

N-DD: New Approach for Drift Detection Based on Neutrosophic Support Vector Machine

Rania Lutfi

(Faculty of Informatics Engineering, Al Baath University, Homs 96331, Syria)

Abstract: Many real-world machine learning applications face the challenge of dealing with changing data over time, known as concept drift, and the issue of data indeterminacy, where all the true labels available are unrealistic. This can lead to a decrease in the accuracy of the prediction models. The aim of this study is to introduce a new approach for detecting drift, which is based on neutrosophic set theory. This approach takes into account uncertainty in the prediction model and is able to handle indeterminate information, considering its impact on the model's performance. The proposed method reads data into windows and calculates a set of values based on the concept of neutrosophic membership. These values are then used in the Neutrosophic Support Vector Machine (N-SVM). To address the issue of indeterminate true label data, the values issued by N-SVM are expressed as entropy and used as input for the ADWIN (Adaptive Windowing) change detector. When a drift is detected, the prediction model is retrained by including only the most recent instances with the original training data set. The proposed method gives promising results in terms of drift detection accuracy compared to the state of existing drift detection methods such as KSWIN, ADWIN, and DWM.

Keywords: drift detection; indeterminate labels; uncertainty; neutrosophic set theory; data stream

CLC number:

Document code: A

Article ID: 1005-9113(2024)00-0000-09

0 Introduction

Machine learning is becoming increasingly common in real-world applications. Many industries now use machine learning algorithms to process large amounts of continuously changing data. This is called concept drift. As described in the literature, this phenomenon can lead to poor prediction performance^[1-2]. Although some existing learning algorithms can detect concept drift when all labels are available^[3-5], it is important to note that in most cases it is unrealistic to access all labels^[6]. To address these limitations, this study aims to develop a drift detection approach for situations where labels may be uncertain and the model considers the influence of current inputs on the attributes. Several methods have been used to define the concept drift.

One approach assumes that real labels can be used for retraining and drift detection. This process is known as active learning, where certain decisions are made to provide labels^[7-9]. These labels are then used

to update the model when concept drift is detected, ensuring that the model remains relevant and accurate over time. This approach is particularly useful in complex situations where the data distribution is highly susceptible to change. However, it may face challenges when there is not enough new data to support the update of the model after concept drift occurs, resulting in poor predictions.

Another approach to detecting concept drift is error-based detection, which uses real-time truth labels to identify anomalies and adjust training accordingly^[10-11]. Some commonly used algorithms in this category include the Page-Hinckley test^[12], ADWIN adaptive windowing^[13], and control charts^[14]. The Page-Hinckley test^[12] is known for its simplicity and sensitivity in detecting drift. However, it assumes that changes in the mean always indicate drift, which may not be true in all data sets. Additionally, this method has limitations related to its assumptions and label dependency, which may require domain knowledge. The ADWIN adaptive windowing^[13] has shown good accuracy in real-time

detection, but its performance can be influenced by factors such as the rate at which concept drift occurs. It is also important to note that while ADWIN can detect changes in data distribution, it may not always be able to adapt perfectly to these changes, especially if they are complex.

Control charts^[14] are a prediction error measure commonly used in continual learning to detect concept drift. The experiments used simulated streams of metadata to enable real-time change detection and integration. The results demonstrate that control charts serve as a reliable indicator for detecting concept drift. However, it is worth noting that the learning model may have a tendency to prioritize newly acquired knowledge over past knowledge, potentially leading to the forgetting of previously learned concepts.

Additionally, some methods can be used with error-based detection to monitor the classification error rate and detect drift, such as dynamically weighted majority DWM^[15]. DWM has been shown to outperform other learners in terms of accuracy, particularly in its ability to incrementally learn concept descriptions, utilize previously encountered examples, and employ an unweighted, fixed-size ensemble of experts. This method is specifically designed for tracking concept drift. However, it should be noted that the performance of DWM may be affected by factors such as noise level and the sequence of concept drift.

Concept drift can also be detected by analyzing contextual information, which depends on the input and only requires actual labels only when deviations are detected. One method for detecting drift is to monitor the sample rate within the support vector machine solution limits^[16-17]. Ayad^[16], proposed an approach to recognize and handle concept changes using support vector machines. This approach integrates a mechanism that utilizes only recent and representative patterns to update the SVMs without causing forgetting. However, in a dynamic environment, data characteristics may evolve over time, leading to a significant decline in the performance of this approach. This is because the data used may no longer be consistent with the characteristics of the new incoming data. Nyati et al.^[17] compared the accuracy of various classifiers with and without concept drift. When concept drift is present, the maximum accuracy is 90.4%. The authors

noted that frequent occurrence of concept drift in a data stream significantly decreased the accuracy.

