdots\}$, where x_i is the feature vector and y_i its target label. Generally speaking, the goal of a classifier is to define the relationship between the dependent variables and the independent variable and then make predictions for new data: $\hat{y}_i = f(x_i)$ [7]. More precisely, based on the Bayesian Decision Theory, the classifier predicts $\hat{y}$ using the equation below [4], [10]:

$$\hat{y} = \arg \max_{y \in 0,1,..,l} P(y|\mathbf{x}) \tag{1}$$

where $P(y|\mathbf{x})$ denotes the posterior probability of class y given the input features x, and l is equal to the size of the label set. This conditional probability is defined as follows using the Bayes' theorem:

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \tag{2}$$

where $P(\mathbf{x}|y)$ is the likelihood distribution of the input features given the class label, $P(y)$ is the prior distribution of the target label, and $P(\mathbf{x})$ is the marginal distribution of the features. Hence, a classification problem can be represented as the joint probability distribution of the class and feature variables, $P(y, \mathbf{x})$.

When CD occurs, the two probability distributions $P(\mathbf{x}|y)$ and $P(y)$ have changed, leading to a shift to the posterior distribution $P(\mathbf{y}|x)$ [6]. Therefore, the Bayesian framework is a strong solution for CD representation and detection, as it captures the shifts in the joint distribution $P(y, \mathbf{x})$ and classifies samples accordingly [10]. Based on the notations given in [7], [19], the arriving data up to time t can be defined as follows [19]:

$$Concept_{[t]} = P_{[t]}(\mathbf{x}, y) \tag{3}$$

We assume the learned concept at time t remains stable for a period, and then it may turn into a new concept at time u. The latter is a time after t when the concept has shifted:

$$Concept_t \neq Concept_u <=> P_t(\mathbf{x}, y) \neq P_u(\mathbf{x}, y) \tag{4}$$

In the literature, two types of CD have been defined: (1) Real drift, where only the decision boundary has changed, and (2) Virtual drift, where only the input-feature distribution has changed. This study focuses on the Real drift, which means the probability distribution $P(y|x)$ shifted, while there is no change in the distribution probability $P(x)$ [13]:

$$P_t(y \mid \mathbf{x}) \neq P_u(y \mid \mathbf{x}) \quad such\ that\ P_t(\mathbf{x}) = P_u(\mathbf{x}) \tag{5}$$

Another characteristic of CD is the drift transition (i.e., the speed of change), which has often been categorized as Abrupt, Gradual, and Incremental to express whether the change levels are small or significant. These aspects carry essential information that can be utilized to develop drift-handling mechanisms. This study focuses on the Abrupt (sudden) changes for real CD. In abrupt drifts, a learned concept C_t switches suddenly to another concept C_u, and the progression of change is very rapid. In real-world environments, this abrupt shift can happen for several reasons, such as the outage of an essential service, degradation of a sensor, failure of equipment, and an unexpected weather event.

III. CD Detection Methods

CD detection has been a topic of considerable interest in the ML community, with numerous methods proposed to identify changes in data streams [11], [20]. Among these methods, supervised detection approaches, which rely on monitoring predictive performance or error rates, have garnered significant attention. These methods typically employ statistical tests or data distribution monitoring to capture significant changes in a learner's performance. These methods focus on identifying CD in general, regardless of its type (Real or Virtual), and also they are sensitive to noisy data.

In this section, we describe some of the most prominent supervised CD detection methods, including statistical process control techniques and error-distribution monitoring approaches:

- *EDDM (Early Drift Detection Method)*: EDDM [1] is a statistical detector and an estimator that monitors the distribution of distances between consecutive classification errors. It utilizes an exponentially weighted moving average approach to track changes by assigning decreasing weights to old samples and giving more weights to recent samples. It is designed to detect both gradual and abrupt changes while maintaining low false positive rates. EDDM is particularly effective at detecting early signs of drifts.

