Labelled Classifier with Weighted Drift Trigger Model using Machine Learning for Streaming Data Analysis

Original Scientific Paper

Gollanapalli V Prasad
GNI Technical Campus,
Hyderabad, Telangana
prasad.venkata8@gmail.com

S Krishna Mohan Rao
Sidartha Institute of Enginiring Techology,
Hydeabad, Telangana
Krishnamohan6@yahoo.com

Kapil Sharma
Computer Science and Engineering,
Amity University,
Gwalior, India
ksharma@gwa.amity.edu

Venkatadri M
Amity School of Engineering & Technology (ASET)
Gwalior, India
vmarriboyina@gwa.amity.edu

Rama Krishna B
GNI Technical Campus,
Hyderabad, Telangana
ramakrishnabasude@gmail.com

Abstract *– The term "data-drift" refers to a difference between the data used to test and validate a model and the data used to deploy it in production. It is possible for data to drift for a variety of reasons. The track of time is an important consideration. Data mining procedures such as classification, clustering, and data stream mining are critical to information extraction and knowledge discovery because of the possibility for significant data type and dimensionality changes over time. The amount of research on mining and analyzing real-time streaming data has risen dramatically in the recent decade. As the name suggests, it's a stream of data that originates from a number of sources. Analyzing information assets has taken on increased significance in the quest for real-time analytics fulfilment. Traditional mining methods are no longer effective since data is acting in a different way. Aside from storage and temporal constraints, data streams provide additional challenges because just a single pass of the data is required. The dynamic nature of data streams makes it difficult to run any mining method, such as classification, clustering, or indexing, in a single iteration of data. This research identifies concept drift in streaming data classification. For data classification techniques, a Labelled Classifier with Weighted Drift Trigger Model (LCWDTM) is proposed that provides categorization and the capacity to tackle concept drift difficulties. The proposed classifier efficiency is contrasted with the existing classifiers and the results represent that the proposed model in data drift detection is accurate and efficient.*

Keywords: *Data Clustering, Data Classification, Data Stream Mining, Streaming Data, Drift Detection, Drift Trigger Model, Labelled Classifier*

1. INTRODUCTION

With the help of Data Streaming Mining (DSM), knowledge structures can be gleaned from rapidly changing [1], continuous data streams. Streams of data can be read only once or a few times using limited computational and storage resources in many applications of data stream mining [2]. To forecast the class or values of new instances in a data stream, many common data mining applications rely on prior knowledge of class membership [3] or values of earlier data stream instances [4]. It is possible to learn this prediction model from labelled samples using machine learning techniques in an automated manner. Concepts from the incremental learning area are frequently used to deal with structural changes, online learning and real time requirements [5]. Non-stationary environments, where the distribution of instances or their labelling criteria may change over time [6], may need a change in the purpose of a prediction, such as which class to predict, or what target value to predict [7]. Concept drift is the name given to this phenomenon. Data stream mining relies heavily on the detection of concept drift. Additionally, when implementing machine intelligence to streaming data [8], there are other issues that need to be addressed, such as partially and delayed labelled data [9], concept drift recovery, and temporal dependencies [10].

Data repositories on the World Wide Web are expanding at a quicker rate than ever before, using real-time web applications [11]. Apps have started to use data mining techniques to analyse the massive amounts of data, in order to identify trends or patterns that may be used to make better business decisions [12], as the amount of data grows dramatically. Real-time decision making is becoming increasingly crucial in computer science and engineering, and data mining is becoming a major study topic. This is because of data mining techniques, which are able to successfully deal with the storage and processing constraints [13]. It has recently been proposed to use data mining techniques to handle streaming data, which is a difficult task. Data streams can be thought of as a continuous stream of training instances that arrive from one or more sources at a high rate of speed [14]. Mining continuous real-time streaming data with reasonable performance is a process known as data stream mining [15] [16].

Data stream mining is essential in a wide range of real-time applications, including detecting attacks, stock market monitoring, and web personalization [17]. It is challenging to build strategies for real-time mining of streaming data because it is so time-consuming [18]. One or more scans of the data may be required to convert it into information in traditional Online Analytical Processing systems (OLAP) [19]. Due to the particular properties of data stream mining, this is not possible. Because of this, standard data mining techniques must be reworked in order to manage data that is constantly flowing from several sources through the network [20]. It has become increasingly important in recent years to process data streams for uncover new information because such data is increasingly accessible via rich internet applications. In developing novel strategies to handle streaming data [21], there are two major obstacles. Fast mining methods for streaming data are the first problem, while the challenge today is to detect data distribution [22] and evolving concepts in an ever-changing environment. The process of streaming data drift detection is represented in Figure 1.