Another approach is to use the Kolmogorov-Smirnov test^[18], which has also been successfully used to detect drift. The main contribution of the paper is the utilization of the Incremental Kolmogorov-Smirnov test to identify concept drifts without the need for true labels. The study demonstrates a strong level of accuracy, however, but it should be noted that the alpha parameter in KSWIN is highly sensitive and should be set below to 0.01. This indicates that the effectiveness of KSWIN can be greatly influenced by the selection of alpha, and may require precise calibration.

On the other hand, the UDD approach^[19] does not rely on true labels for drift detection. This is achieved using deep neural networks and Monte Carlo methods to retrain the model when it drifts. UDD outperforms other state-of-the-art strategies, such as KSWIN, in terms of accuracy on two synthetic datasets and ten real-world datasets for both regression and classification tasks. However, UDD utilizes Monte Carlo Dropout in conjunction with uncertainty estimates from a deep neural network. This means that the network must be trained with dropout, which may not always be feasible.

One approach available for drift detection is type-driven^[20]. This method aims to address concept drift in data streams by identifying the specific type of drift and using this information to improve the accuracy of adaptation. The authors conducted experiments on both synthetic and real-world data, and the results showed demonstrated that accurately identifying the type of drift can significantly enhance the accuracy of adaptation. However, it should be noted that the drift type identifier is pre-trained on data with known drift types. This could potentially limit the effectiveness of the model on real-world data sets where the all drift type points may not be known.

There are methods to detect drift without requiring access to labels^[21]. However, with this type of methodology, knowledge of changes is limited, which can reduce the effectiveness of drift detection methods.

In this paper, we propose a method that is more suitable for the characteristics of real data, where data streams may have uncertain real labels and structural changes.

Our goal is to improve concept drift detection by considering the impact of current input data on the prediction model. The proposed approach uses Neutrosophic set theory and computation of Neutrosophic entropy into adaptive windows to detect uncertain concept changes over time in data streams with uncertain true labels.

The rest of the paper is organized as follows: Section 1 describes the materials and methods used; Section 2 presents the proposed concept drift detection methodology; Section 3 presents the experimental results and compares them with the results of existing methods; Finally, Section 4 concludes the paper and discusses future research directions.

1 Materials and Methods

1.1 Neutrosophic Support Vector Machine (N-SVM)

Neutrosophic is a powerful tool for classification, particularly in cases where the data is imprecise or uncertain^[22]. The N-SVM method is based on the concept of neutrosophic sets, which contain elements with varying degrees of truth, falsity, and indeterminacy. This allows N-SVM to accurately quantify indeterminacy and make predictions. The definition of N-SVM is based on the definition of Support Vector Machine (SVM)^[23]. To use N-SVM, we start with a set of M training data points $\{(\mathbf{x}_i, y_i)_{i=1}^M\}$ where \mathbf{x}_i is a multidimensional vector and y_i is the corresponding class. In N-SVM, each point \mathbf{x}_i belonging to class y_i which contains a neutrosophic triple $\{T(\mathbf{x}_i), F(\mathbf{x}_i), I(\mathbf{x}_i)\}$ that represents the truth, falsity, and indeterminacy values of point i for class j ^[24]. A data sample (x_i) is presented as $\{T(\mathbf{x}_i), F(\mathbf{x}_i), I(\mathbf{x}_i)\} / \mathbf{x}_i$, where $\{T(\mathbf{x}_i), F(\mathbf{x}_i), I(\mathbf{x}_i)\}$ are the membership values to the truth, indeterminate, and false sets. $T(\mathbf{x}_i)$ is used to measure the belonging degree of the sample to the corresponding labeled class, $I(\mathbf{x}_i)$ for indiscriminate degree between classes and $F(\mathbf{x}_i)$ for the belonging degree to the outliers.

To determine the optimal hyperplane, we can obtain the interval margin by defining a set M of non-negative variables denoted by $\{\theta_1, \theta_2, \dots, \theta_m\}$, N-SVM find an optimal solution by optimizing w with the constraint:

$$\text{For each element } x_i, \text{ we have;} \\ (w \cdot \varphi(x_i) + b) y_i \geq (1 - \theta_i), i = 1, 2, \dots, M \quad (1)$$

where w stands for the weight for optimal hyperplane. The optimal hyperplane is then determined as:

$$\text{Minimum} \left(\frac{1}{2} w^2 + C \sum_{i=1}^M \theta_i \right) \quad (2)$$

where C is a constant that balances between the maximum margin and the minimum error.