- *KSWIN (Kolmogorov-Smirnov Windowing)*: KSWIN [15] utilizes the Kolmogorov-Smirnov (KS) test, a non-parametric method used to compare two samples to check if they are drawn from the same distribution. Within this framework of a sliding window, KSWIN continuously compares the distribution of the latest data with previous ones. If the KS test yields a value exceeding a predefined threshold, it signals a potential drift. One of the advantages of this method is that it can detect distribution changes without any assumptions about the specific distributions in play. Moreover, as data arrive, older data are naturally phased out from the window, ensuring adaptability to the most recent data trends.

- *HDDM (Hellinger Distance Drift Detection)*: HDDM, introduced in [5], is a change detection method that measures the dissimilarity between two probability distributions by harnessing the Hellinger distance. Compared to other metrics, like the Kullback-Leibler divergence, the Hellinger distance provides a bounded metric, making it more interpretable in many contexts. Furthermore, HDDM incorporates Hoeffding's bounds to provide statistical guarantees on the detected drifts, making it robust against false alarms. HDDM offers two variants: HDDM_A, which is fine-tuned to detect abrupt changes by being more sensitive to sudden shifts, and HDDM_W, optimized for detecting gradual drifts by accumulating evidence over time before signaling a change.

Recent trends in the field of CD detection have seen a growing interest in leveraging probabilistic methods and model uncertainty insights. A study by Baier et al. [2] highlighted the use of uncertainty in neural network models to detect concept drift, identifying changes in data patterns using specific uncertainty metrics. Moreover, another important contribution in this domain is the study by [14], which introduced the SPNCD algorithm. This algorithm uses a Sum-Product Network (SPN) model to gain a clear understanding of the data stream's probability distributions. More specifically, SPNCD leverages predicted probabilities from the SPN model and combines them with the base ML model's prediction results to detect drifts effectively. However, the SPNCD's dependence on the SPN as a distinct model has added computational demands, which can be a concern in streaming settings where quick processing is essential.

Building on such insights, we introduce the NPRDD algorithm in this paper. Unlike the above methods, NPRDD uses the predicted probabilities from the classifiers directly, eliminating the need for additional models. This method helps us understand the confidence level of the classifier as well as distinguish between actual data changes and noise effectively. As a result, NPRDD appears as a reliable method for detecting real concept drift in noisy data streams, which will be elaborated on in the subsequent sections.

IV. The Proposed Method for Real CD Detection

Our CD detection algorithm NPRDD is designed to operate robustly under noisy data streams, leveraging any base classifier that is capable of providing class probability estimates. The algorithm employs a moving window of recent samples, maintaining statistics, such as probabilities, surprise levels, and the real drift ratio. Our new approach emphasizes balancing the detection of genuine drifts with minimizing false alarms due to noise, utilizing a combination of predicted class probabilities and cross-entropy-based surprise measures. The relative surprises, calculated within the current window, are used to weigh real drift candidates, facilitating differentiation between noise and actual changes in the data distribution.

Under the prequential evaluation framework, the classifier incrementally learns from consecutive samples, maintaining stability as long as the underlying concept remains stable. Upon detecting a conceptual shift, the algorithm trains a new classifier to adapt to the change, ensuring a more precise and responsive adaptation. In the following sections, we present the main steps of the proposed algorithm. The algorithm detects drifting samples, known as real drift candidates, computes their proportions within the recent window, and weighs them using the surprise measure. A drift alarm will be triggered if the proportion surpasses a given threshold (called T_{alarm}).