It is important to note that data mining is a significant part of data management. Offline data processing is at the heart of the majority of data mining systems [23]. It is common practice to train predictive models with a pair of data sets. Models trained on previously unseen data are used to anticipate the output of fresh data. It is impossible to handle streaming data concurrently because of the volume of data that is generated on a daily basis [24]. In order to fit all of this data into the machine's main memory, the only viable option is to use online data processing. It is possible to train predictive models by continuously updating them or by guaranteeing that the model retains its accuracy through the use of batches of data [25].

It's possible that the data distribution will shift over time, resulting in settings conducive to concept drift in ever-changing situations. Concept drift occurs when the conditional probability of varying output changes despite the input remaining unchanged [26]. A famous example of real concept drift is when a user's degree of interest shifts as they are following a news stream online [27]. For example, despite the fact that the distribution of a news item that is frequently shared may remain the same, the conditional probability [28] of interesting news items for the user may change. It is possible that the future predictions online are responding to idea drifts as a result of the adaptive learning process.

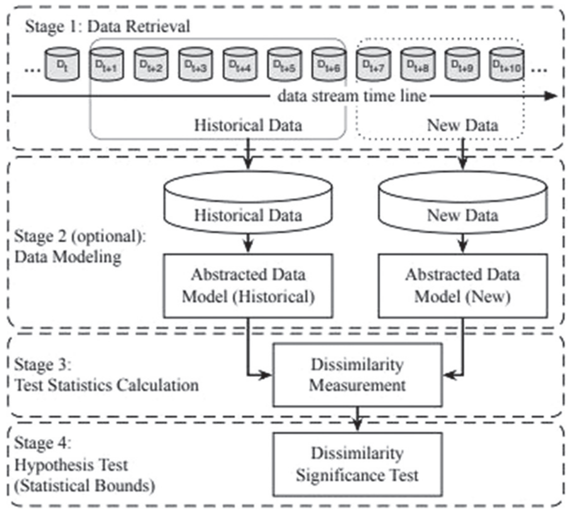

Fig. 1. Drift Detection Process

Predictive models that are able to effectively characterise the actual concepts concealed in data streams are the goal of DSM. As time goes on, new concepts are introduced, and rare data samples are discovered. This is the fundamental problem that arises during this process. The phenomena are referred to as notion drift. Class borders can be actual or virtual, depending on how the data is distributed in relation to those boundaries. Depending on how long it's been going on, data can be classified as either sudden or progressive. A continually changing environment necessitates that previously constructed models be constantly updated. There are two major ways to approach this task. As a blind adaptation, we constantly update our model independent of the stream's real condition.

A significant limitation on every unsupervised adaptive machine learning model, the restricted labelling plan could quickly make this technique unfeasible. Labelled instances are necessary for updating an algorithm that exists. It's impossible to expect that a corporation will have time or money to give annotations for all of the data that comes in, since collecting the accurate labels entails both a cost and time. There is no reason to waste time and money on fresh data points if there has been no change in the situation. The second strategy aims to update a system only when it is truly necessary, such as after a drift has occurred. A drift detector is a critical part of every such solution. In order to keep track of a stream's current condition, it monitors it using a model and alerts the user when it changes. As a result, we have the ability to better manage our budget expenditure. The model can be kept stable, and we can use it when drift has been triggered and the model has become obsolete.

2. LITERATURE SURVEY

Khamassi et al. [1] proposed a model that take account a subcategory that includes unsupervised approaches, however none of those works are unique to unsupervised drift detection. Only unsupervised detection techniques are covered in their research. To put it another way, the proposed taxonomy is innovative because it focuses on features of detectors developed for unsupervised situations. The classification of autonomous drift detection systems given by Fernández et al. [2] is divided into three broad classes. In the first category, approaches that monitor error rates are considered, such as the Drift Detection Method (DDM), the Early Drift Detection Method (EDDM) and others. Distance measurements are used to estimate the resemblance between prior and present data distributions in the second class. There are further ways that employ several hypothesis tests to look for changes in an idea. No one class is specific to unsupervised methods but Cano et al. [3] presented an unstructured or semi-supervised approaches that fit into the last two classifications. According to the taxonomy established by Idrees et al. [5], autonomous detectors may fall within the two last categories.