1.2 Adaptive Windowing (ADWIN)

ADWIN is an adaptive windowing algorithm designed to detect changes and retain statistical values for a data stream. It maintains an adaptive window, which is crucial for the initial machine learning model. ADWIN continuously updates the window as long as no concept drift is detected. We selected this algorithm because it is proven to effectively handle real data without prior knowledge of data distribution or structure model^[25].

1.3 Entropy for Neutrosophic Sets

Entropy is a measure of the uncertainty present in a set of information. This uncertainty can be attributed to two reasons: partial belongingness and non-belongingness, as well as indeterminacy. For any neutrosophic set X , a sample $\mathbf{x}_i \in X$ is represented as $\{T(\mathbf{x}_i), F(\mathbf{x}_i), I(\mathbf{x}_i)\} / \mathbf{x}_i$, where $T(\mathbf{x}_i)$, $F(\mathbf{x}_i)$ and $I(\mathbf{x}_i)$ represent the membership values for true, false, and indeterminate labels, respectively. The entropy of a neutrosophic set X is calculated using the formula^[26]:

$$E(x) = \text{Min} \{1, (2 - t_x - f_x) \cdot i_x\}, x \in X \quad (3)$$

In our method, the entropy is used as an input for the ADWIN change detector.

2 Methodology

The proposed method takes into account the impact of the indeterminate current input data on the prediction model. Fig.1 outlines the steps in the proposed workflow. Firstly, the data stream is read in windows and presented with neutrosophic components (T, F, I) . Given a set of N training data, each instance inst_i is represented as $\{T(\text{inst}_i), F(\text{inst}_i), I(\text{inst}_i)\} / \text{inst}_i$, where $T(\text{inst}_i)$, $F(\text{inst}_i)$, and $I(\text{inst}_i)$ represent the membership values for true, false, and indeterminate labeling, respectively^[24]. To make predictions, a N-SVM was utilized, as it demonstrated the best performance during our experiments. The values generated by the N-SVM are expressed as entropy and used as input for the ADWIN change detector. When a drift is detected, the prediction model is retrained by including only the

most recent instances (D_{recent}) with the original training data set (ND_{tr}). This guarantees that the

model is updated with the most current data and has enough data for good accuracy.

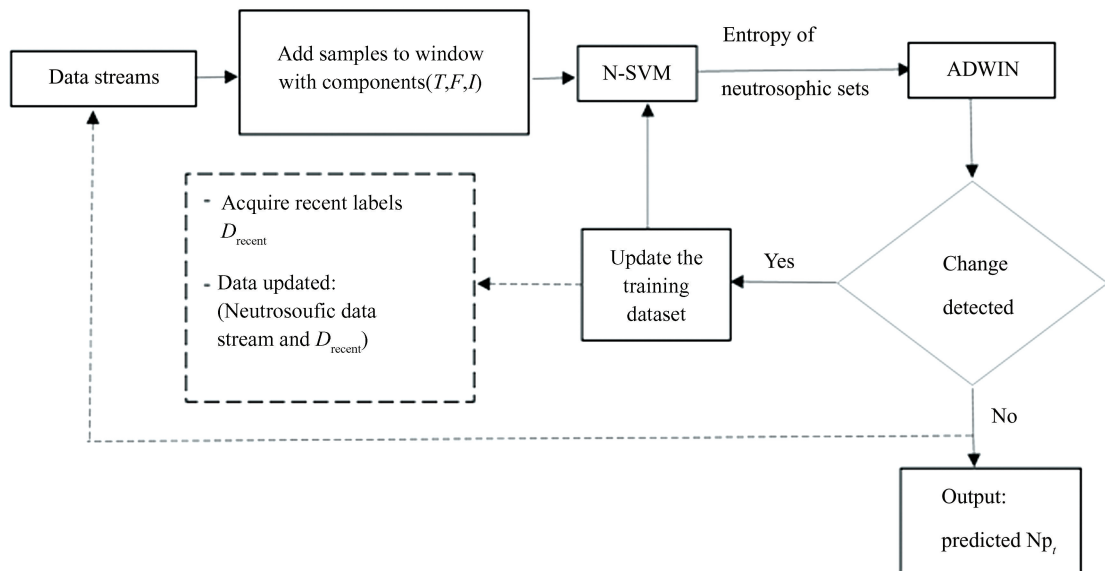


Fig. 1 Steps in the proposed Neutrosophic Drift Detection (N-DD) for drift detection

The semantic algorithm of the proposed method Neutrosophic Drift Detection (N-DD) is outlined below:

Algorithm: N-DD

Input: Neutrosophic data stream (ND) and Neutrosophic training data set ND_{tr}

While do not end neutrosophic data stream ND

Receive instance $inst_t(T, F, I)$

$Np_t, NUnc_t$ predicted $inst_t$

Add (uncertainty as Entropy) $NUnc_t$ to ADWIN

if ADWIN detects changes then

Receive recent labels D_{recent}

Training model with (ND_{tr} and D_{recent})

end if

Output: Neutrosophic prediction Np_t at time t

In the above algorithm, $NUnc_t$ stands for neutrosophic uncertainty, and Np_t stands for neutrosophic prediction at time t .

possess various characteristics, including binary class, multi-class, temporal dependencies, and imbalanced data, making them well-suited for comparing different drift detection approaches and demonstrating the effectiveness of the proposed method, N-DD. Table 1 presents the data sets used in this study, which includes four real data sets for binary classification with temporal dependencies: Electricity (ELE), Ozone (OZO)^[27], GMSC^[28] (GMS), and Airlines (AIR)^[29]. Ozone (OZO) is an imbalanced data set with temporal dependencies, while Keystroke (KEY)^[30] is also included as a real multi-class data set with both temporal dependencies and imbalances. Additionally, Gas Sensor (GAS)^[27] and Covertypes (COV)^[31] are incorporated as real-world multi-class data sets with temporal dependencies and imbalances, respectively.

3 Experimental Work and Results

3.1 Data Sets Description

We conducted our experiments using widely-used data sets in concept drift research. All of the data sets used in this study are publicly available on open platforms such as OpenML, Kaggle, and the USP Data Stream Repository.

To evaluate the performance of data stream classification and drift detection. These data sets

Table 1 Data sets

Data set	Instances	Features	Classes
Electricity	45 312	8	2
Ozone	2 534	72	2
GMSC	15000	11	2
Airlines	37 900	7	2
Keystroke	1 600	10	4
Gas sensor	13 910	128	6
Covertypes	581 012	54	7

3.2 Parameters Setup

The purpose of this study is to analyze how current input data affects the properties of prediction

models, rather than solely focusing on the parameters of the models themselves. In order to accomplish this, we have conducted experiments to carefully select parameters that do not require unnecessary retraining and will ultimately produce the most accurate results. Therefore, the parameters configured thoughtfully, taking into account the trade-off between detection accuracy and the number of times the model needs to be retrained. Sklearn and the Scikit–Multiflow were used in the implementation.

The data stream was divided into three parts: 10% for training, 20% for validation, and 5% for data evaluation. The first 10% of data stream points were used for training the model. Validation for ADWIN was performed using the following 20% of the data stream. A change was detected when the absolute difference in means between two sub-windows of recent observations was greater than the constant α . The sub-windows width was experimentally fixed at 5% of the data stream, and the α value was set to 0.002 if no drift was detected in the validation data. When a change was detected, the last data points and corresponding labels equivalent to 5% of the overall data stream length were used for retraining.

The experimental setup was as follows:

The N-SVM parameter C was searched in the range of $[10^{-2}, 10^2]$ with a step size of 10^{-1} . In addition, $w = 0.125$.

The Kolmogorov-Smirnov test-based drift detector KSWIN does not require a true label to detect concept drift. Instead, it uses them for retraining the model in case of a drift. KSWIN requires a suitable α value to determine the sensitivity for concept drift detection^[32]. We optimized α using the same procedure as for N-DD. The scikit-multiflow library was used for KSWIN with a window size of 200 and `state_size` of 100.

The Dynamic Weighted Majority (DWM) is an ensemble method for concept drift. It maintains an ensemble of base learners, and the ensemble dynamically creates and deletes experts when a change is detected. Unlike KSWIN, DWM relies on continuously obtaining true labels over time in order to be able to detect drift. The parameters of DWM were set to typical values of $\beta = 0.5$, $\theta = 0.01$, and $p = 1$, as mentioned in Ref.[15].

3.3 Evaluation Metrics

In the real world, data often do not have clearly

defined drift points, making it challenging to accurately detect and measure the accuracy of drift detection. This is because the actual drift points are unknown, so we must rely on a decrease in prediction performance over time as an indication of drift^[33]. To address this issue, we use the Matthews and Correlation Coefficient (MCC) in this study, as it is able to handle real and imbalanced data^[34]. A higher MCC value indicates better performance on the classification data sets. The MCC is defined as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (4)$$

where, TP: True Positives, TN: True Negatives, FP: False Positives, FN: False Negatives.