A. Identification of Real Drift Candidates

In our algorithm, we introduce a new criterion for identifying real drift candidates, in line with the Bayesian definition of real CD [13]. Specifically, when the classifier makes an error but is highly confident in its judgment, we consider the sample as a real drift candidate. This mechanism reflects an underlying change in $P(y \mid \mathbf{x})$, distinguishing real concept drift from noise and other variations in the data stream. A drift candidate for a given sample is defined as:

$$d = \begin{cases} 1 & \text{if } y_{\text{pred}} \neq y_{\text{true}} \text{ and } \max(q) \geq T_{\text{real}} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where y_{pred} is the predicted label, y_{true} is the true label, and q is the predicted probability of y_{pred}. The threshold T_{real} denotes the Exponential Moving Average (EMA) of the prediction probabilities of the classifier, offering a balance between recent and historical prediction performance. It is updated continuously to ensure the algorithm distinguishes genuine drifts from the noise.

B. Evaluation of Surprise Level

The surprise level in our CD detector is quantified using the cross-entropy $H(p, q)$, calculated between the true probability distribution p and the predicted distribution q. Cross-entropy is widely employed in information theory and ML and assesses the dissimilarity between the true and predicted distributions. The binary classification form of the cross-entropy is defined as follows [9]:

$$H(p, q) = -p_0 \cdot \log(q_0) - p_1 \cdot \log(q_1) \tag{7}$$

where p_0 and p_1 represent the true probabilities of class 0 and 1, respectively, and q_0 and q_1 denote the corresponding predicted probabilities. We apply the cross-entropy to compute the relative surprise, representing the sample's unexpectedness compared to the recent window of samples:

$$S(p, q) = H(p, q) - \frac{1}{l} \sum_{i=1}^{l} H(p_i, q_i) \tag{8}$$

where l denotes the size of label set.

C. Weighting the Real Drift Candidates

The proportion of drift candidates is computed and weighted based on the relative surprise within the sliding window of size n. The weighted drift ratio D_w is formulated as below:

$$D_w = \frac{\sum_{i=1}^{n} w_i \cdot d_i}{\sum_{i=1}^{n} w_i} \tag{9}$$

with the weight for the i-th sample as:

$$w_i = \frac{1}{1 + \sqrt{|S(p_i, q_i)|}} \tag{10}$$

Our approach ensures precise adaptation to concept changes, accounting for localized anomalies within the data stream. The nuanced handling of real drift candidates, denoted by (d_i), ensures the detection mechanism remains sensitive to true underlying changes while being resilient to noise. Our algorithm efficiently computes D_w without exhaustive re-calculations. Rather than iterating through the entire sliding window for each new data point, we maintain a running sum. With each new sample, we adjust this sum by subtracting the oldest value and adding the new one, achieving $O(1)$ time complexity. This streamlined process ensures rapid updates, making our method ideal for real-time applications that demand immediate response.

D. Drift Detection: Alarming Drift

The real drift ratio (D_w) is calculated based on the real drift candidates within a sliding window. A drift alarm is triggered if D_w surpasses a predefined threshold T_{alarm}. This threshold is used to identify when the real drift rate within the window has surpassed a level that justifies the declaration of a real CD.

$$\text{Drift Alarm} = \begin{cases} 1 & \text{if } D_w \geq T_{\text{alarm}} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

E. Algorithm Design

Our Real CD Detector, presented in Algorithm 1, encompasses the essential aspects of drift detection, including the precise adaptation to concept changes and the nuanced handling of real drift candidates, denoted by d. The remaining parameters include the warmup threshold, allowing the model to acclimate to the data stream, and the static threshold T_{alarm}, empirically set to 0.47. The value for T_{alarm} and the window size have been set to ensure a balance between sensitivity to genuine drifts and robustness against noise. The adaptive real drift threshold T_{real} is updated using $\alpha = 0.3$, facilitating the algorithm's effective adjustment to CD in the data stream.