According to Fernandez et al. [6], concept drift detection systems can be classified as either performance-based or data distribution-based. Some error-related parameter is continuously monitored in performance-based methods such as precision and recall. Generally speaking, a drop in an important statistic indicates a drift. These methods are inapplicable to unsupervised tasks since true labels are required to estimate mistakes. Distribution-based techniques, on the other hand, use metrics like location, density, and range to monitor distribution. This category includes both supervised and unsupervised methods. Barddal et al. [8] presented a categorization of approaches based on these two classes, and remark that this group contains these two methods.

Lin et al. [9] proposed classification approaches that are used under the guidance of an investigator. As a starting point, the techniques are divided into four main categories: statistical, window-based, and block-based methods, and incremental-based methods. For example, the Cumulative Average and the Page–Hinckley Test are examples of statistical detectors and drift detection methods are all part of this group. Approaches that monitor the accuracy of the classifier in a window are part of a second class that includes window-based methods. Methods based on monitoring ensemble classification accuracy differ in how data are processed as a response to drift, as shown by the last two classes. Methods in the class retrain classifications on chunks or blocks of examples, whereas methods in the last class retrain progressively with each new arrival of a new classifier.

"How are data processed?" "How is learning processed?" "How is concept drift monitored?" "How is concept drift handled?" and "What are the performance criteria?" are some of the questions posed by Junior et al. [11]. There are a number of different ways to answer this question, and they're categorised according the type of analysis they employ. The classification of unsupervised approaches is further refined into resemblance in time, similarities in space, and model complexity metrics by the authors. The first has to do with the differences in distribution between two timestamps, which are typically found using hypothesis testing, all used distance functions such Euclidean, Heterogeneous Euclidean-overlap, and Mahalanobis distances to monitor the evolution of data distribution in space. Changes in structural models and/or parameters are the focus of the last group.

According to Montiel et al. [12], there are just a few studies that take into account the temporal dependence of concept drift detection in the literature. This criterion is not included in our taxonomy because the author focused on the more modern aspects of supervised drift detection systems. Accordingly, all of the strategies mentioned here can be classified as belonging to the similarity-in-space category. Because of this, our taxonomy has been expanded to include more ways. Lastly, the final group is algorithm-dependent, but the works mentioned in this article can be produced using any machine learning technique

Even if conceptual evolution is solved, it still has a high false alarm rate for particular datasets and therefore can distinguish between distinct novel class problems. Pesaranghader et al. [15] came up with a way to deal with idea evolution induced by the emergence of new classes. Additional classifier sets are added to the main one. As soon as the primary classification set and the related classifier set establish that an incoming instance is an outlier, it is briefly kept in a buffer. The new class identification module is invoked for detection until there are enough examples in the buffer. An instance of a novel class will be indicated if it exists. According to the literature, the feature translation technique is proposed to handle the evolution of streaming data features. To address the issue of feature evolution in the data stream, the classic data stream integrating classifier is paired with a new class detection method.

Data mining relies on traditional learners, which are well-known classifiers, to meet their stream mining needs [16]. A forgetting process and other characteristics of an online learner are present. Naive Bayes, Neural network, and Decision tree rules are some of the techniques employed in [17]. When dealing with data that changes over time, a technique known a windowing technique may be used. There is a restriction on how many examples can be presented. FISH, ADWIN, and weighted windows are all examples of windowing strategies. Any learner can be adapted to evolving stream data using drift detector algorithms. When a conceptual model begins to stray, an alert goes off, prompting the learner to make corrections. It is possible to discover drifts in conceptual thinking when utilizing DDM and EDDM [18].

A number of drift instances cannot be accommodated by the ensemble approach utilized in [19]. This problem can be better handled with adaptive classifiers. Recently, a number of studies have focused on adaptable learning methods using ELM-based single classifiers and ensembles for CD adaptation. A good example is Incremental Data Stream ELM, which trained its classifier in a gradual manner. A dynamic number of hidden neurons and the selection of certain Activation Layer enhances the model's performance in this method. This technique, on the other hand, soley considers stream data in the case of slow drift [20]. Current ML models aren't robust enough to operate in a nonstationary environment because of the necessity for significant improvements in accuracy and adaptability.