3.4 Experimental Results and Discussion

The results of the experiments for all data sets are presented in Table 2, which displays the MCC values and the corresponding number of retraining for each real data set. It is evident that the proposed method, N-DD, outperforms the other strategies KSWIN, ADWIN, and DWM on data sets with two or multiple classes. It is noteworthy that while KSWIN and DWM show promising results for data sets (ELE, OZO, GMS, AIR, KEY, GAS) due to their adaptability to temporal dependencies, they require a significantly higher number of retraining compared to N-DD. This is because both methods, KSWIN and DWM, assume that the data have the same distribution. In contrast, the proposed approach, N-DD based on neutrosophic set, is better suited for real-world data. This is because it takes into account the nature of the data, which may not reflect the assumed distribution.

Similarly, for the OZO data set, both DWM and KSWIN show good results as they are designed to handle imbalanced and temporal dependencies. However, for the AIR data set, the most significant difference lies in the number of retrains required for N-DD compared to KSWIN and DWM. While N-DD only requires 15 retrains, KSWIN and DWM require 148 and 32 retrains, respectively. Similarly, for the GAS data set, N-DD requires 16 retrains, while KSWIN, DWIN, and DWM require 109, 45, and 41 retrains, respectively. In the COV data set, N-DD requires 32 retrains, while KSWIN, ADWIN, and DWM require a total of 141, 62, and 58 retrains, respectively. It is evident that N-DD requires the least number of retrains and

has the best predictive accuracy, this is because ADWIN needs all true labels for drift detection itself, while KSWIN and DWM rely on continuously obtaining all true labels over time to detect drift and make corresponding training updates. On the other hand, N-DD is capable of detecting drifts even when the true labels are unknown. This is particularly advantageous in real-world applications where obtaining

true labels is difficult and costly. Additionally, N-DD utilizes uncertainty estimates from entropy neutrosophic sets, allowing it to take into account the impact of current input data on the prediction model's properties, rather than solely detecting changes in the input data. By considering the dependence of the prediction model's properties on uncertainty estimates, we can avoid unnecessary retraining.

Table 2 MCC values with number of retraining

Data sets	N-DD	No. of training	KSWIN	No. of training	ADWIN	No. of training	DWM	No. of training
ELE	0.595	12	0.548	92	0.412	44	0.532	42
OZO	0.482	14	0.481	101	0.479	50	0.481	24
GMS	0.496	15	0.467	98	0.472	21	0.453	31
AIR	0.598	15	0.571	148	0.489	49	0.591	32
KEY	0.612	22	0.623	117	0.592	71	0.592	37
GAS	0.514	16	0.511	109	0.513	45	0.498	41
COV	0.846	32	0.837	141	0.834	62	0.837	58

In contrast, N-DD performs better than KSWIN, DWIN, and DWM on binary or multiple classes, in terms of both the number of retrainings, and predictive performance with minimal differences in the latter. The reason N-DD can solely detect changes in the input data is by considering the dependence of the prediction model's properties on uncertainty estimates. This allows us to avoid unnecessary retraining.

Table 3 presents the accuracy of the proposed method, N-DD. It shows the prediction performance of N-DD with the highest mean accuracy calculated at the lowest (scale 1) neutrosophic entropy, which represents uncertainty.

Table 3 Accuracy of proposed method on the data sets

Data sets	N-DD Max acc. (%)
ELE	89
OZO	47
GMS	59
AIR	92
KEY	91
Gas	83
COV	95

The proposed method, N-DD has been proven to be highly accurate in producing results for data sets that contain multiple or binary classes. For example, it has achieved an impressive accuracy rate of 95% when applied to the COV data set, which is a large data set with multiple classes. Additionally, it has also been successful in predicting for binary classes in the AIR

dataset, with an accuracy rate of 92%. This can be attributed to the incorporation of uncertainty estimates from entropy neutrosophic sets in the N-DD method. This allows the method to effectively consider the influence of current input data on the prediction model, regardless of the type of data set.

Fig. 2 illustrates the accuracy of the proposed method when applied to various data sets. The x -axis represents the level of uncertainty, determined by sorting the instances from lowest to highest based on their entropy values.

As shown in Fig. 2, the accuracy of classifying real data sets is dependent on the level of uncertainty. This means that as uncertainty increases, the accuracy decreases. This finding supports our research methodology, which considers both the current input data and the indeterminacy presented by neutrosophic sets in the properties of the production model. This approach differs from simply detecting changes in the input data. As a result, we have developed a solution for the indeterminate true label problem by measuring the level of uncertainty in the prediction made by N-SVM using neutrosophic set theory.

Through comparative analysis with other methods, N-DD has been proposed as a method for detecting drift with indeterminate true labels. This method has been found to outperform others for several reasons. While ADWIN is able to obtain accurate true labels in real time and use them to identify drift and train accordingly, N-DD is able to detect drift even with indeterminate true labels.