Algorithm 1 Weighted Real CD Detection Algorithm

Require: dataStream (continuous), windowSize = 20, T_{alarm} = 0.47, warmupThreshold = 20, $\alpha = 0.3$
Ensure: Drift detection and classifier update

Initialize classifier and window parameters
sampleCount = 0, T_{real} = 0, $D_w = 0$, P_{sum} = 0
Initialize $windowProbs$ as empty
for each sample x_i in dataStream **do**
    sampleCount += 1
    Predict the label y_{pred} for x_i
    Calculate probability q associated with y_{pred}
    Update the classifier using the true label y_{true}
    P_{sum} += q
    Append q to $windowProbs$
    if sampleCount $>$ windowSize **then**
        P_{sum} -= $windowProbs[0]$
        Remove $windowProbs[0]$
    end if
    if sampleCount $>$ warmupThreshold **then**
        Compute relative surprise $S(p, q)$ (*eq. 8*)
        Calculate weight w_i of x_i (*eq. 10*)
        if (label is incorrect) and ($\max(q) \geq T_{\text{real}}$) **then**
            $d_i = 1$ (*real drift candidate*)
            Update weighted real drift ratio D_w (*eq. 9*)
        end if
        $Q_{\text{avg}} = P_{\text{sum}} / \min(windowSize, sampleCount)$
        if label is incorrect **then**
            $T_{\text{real}} = (1 - \alpha) \times Q_{\text{avg}} + \alpha \times T_{\text{real}}$
        end if
        if (window is full) and ($D_w > T_{\text{alarm}}$) **then**
            Signal drift
            Reset window parameters and P_{sum}
            Re-initialize classifier
            sampleCount = 0
        end if
    end if
end for

V. Experimental Setup

a) Datasets with Different Noise Levels: We utilize three synthetic Abrupt datasets for our experiments, *Mixed_0101*, *RandomTree_2563789698568873* and *Sine_0123*, which are publicly available on the Harvard Dataverse platform. These datasets were produced using existing stream generators: Mixed, Random Tree and Sine. Each dataset comprises 40,000 samples and is designed for binary classification tasks [12]. These datasets encompass four distinct concepts and incorporate three Abrupt drifts located at positions 10,000, 20,000, and 30,000 within the data stream. To investigate the influence of label noise on the models' performance, we introduce artificial noise by flipping labels in accordance with the predetermined noise levels: 5%, 10%, 15%, and 20%, with a total of 12 datasets.

b) Concept Drift Detection Algorithms: In addition to our method NPRDD, we evaluate four renowned CD detectors, namely HDDM_A, HDDM_W, KSWIN and EDDM. We also consider the baseline scenario without any drift detection part (called NoDetector) to elucidate the influence of drift detection on models' efficiency. With regard to KSWIN method, it possesses a degree of non-determinism stemming from its built-in sampling process. To accommodate this non-