Researchers have focused on Concept Drift for the past decade since it has a wide range of vital applications. There are numerous studies on Concept Drift detection and adaptation that are well-researched, however there isn't any integrated information in the literature. Olorunnimbe et al. [21] provide an overview of Concept Drift Learning in a new survey, focusing on adaptation and detection strategies as well as CD datasets utilized in previous works. Researchers did not conduct a comparative examination of existing adaptation and detection tools or protective research directions for CD concerns in this study.

3. PROPOSED MODEL

Traditional machine learning applications use batch learning algorithms to analyse static datasets. Batch learning is a type of learning strategy in which all of the training data is available at once. Depending on the algorithm, the information may be disclosed once or numerous times. A variety of factors preclude batch learning from working with data streams. Since examples arrive sequentially and continually in a distributed method, common approaches have all of the data at their fingertips. A stream-mining algorithm must be built to function with just one pass of data, unlike batch and multi-pass instructional strategies.

The data distribution is assumed to be constant in classic machine learning approaches. An inherent property of a stream is that it is subject to change throughout time. This has rendered batch processing learning methods outdated. A data stream's distributing of instances is said to be 'drifting' when the term 'concept drift' is used. Over time, ideas evolve, but the speed at which they do so varies. Some instances can become obsolete because their dispersion no longer truly depicts their class classification; in other circumstances, it can be a problem. Models must be able to forget previous examples in order to keep up with new ideas once the concept has rambled.

The proposed classifier handles data drifts effectively. It is a classic learner modelled for stationary data mining that has the characteristics of an online learner mechanism. When it comes to classifier, a polling procedure is used to assemble the results. The classification accuracy is superior to that of a single classifier's combined decision. Adapting to new concepts is a natural process for them because of their modularity. Concept drift is a result of data distribution changing over time in dynamic or non-stationary situations. It is possible to fast adapt the concept drifts by saving concept descriptions, which may be re-examined and repurposed after words. Adopting an adaptive learning strategy bi therefore necessary when dealing with data in non-stationary settings. To maintain correctness, an existing model must be updated when conceptual drift is discovered.

An effective drift detector must be capable of distinguishing between real changes and false alarms, and this balance must be maintained. Using local output of feature subspace-based judgements, it is necessary to assess whether the entire data stream is affected by a concept drift. As long as the choice is made by a simple majority, it may not be as sensitive as one might prefer.

The drift data is considered for analysis and for detecting change of data. The records are initially analysed from the input and the initial analysis is performed as

The proposed model drift detection procedure is represented in Figure 2. The figure represents the flow of drift detection in streaming data.

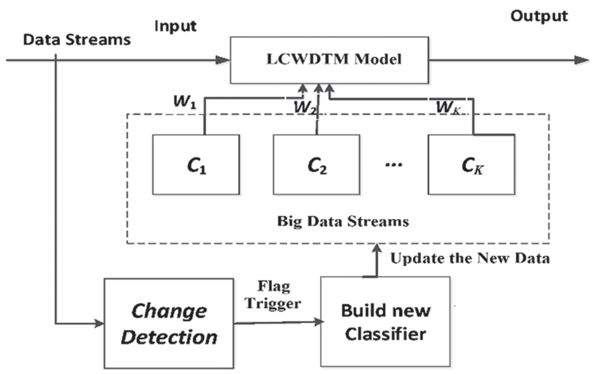

Fig. 2. Proposed Framework

$$Data\_Set[M] = \sum_{i=1}^{N} T_N \left(Record(row(i) + V_i^n(value))\right) \quad (1)$$

The data will be clustered based on the similarity values of the records and the cluster sets are monitored for drift detection by allocating labels to the determined attributes in the record set. The clustering process is applied as

$$\begin{aligned} &Cluster\_Set(Data\_Set[M]) = \\ &\sum_{i=1}^{M} \max\left(\frac{sim(Record(i), Record(i+1)}{count(Records(Data\_Set[M]))}\right) \\ &+ \sum_{j=1}^{M} comp(V, V+j) \end{aligned} \quad (2)$$

The record features are considered and then labelling is perfomed for each and every variable vector so that change in the value will be triggered based on the label it is easy to identify the drift in the data. The labelling procedure of features are performed as

$$Labelled(Cluster_{set[M]}) = \left(\sum_{i\in M}\sum_{i=1}\frac{\min(\text{Cluster\_Set(Record(i),Cluster\_Set(Record(i+1)}}{\text{sizeof(Cluster\_Set[M])}} + random(V)\right) \quad (3)$$