Additionally, KSWIN assumes that the data are identically distributed, which may not hold true in real-world datasets and can affect the performance of the method. Furthermore, DWM requires retraining of the prediction model, which can be computationally expensive, especially for large datasets. Moreover, DWM also assumes that the data are independent and identically distributed, which may not hold true in

real-world datasets. In contrast, N-DD is less computationally expensive compared to other methods as it has the ability to avoid unnecessary retraining by considering the impact of current input data on the properties of the prediction model. This is made possible by the use of Neutrosophic theory, which is well-suited for real-world data.

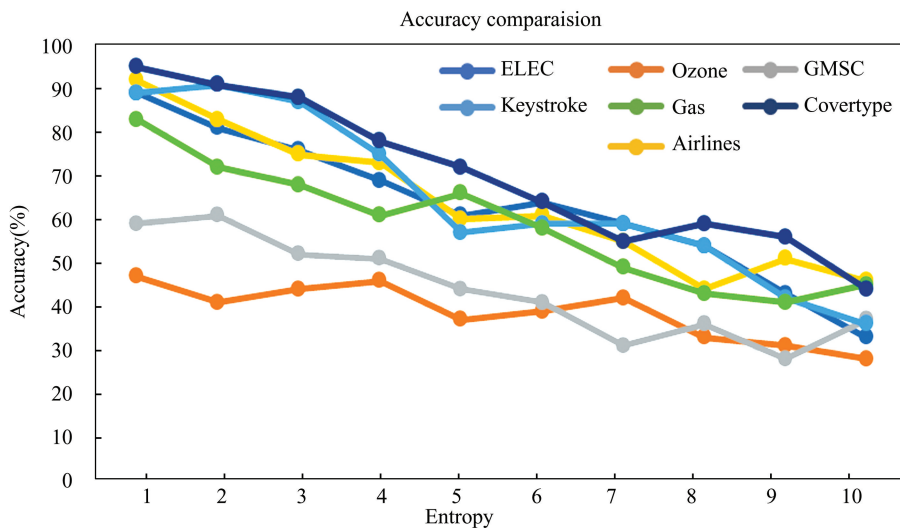


Fig. 2 Relation between prediction accuracy and uncertainty

The average running time of the methods studied was compared to that of the proposed method N-DD by applying them to real data sets using a Core-i7-8550U processor with 8 GB of memory.

After conducting our research, we have determined that the N-DD method has a lower average running time (0.96 s) compared to the KSWIN (1.1 s) and DWM (1.3 s) methods. However, we also observed that the ADWIN method has an even better average running time of 0.91 s, slightly surpassing N-DD's. This is due to our experimental tuning of parameters to enhance its effectiveness. As a result, N-DD required a smaller number of training, making its average running time acceptable, while the other methods required a larger number of training.

From these experiments we highlight the following results.

1) N-DD is capable of detecting drifts even when the true labels are unknown. This is particularly advantageous in real-world applications where obtaining true labels is difficult and costly.

2) One of the main strengths of N-DD is its use of uncertainty estimates from entropy neutrosophic sets. This enables the algorithm to not only identify

changes in the input data, but also consider the effects of these changes on the prediction model's characteristics. This approach helps to prevent unnecessary retraining, making N-DD more efficient and cost-effective.

3) N-DD has an acceptable average running time compared to other methods, such as KSWIN and DWM. This can be attributed to the proposed method, which requires a smaller number of retrainsings.

In the N-DD approach, the reliability of drift detection may be compromised if the uncertainty estimates are unreliable. This is often a common issue with other methods as well, as poor data quality can lead to inaccurate results. However, this is a rare occurrence and can be eliminated by measuring the uncertainty using neutrosophic entropy. This allows for the consideration of the nature of the data, which may not accurately reflect the assumed distribution.

4 Conclusions

In this study, we utilize neutrosophic set theory to develop an efficient algorithm, called N-DD, for detecting concept drift. This algorithm is specifically

designed for situations where the true labels are indeterminate, and takes into account the impact of the current input data on the properties of the model. To achieve this, our proposed method reads data into Windows and calculates a set of values based on the concept of neutrosophic membership. These values are then used in the N-SVM. We measure the uncertainty values of the corresponding prediction made by N-SVM as neutrosophic entropy. These values are then used as input for the ADWIN change detector. When a drift is detected, the prediction model is retrained by including only the most recent instances with the original training data set. This is a crucial adaptation for real-world drift detection, where true labels are expensive, indeterminate, and not always readily available. Unlike most state-of-the-art algorithms that require access to the entire set of true labels, our method is based on the theory of neutrosophic, which deals with indeterminate values. This makes our method highly suitable for detecting drift in situations where true labels are limited and indeterminate. Our experiments were conducted on seven real data sets, and the results demonstrate that N-DD outperforms other methods, such as ADWIN, KSWIN, and DWM, in detecting drift with the least amount of retraining needed. Additionally, the N-DD approach had the lowest average running time compared to KSWIN and DWM, with only a minimal difference from ADWIN.