TABLE I
MODEL PERFORMANCE (ROCAUC, ACCURACY AND NUMBER OF DRIFT POINTS) ACROSS ABRUPT DATASETS

Model	**NoDetector**			**KSWIN**			**EDDM**			**HDDM_W**			**HDDM_A**			**NPRDD**		
Dataset	ROCAUC.	ACC	#drifts	ROCAUC.	ACC	#drifts	ROCAUC.	ACC	#drifts	ROCAUC.	ACC	#drifts	ROCAUC.	ACC	#drifts	ROCAUC.	ACC	#drifts
Mixed_05%	0.6208	0.5709	0	0.9228	0.8713	3	0.7544	0.6798	701	0.7771	0.6678	3	0.8321	0.7372	11	**0.9233**	**0.8717**	3
Mixed_10%	0.6238	0.5710	0	0.8699	0.8242	4	0.7208	0.6583	640	0.7933	0.7134	5	0.7570	0.6700	4	**0.8710**	**.8254**	3
Mixed_15%	0.6070	0.5557	0	0.8214	**0.7828**	4	0.7905	0.7539	14	0.8219	0.7814	9	0.8210	0.7812	5	**0.8229**	0.7823	4
Mixed_20%	0.5841	0.5471	0	0.7446	0.7037	6	0.7447	0.7104	10	0.7733	0.7385	13	0.7723	0.7398	7	**0.7748**	**0.7419**	4
Sine_05%	0.6805	0.6023	0	0.9244	0.8757	3	0.8992	0.8516	15	0.9243	0.8739	3	0.9240	0.8747	3	**0.9244**	0.8754	3
Sine_10%	0.6692	0.6042	0	0.8678	0.8223	4	0.8429	0.7989	7	0.8714	0.8274	4	0.8712	0.8267	3	**0.8720**	0.8279	3
Sine_15%	0.6471	0.5950	0	0.8150	0.7735	4	0.7862	0.7485	14	0.8187	0.7783	5	0.8183	0.7763	4	**0.8201**	**0.7785**	4
Sine_20%	0.6138	0.5804	0	0.7700	0.7343	5	0.7365	0.7039	20	0.7641	0.7318	10	0.7704	0.7342	3	**0.7713**	**.7353**	4
RT_05%	0.7565	0.6983	0	0.8017	0.7381	5	0.7946	0.7502	6	0.7944	0.7485	13	**0.8243**	**0.7672**	3	0.8116	0.7667	5
RT_10%	0.7181	0.6702	0	0.7388	0.6988	4	0.7433	0.7054	23	0.7541	0.7192	18	**0.7837**	**0.7402**	4	0.7803	0.7371	4
RT_15%	0.6777	0.6453	0	0.7201	0.6862	4	0.7057	0.6760	16	0.7121	0.6868	19	**0.7333**	**0.6982**	5	0.7321	0.6948	8
RT_20%	0.6420	0.6205	0	0.6763	0.6516	6	0.6614	0.6426	11	0.6771	0.6585	17	**0.6989**	**0.6749**	**3**	0.6910	0.6676	7
Average	0.653	0.605	0.000	0.806	0.764	**4.333**	0.765	0.723	123.083	0.790	0.744	9.91	0.801	0.752	4.583	**0.816**	**0.775**	4.33

determinism, we conduct a series of 10 independent runs for each dataset when assessing with KSWIN. The reported results for this method represent the average outcomes of these multiple runs, offering a more reliable measure of its performance.

c) Base Learner: We employ the *HoeffdingTreeClassifier* as the base classifier for the five CD detection methods. We adopt the prequential learning approach, where each unseen sample from the data stream is utilized for testing the current classifier. Subsequently, this sample is incorporated into the training phase to update the classifier incrementally. The online learning paradigm ensures the model's continuous adaptation to evolving data.

d) Evaluation Metrics: We employ Accuracy and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) as our primary evaluation metrics. These metrics gauge the classifiers' performance across different noise levels and drift detection methodologies. Additionally, we record the number of drift points (locations) detected by each algorithm.

VI. EVALUATION

A. Performance Results and Interpretation

Table I presents the outcomes derived from evaluating six CD detection algorithms across 12 synthetic datasets with different noise levels. As anticipated, increasing noise levels corresponded to a notable decline in the models' performance across all scenarios. The NoDetector method is the least performing across all datasets, demonstrating the necessity of CD detection and classifier adaptation. On the other hand, our drift detection algorithm showed resilience to noise, delivering, most of the time, a superior performance in terms of ROC-AUC and Accuracy.

Our experimental results reveal that our proposed method, NPRDD, outperforms other well-established CD detection methods across most datasets regarding ROC-AUC and Accuracy. Specifically, NPRDD emerged as the top performer in 8 out of 12 datasets in terms of ROC-AUC and achieved a very satisfactory Average ROC-AUC of 0.816 across all datasets. In the case of the RT dataset, NPRDD was the second-best performer, trailing only HDDM_A. Regarding Accuracy, NPRDD achieved the highest overall average and secured the top spot in 5 out of 12 datasets. KSWIN was the second-best overall performer, although it only ranked first in the Mixed dataset with 15% noise. HDDM_A dominated the Accuracy metric in 4 datasets, with NPRDD closely following with comparable results.