The weights for the variables are assigned for accurate drift detection so that clusters sets are verified based on weights each cluster set is verified for drift data in sequence. The weights are allocated as

$$Weight\_Alloc(Labelled(R(M)))_N = \left(\frac{comp(\text{Prev}_{\text{Val(V)}},\text{New}_{\text{val(V)}})>(MDiff(Pev\_Val))^N}{\text{size(Labelled\_Set)}}\right) \quad (4)$$

Moving Average Convergence Divergence (MACD) is an indicator that shows the trend of the streaming data changes, and is based on Exponential Moving Average (EMA). MACD is calculated as

$$MACD(DSP) = EMA(Pr\,e\,v(V), New(V), L) \quad (5)$$

Here EMA signifies the more recent updated price based on the length L that represents the number of days values considered in the analysis of drift detection. Based on the MACD calculated, the EMA is updates as

$$EMA(New(V), L) = New(V) - Pr\,e\,v(V) + MACD \quad (6)$$

Relative Strength Index (RSI) is an oscillator-based indicator that focuses on the strengths and weaknesses of the data changes and updates weights to the variables. It is formulated as follows:

$$RSI = 100 - \frac{100}{1+\frac{average\ gain}{average\ loss}} + \Sigma_{j=1}^{Q} weight\_alloc_j \left(\Sigma_{i=1}^{N} weight\_alloc(Cluster\_Set)\right) \quad (7)$$

The labelling is based on the weight drift trigger that automatically triggers the variable change prediction update based on the weights allocated for variables. The labelling is performed as

$$Labelling(W_{Ut}) = \frac{1}{MACD}\Sigma_{i=1}^{p}\begin{cases} max \\ 0 \end{cases} \left[\frac{1}{1+abs(RSI(Pr\,ev\_Loss)_N^{EMA})}, \frac{1}{1+min(MDiff(Ut(i,j)))_{Dt}^{Pr\,ev\_Loss}}\right] \quad (8)$$

The drift detection is performed and the drift identified set of clusters are maintained for considering the updated drift dataset that is used for further analysis. The updated drift detected set is maintained as

$$Drift\_Detect - Set(Labelled(\mathrm{Re}\,cord(M))) = \frac{\max(RSI) + \min(Labelled(Weight\_Alloc(i)) + \max(MACD)}{size(Labelled\_Set(M))} \quad (9)$$

4. RESULTS

The proposed stock market prediction model is implemented using python and evaluated in Google Colab. The dataset is available from the link https://www1.nseindia.com/products/content/equities/indices/historical_index_data.htm. This dataset provides historical information as well as real time data based on the time range provided. The Proposed Labelled Classifier with Weighted Drift Trigger Model (LCWDTM) is compared with the traditional Machine Learning Algorithm for Continuous Concept Drift Detection (MLA-CCDD) Model. The proposed model is compared with the traditional models in terms of Data Records Analysis Time Levels, Data Clustering Accuracy Levels, Drift Detection Time Levels, Drift Trigger Accuracy Rate, Classifier Accuracy for Drift Detection and Error Rate.

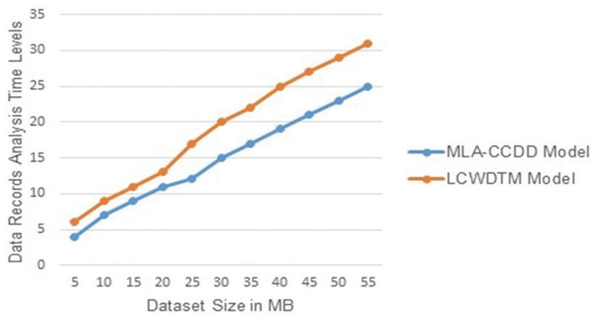

Fig. 3. Data Records Analysis Time Levels

Analysing raw data is the act of identifying and synthesising useful information and drawing conclusions from it. It's during this phase that a dataset is organised, transformed and modelled by an investigator or data analyst. In the proposed model, streaming data is considered and the data records are analysed for detection of drift in data. The data records analysis time levels of the proposed and traditional models are shown in Figure 3.

Data Clustering is a machine learning technique for discovering and grouping related data points in huge datasets without regard for the outcome. Clustering is a technique for organising data into patterns that are easier to comprehend and manipulate. Clustering is the process of splitting a population or set of data points into many groups so that measured values in the same category are more similar than data points in other groups. The goal is to separate groups with similar characteristics and assign them to clusters. The data clustering of streaming data accuracy levels of the proposed and existing models are represented in Figure 4.