In future work, we plan to enhance the N-DD method by incorporating a study of neutrosophic concepts and deep learning techniques. Furthermore, it is important to thoroughly evaluate the use of N-DD in cases where different types of indeterminacies may arise, such as uncertainty, conflicting information, and vague data. One approach to addressing these indeterminacies is through the use of refined neutrosophic logic^[35], which specifically deals with inherent indeterminacy and allows for a more comprehensive analysis of uncertainty.

References

- [1] Gama J, Zilobaite I, Bifet A, et al. A survey on concept drift adaptation. *ACM Computing Survey*, 2014, 46(4): 1–37. DOI: 10.1145/2523813.
- [2] Zenisek J, Hlölzinger F, Affenzeller M. Machine learning based concept drift for predictive maintenance. *Computers & Industrial Engineering*, 2019, 137(1): 106031. DOI: 10.1016/j.cie.2019.106031.
- [3] Shervashvili S. Online learning and online convex optimization. *Now Foundations and Trends*, 2012, 4(2): 107–194. DOI: 10.1561/22000000018.
- [4] Lu J, Liu A, Dong F, et al. Learning under concept drift: A review. *Transactions on Knowledge and Data Engineering*, 2019, 31(12): 2346–2363. DOI: 10.1109/TKDE.2018.2876857.
- [5] Leonardo L, Minkunand X Y. DDD: A new ensemble approach for dealing with concept drift. *Transactions on Knowledge and Data Engineering*, 2012, 24(4): 619–633. DOI: 10.1109/TKDE.2011.58.
- [6] Krawczyk B, Minku L L, Gama G, et al. Ensemble learning for data stream analysis: A survey. *Information Fusion*, 2017, 37(1): 132–156. DOI: 10.1016/j.inffus.2017.02.004.
- [7] Zliobaite L, Bifet A, Pfahringer B, et al. Active learning with drifting streaming data. *IEEE Transactions on Neural Networks and Learning Systems*, 2014, 25(1): 27–39. DOI: 10.1109/TNNLS.2012.2236570.
- [8] Janardan, Shikha M. Concept drift in streaming data classification algorithms, platforms and issues. *Proceeding Computer Science*, 2017, 122: 804–811. DOI: 10.1016/j.procs.2017.11.440.
- [9] Krawczyk B, Pfahringer B, Woźniak M. Combining active learning with concept drift detection for data stream mining. *International Conference on Big Data (Big Data)*. Piscataway: IEEE, 2018: 2239–2244. DOI: 10.1109/BigData.2018.8622549.
- [10] Iwashita A S, Papa J P. An overview on concept drift learning. *IEEE Access*, 2019, 7: 1532–1547. DOI: 10.1109/ACCESS.2018.2886026.
- [11] Wang P F, Jin N L, Davies D, et al. Model-centric transfer learning framework for concept drift detection. *Knowledge-Based Systems*, 2023, 275: 1107075. DOI: 10.1016/j.knsys.2023.110705.
- [12] Sebastião R, Fernandes J M. Supporting the Page-Hinkley test with empirical mode decomposition for change detection. *International Symposium on Methodologies for Intelligent Systems, Lecture Notes in Computer Science*. Berlin: Springer, 2017, 10352: 492–498. DOI: 10.1007/978-3-319-60438-1_4.
- [13] Bifet A, Gavalda R. Learning from time-changing data with adaptive windowing. *Proceedings of the Seventh SIAM International Conference on Data Mining*. Philadelphia: SIAM, 2007: 443–448. DOI: 10.1137/1.9781611972771.42.
- [14] Bugnicourt A, Mokadem R, Morvan F, et al. An error-based measure for concept drift detection and characterization. *17th International conference Learning and Intelligent Optimization*. Berlin: Springer, 2023: 239–253. DOI: 10.1007/978-3-031-44505-7_17.
- [15] Kolter Z, Maloof M. Dynamic weighted majority: An ensemble method for drifting concepts. *Journal of Machine Learning Research*, 2007, 8: 2755–2790. DOI:

10.5555/1314498.1390333.