The superior performance of NPRDD can be attributed to its robustness to noise and its ability to accurately identify true concept drifts. Unlike other methods such as EDDM, KSWIN, and HDDM_W, both NPRDD and HDDM_A exhibit lower sensitivity to noise. However, in the Mixed dataset, HDDM_A demonstrated significantly lower performance in ROC-AUC and Accuracy at 5% and 10% noise levels compared to NPRDD.

The number of detected drifts further supports the effectiveness of NPRDD. The ideal number of detected drifts for each dataset is 3, and NPRDD achieved the lowest average number of detected drifts (4.33) across all datasets, tied with KSWIN. The high classification performance of NPRDD underscores the quality of the detected drifts, making it the top performer in this regard.

B. Ranking Analysis of Concept Drift Detection Methods

Table II presents the ranking analysis of various CD detection methods based on the ROCAUC metric across different datasets. The evaluated methods include No Detector, KSWIN, EDDM, HDDM_W, HDDM_A, and NPRDD. The ranks are assigned based on the ROCAUC values, with lower ranks indicating better performance.

Table II shows that the NPRDD method consistently achieves the best performance, as indicated by its lowest average rank of 1.35. NPRDD effectively detects CD across various datasets. The standard deviation of the ranks for NPRDD is as low as 0.47, indicating consistent ranking performance across various datasets. Among the other methods, HDDM_A and KSWIN show relatively better performance with average ranks of 2.54 and 3.12, respectively. On the other hand, the 'No Detector' method received the highest average rank, consistently placing it last among all methods evaluated in our experiments. Overall, the results indicate that NPRDD is a promising method for detecting CD under noisy data stream.

TABLE II
RANKING ANALYSIS OF CD DETECTION METHODS BASED ON ROCAUC

Model	NoDetector	KSWIN	EDDM	HDDM_W	HDDM_A	NPRDD
Mixed_05%	6.0	2.0	5.0	4.0	3.0	**1.0**
Mixed_10%	6.0	2.0	5.0	3.0	4.0	**1.0**
Mixed_15%	6.0	3.0	5.0	2.0	4.0	**1.0**
Mixed_20%	6.0	4.0	5.0	2.0	3.0	**1.0**
Sine_05%	6.0	**1.5**	5.0	3.0	4.0	**1.5**
Sine_10%	6.0	4.0	5.0	2.0	3.0	**1.0**
Sine_15%	6.0	4.0	5.0	2.0	3.0	**1.0**
Sine_20%	6.0	3.0	5.0	4.0	2.0	**1.0**
RT_05%	6.0	3.0	4.0	5.0	**1.0**	2.0
RT_10%	6.0	5.0	4.0	3.0	**1.0**	2.0
RT_15%	6.0	3.0	5.0	4.0	**1.0**	2.0
RT_20%	6.0	4.0	5.0	3.0	**1.0**	2.0
Average	6.00	3.12	4.85	3.15	2.54	**1.35**
Std Dev	0.00	1.04	0.38	0.99	1.20	**0.47**

VII. CONCLUSION AND FUTURE WORK

Our findings have important implications that extend beyond the scope of the experimental study. The robustness of NPRDD to noise and its ability to detect CD accurately make it an ideal candidate for real-world applications where the precise identification of CD is crucial. In various domains, such as banking fraud detection, network intrusion detection and healthcare diagnosis, NPRDD can improve the overall effectiveness of those systems by providing accurate responses to emerging threats or changes in system behavior. By employing predicted class probabilities and cross-entropy-based surprise measures, NPRDD effectively distinguishes between noise and genuine changes in the data distribution, addressing the challenges posed by noisy data streams. Furthermore, our research contributes to the advancement of CD detection by introducing NPRDD for noisy data streams.