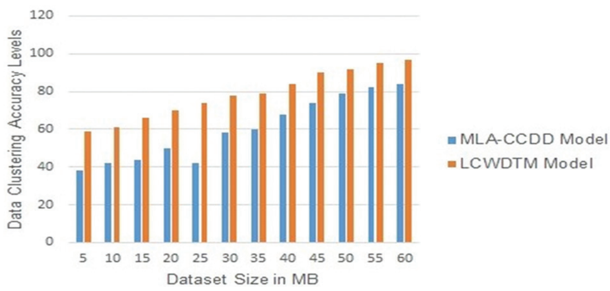

Fig 4: Data Clustering Accuracy Levels

To determine whether a data real configuration has drifted from its expected configuration, drift detection is an essential tool. By monitoring the statistical features of data, the model's predictions and their relationship with other parameters, one can detect data drifts. Modern data architectures result in unforeseen and unrecorded changes in data format, interpretation, and infrastructure. However, data drift has the potential to uncover new avenues for data utilisation. The drift detection time levels of the proposed and traditional models are indicated in Figure 5.

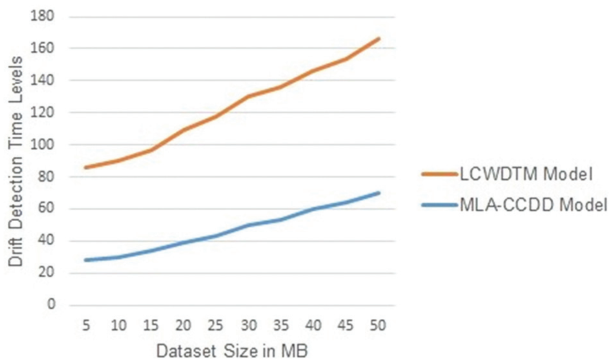

Fig. 5. Drift Detection Time Levels

Prescriptive modelling and machine learning might suffer from "concept drift," which refers to a change in statistical features of the target variable. As time passes, the accuracy of the projections deteriorates, which is problematic. The proposed model triggers a flag when there is a data drift observed. The drift trigger accuracy levels of the proposed model is high compared to existing model. The drift trigger accuracy rate of the proposed and traditional models are represented in Figure 6.

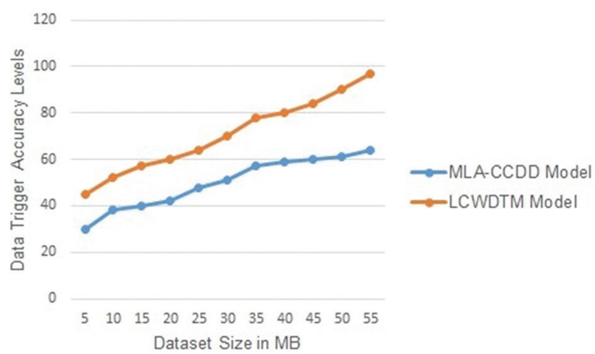

Fig. 6. Drift Trigger Accuracy Rate

A data stream is a collection of data that comes from a variety of sources. Real-time analytics necessitates the ability to analyse data as it moves through a system. Traditional mining methods have been ineffective since the nature of data has changed. Data streams have additional hurdles, such as memory and running time limits, as well as a single scan of the data. When a dataset's notion or distribution changes over time, it is referred to as "concept drift".

Due to idea drift even when data is stationary, dealing with this problem becomes more difficult for models and classifiers in data streams. The classifier designed will effectively detects data drifts in handling big streaming data. The proposed classifier achieves an accuracy of 98% in drift detection that represents that the proposed model is efficient that the traditional methods. The classifier accuracy of the proposed and traditional models are shown in Figure 7.