- [16] Ayad O. Learning under concept drift with support vector machines. *Artificial Neural Networks and Machine Learning*. Berlin: Springer, 2014; 587 – 594. DOI: 10.1007/978-3-319-11179-7_74.
- [17] Nyati A, Bhatnagar D, Panwar A. Analyzing performance of classification algorithms on concept drift data streams. *International Journal of Computer Applications*, 2017, 159 (9): 13 – 17. DOI: 10.5120/IJCA2017913065.
- [18] Dos Reis D M, Flach P, Matwin S, et al. Fast unsupervised online drift detection using incremental Kolmogorov-Smirnov test. *KDD 16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: AC, 2016; 1545–1554. DOI: 10.1145/2939672.2939836.
- [19] Baier L, Schlor T, Schoffer J, et al. Detecting concept drift with neural network model uncertainty. *Proceeding of the 56th Hawaii International Conference on System Sciences*, 2023; 836 – 844. arXiv: 2107.01873 [cs.LG]. DOI: 10.48550/arXiv.2107.001873.
- [20] Li J P, Yu H, Zhang Z Y, et al. Concept drift adaptation by exploiting drift type. *ACM Transactions on Knowledge Discovery from Data*, 2024, 18 (4): 1 – 22. DOI: 10.1145/3638777.
- [21] Oikarinen E, Tiittanen H, Henelius A, et al. Detecting virtual concept drift of repressors without ground truth values. *Data Mining Knowledge and Discovery*, 2021, 35: 726–747. DOI: 10.1007/s10618-021-00739-7.
- [22] Samarandache F. *Introduction to Neutrosophic Measure, Neutrosophic Integral, and Neutrosophic Probability*. Neutrosophic Sets and Systems. Columbus: Education Publishing, 2013. DOI: 10.5281/zenodo.8843 License CC BY 4.0.
- [23] When J, Cheng H D. A novel neutrosophic logic SVM (N-SVM) and its application to image categorization. *New Mathematics and Natural Computation*, 2013, 9(1): 27–42. DOI: 10.1142/S1793005713500038.
- [24] Elhassouny A, Idbrahim S, Samarandache F. Machine learning in neutrosophic environment: A survey. *Neutrosophic Sets and Systems*, 2019, 28(1). 58–68.
- [25] Grulich P M, Saitenmacher R, Traub J, et al. Scalable detection of concept drifts on data streams with parallel adaptive windowing. *Open Proceedings of the 21st International Conference on Extending Database Technology (EDBT)*, 2018; 477 – 480. DOI: 10.5441/002/EDBT.2018.51.
- [26] Majumdar P. On new measures of uncertainty for neutrosophic sets. *Neutrosophic Sets and Systems*, 2017, 17(1) 50–57.
- [27] Souza V M A, Dos Reis D M, Maletzke A, et al. Challenges in benchmarking stream learning algorithms with real-world data. *Data Mining and Knowledge Discovery*, 2020, 34(1): 1805 – 1858. DOI: 10.1007/s10618-020-00698-5.
- [28] GMSC. <https://www.kaggle.com/competitions/GiveMeSomeCredit>.
- [29] Airlines (AIR), https://ijs.si/elena_ikonovska/data.html.
- [30] Keystroke (KEY), <https://www.kaggle.com/datasets/carnegielab/keystroke-dynamics-benchmark-data-set>.
- [31] Killourhy K S, Maxon R A. Comparing anomaly detectors for keystroke dynamics. *Proceedings of the 39th Annual International Conference on Dependable Systems and Networks*. Piscataway: IEEE, 2019; 125–134. DOI: 10.1109/DSN.2009.5270346.
- [32] Wang Z, Wang W. Concept drift detection based on Kolmogorov – Smirnov test. *Artificial Intelligence in China*. Berlin: Springer, 2020, 572; 273–280. DOI: 10.1007/978-981-15-0187-6_31.
- [33] Chicco D, Straoivoitov V, Jurman G. The benefits of the Matthews Correlation Coefficient (MCC) over the Diagnostic Odds Ratio (DOR) in binary classification assessment. *IEEE Access*, 2021, 9; 47112–47124. DOI: 10.1109/ACCESS.2021.3068614.
- [34] Jurman G, Riccadonna S, Furlanello C. A comparison of MCC and CEN error measures in multi-class prediction. *PLoS ONE*, 2012, 7(8): e41882. DOI: 10.1371/journal.pone.0041882.
- [35] Samarandache F. n-Valued refined neutrosophic logic and its applications to Physics. *Progress in Physics*, 2013, 9; 143 – 146. <https://fs.unm.edu/RefinedNeutrosophicSet.pdf>.