Despite the promising results, our study has some limitations. We only tested NPRDD on synthetic datasets, and its performance on real-world datasets remains to be evaluated. Additionally, our experimental setup did not consider gradual or incremental drifts. Future research could extend NPRDD to handle different types of drifts and assess its performance on real-world and synthetic datasets. Moreover, further studies could explore the integration of NPRDD with other ML algorithms and evaluate its performance in a broader range of applications. Exploring the impact of different window sizes and thresholds on the performance of NPRDD could also provide valuable insights into its adaptability and robustness. Also, we aim to compare our active adaptive learning method to past passive methods that adjust continuously as data arrive [16].

REFERENCES

[1] Baena-Garcıa, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavalda, R., Morales-Bueno, R.: Early drift detection method. In: Fourth international workshop on knowledge discovery from data streams. vol. 6, pp. 77–86 (2006)

[2] Baier, L., Schlör, T., Schöffer, J., Kühl, N.: Detecting concept drift with neural network model uncertainty. 2107.01873, arXiv, cs.LG (2022)

[3] Barros, R.S.M., Santos, S.G.T.C.: A large-scale comparison of concept drift detectors. Information Sciences **451**, 348–370 (2018)

[4] Fieguth, P.: An Introduction to Pattern Recognition and Machine Learning. Springer Cham (2022)

[5] Frias-Blanco, I., del Campo-Ávila, J., Ramos-Jimenez, G., Morales-Bueno, R., Ortiz-Diaz, A., Caballero-Mota, Y.: Online and non-parametric drift detection methods based on hoeffding's bounds. IEEE Transactions on Knowledge and Data Engineering **27**(3), 810–823 (2014)

[6] Gama, J., Castillo, G.: Learning with local drift detection. In: Advanced Data Mining and Applications: Second International Conference, ADMA 2006, Xi'an, China, August 14-16, 2006 Proceedings 2. pp. 42–55. Springer (2006)

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

[8] Gonçalves Jr, P.M., de Carvalho Santos, S.G., Barros, R.S., Vieira, D.C.: A comparative study on concept drift detectors. Expert Systems with Applications **41**(18), 8144–8156 (2014)

[9] Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT press (2016)

[10] Hoens, T.R., Polikar, R., Chawla, N.V.: Learning from streaming data with concept drift and imbalance: an overview. Progress in Artificial Intelligence **1**, 89–101 (2012)

[11] Klinkenberg, R., Renz, I.: Adaptive information filtering: Learning in the presence of concept drifts. Learning for text categorization pp. 33–40 (1998)

[12] López Lobo, J.: Synthetic datasets for concept drift detection purposes. Harvard Dataverse, https://doi.org/10.7910/DVN/5OWRGB (2020)

[13] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., Zhang, G.: Learning under concept drift: A review. IEEE transactions on knowledge and data engineering **31**(12), 2346–2363 (2018)

[14] Parasteh, S., Sadaoui, S.: A novel probabilistic approach for detecting concept drift in streaming data. In: Conte, D., Fred, A., Gusikhin, O., Sansone, C. (eds.) Deep Learning Theory and Applications. pp. 173–188. Springer Nature Switzerland, Cham (2023)

[15] Raab, C., Heusinger, M., Schleif, F.M.: Reactive soft prototype computing for concept drift streams. Neurocomputing **416**, 340–351 (2020)

[16] Sadreddin, A., Sadaoui, S.: Chunk-based incremental feature learning for credit-card fraud data stream. Journal of Experimental & Theoretical Artificial Intelligence pp. 1–19 (2022)

[17] Wang, P., Jin, N., Woo, W.L., Woodward, J.R., Davies, D.: Noise tolerant drift detection method for data stream mining. Information Sciences **609**, 1318–1333 (2022)

[18] Webb, G.I., Hyde, R., Cao, H., Nguyen, H.L., Petitjean, F.: Characterizing concept drift. Data Mining and Knowledge Discovery **30**(4), 964–994 (2016)

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

[20] Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine learning **23**(1), 69–101 (1996)