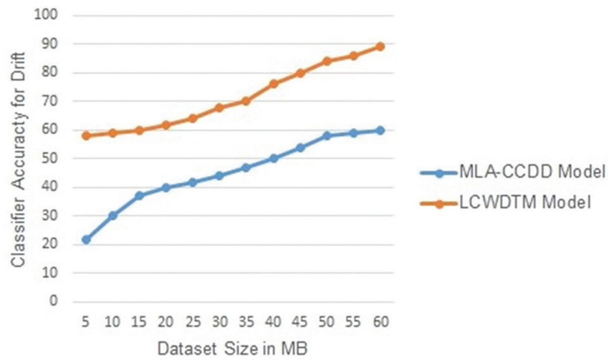

Fig. 7. Classifier Accuracy for Drift Detection

The average number of times we get our target's class classification wrong is known as the error rate. True positives and true negatives divided by the total number of true positives, true negatives, false positives, and false negatives is the true positive/negative ratio. The Error rate of the proposed model is high that represents that the performance of the proposed model is high. The error rate of the proposed and traditional models ate shown in Figure 8.

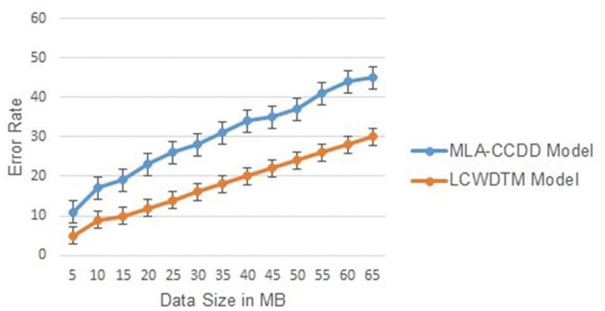

Fig. 8. Error Rate

5. CONCLUSION

In industrial and commercial applications, stream mining is a tough problem, but it has valuable potential yields. Data streams are an untapped supply of descriptive and analytical information that might be used in a variety of ways to improve the profitability and efficiency of enterprises. Machine learning approaches are difficult to implement because of the unrestricted size, uncertain pace, and variable features of data streams. Concept drift complicates the difficulty of developing online classifiers that can interpret streaming input. For data classification techniques, a Labelled Classifier with Weighted Drift Trigger Model is proposed that provides categorization and the capacity to tackle concept drift difficulties. Streams of evolving data with idea drift have a distribution that changes over time and at variable rates of severity. If a stream is observed, its core is always shifting, which can lead to a phenomena known as notion drift. Conventional machine learning models based on historical data may no longer be valid when employing streaming data to solve predictive problems. Stream-drifting demonstrated to be ideal for adaptive models provided with methods to reflect changes in the data. The proposed model trigger will be activated if there is any change observed in the data. The proposed classifier achieves an accuracy of 98% in drift detection that represents that the proposed model is efficient than the traditional methods. Another critical deficiency is the unavailability of test datasets for evaluation. Many researchers use generators since there aren't any real-world benchmark datasets available to them. There are a lot of generators available, and each one relies on a different set of user-specified parameters. It's up to the individual to decide which generator and settings are best suited for a given task. Next-generation research should focus on creating gold-standard datasets that can be used for experimentation.

6. REFERENCES

[1]. I. Khamassi, M. Sayed-Mouchaweh, M. Hammami, K. Ghédira, "A new combination of diversity techniques in ensemble classifiers for handling complex concept drift", Learning from Data Streams in Evolving Environments, Vol. 41, 2018, pp. 39–61.

[2]. A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, F. Herrera, "Learning from imbalanced data sets", Springer, Berlin, 2018.

[3]. A. Cano, B. Krawczyk, "Kappa updated ensemble for drifting data stream mining", Machine Learning, Vol. 109, 2020, pp. 175–218.

[4]. Z. H. Ghaderi, H. Altınçay, "Imbalance learning using heterogeneous ensembles", Expert Systems with Applications, Vol. 142, 2020, p. 113005.

[5]. M. M. Idrees, L. L. Minku, F. Stahl, A. A. Badii, "Heterogeneous online learning ensemble for non-stationary environments", Knowledge-Based Systems, Vol. 188, No. 4, 2020, p. 104983.

[6]. J. L. Fernandez-Aleman, J. M. Carrillo-De-Gea, M. Hosni, A. Idri, G. Garcia-Mateos, "Homogeneous and heterogeneous ensemble classification methods in diabetes disease: a review", Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2019.

[7]. H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, T. Abdessalem, "Adaptive random forests for evolving data stream classification", Machine Learning, Vol. 106, No. 9-10, 2017, pp. 1469–1495.

[8]. J. P. Barddal, H. M. Gomes, F. Enembreck, B. Pfahringer, "A survey on feature drift adaptation: definition, benchmark, challenges and future directions", Journal of Systems and Software, Vol. 127, No. 1, 2017, pp. 278–294.

[9]. C. C. Lin, D. J. Deng, C. H. Kuo, L. Chen, "Concept drift detection and adaption in big imbalance industrial IoT data using an ensemble learning method of offline classifiers", IEEE Access, Vol. 7, 2019, pp. 56198–56207.

[10]. A. V. Luong, T. T. Nguyen, A. V. Liew, "Streaming active deep forest for evolving data stream classification", arXiv:2002.11816v1, 2020.

[11]. B. Junior, M. D. C. Nicoletti, "An iterative boosting-based ensemble for streaming data classification", Information Fusion, Vol. 45, 2019, pp. 66–78.

[12]. J. Montiel, R. Mitchell, E. Frank, B. Pfahringer, T. Abdessalem, A. Bifet, "Adaptive XGBoost for evolving data streams", arXiv:2005.07353v1, 2020.

[13]. D. Brzezinski, J. Stefanowski, "Ensemble diversity in evolving data streams", Proceedings of the International Conference on Discovery Science, Bari, Italy, October 19–21, 2016, pp. 229–244.

[14]. J. Large, J. Lines, A. J. Bagnall, "The heterogeneous ensembles of standard classification algorithms (HESCA): the whole is greater than the sum of its parts", Data Mining and Knowledge Discovery, Vol. 33, 2019, pp. 1674–1709.

[15]. A. Pesaranghader, H. Viktor, E. Paquet, "Reservoir of diverse adaptive learners and stacking fast

hoeffding drift detection methods for evolving data streams", Machine Learning, Vol. 107, 2018, pp. 1711–1743.

[16]. C. T. Li, M. K. Shan, S. H. Jheng, K. C. Chou, "Exploiting concept drift to predict popularity of social multimedia in microblogs", Information Sciences, Vol. 339, 2016, pp. 310–33.

[17]. S. Liu, L. Feng, J. Wu, G. Hou, G. Han, "Concept drift detection for data stream learning based on angle optimized global embedding and principal component analysis in sensor networks", Computers and Electrical Engineering, Vol. 58, No. 8, 2017, pp. 327–336.

[18]. Y. Y. Lo, W. Liao, C. S. Chang, Y. C. Lee, "Temporal matrix factorization for tracking concept drift in individual user preferences", IEEE Transactions on Computational Social Systems, Vol. 5, No. 1, 2018, pp. 156–168.

[19]. D. Marrón, E. Ayguadé, J. R. Herrero, A. Bifet, "Resource-aware elastic swap random forest for evolving data streams", arXiv:1905.05881v1, 2019.

[20]. J. Montiel, J. Read, A. Bifet, T. Abdessalem, "Scikit-multiflow: a multi-output streaming framework", Journal of Machine Learning Research, Vol. 19, No. 72, 2018, pp. 1–5.

[21]. M. K. Olorunnimbe, H. L. Viktor, E. Paquet, "Dynamic adaptation of online ensembles for drifting data streams", Journal of intelligent information Systems, Vol. 50, No. 2, 2018, 291–313.

[22]. S. Ren, B. Liao, W. Zhu, Z. Li, W. Liu, K. Li, "The gradual resampling ensemble for mining imbalanced data streams with concept drift", Neurocomputing, Vol.286, 2018, pp. 150–166.

[23]. D. Ruano-Ordás, F. Fdez-Riverola, J. R. Méndez, "Concept drift in e-mail datasets: an empirical study with practical implications", Information Sciences, Vol. 428, 2018, pp. 120–135.

[24]. O. Sagi, L. Rokach, "Ensemble learning: a survey", Data Mining and Knowledge Discovery, Vol. 8, No. 5, 2018, p. e1249.

[25]. J. N. Van Rijn, G. Holmes, B. Pfahringer, J. Vanschoren, "The online performance estimation framework: heterogeneous ensemble learning for data streams", Machine Learning, Vol. 107, 2018, pp. 149–176.

[26]. S. Wang, L. L. Minku, X. Yao, "A systematic study of online class imbalance learning with concept drift", IEEE Transactions on Neural Networks and Learning Systems, 2018.

[27]. J. Zenisek, F. Holzinger, M. Affenzeller, "Machine learning based concept drift detection for predictive maintenance", Computers and Industrial Engineering, Vol. 137, 2019, p. 106031.

[28]. B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Wozniak, "Ensemble learning for data stream analysis: A survey," Information Fusion, Vol. 37, 2017, pp. 132–156.