Text stream classification: Literature review and current trends

Najet Medssia
RISC Laboratory
National Engineering School of Tunis
University of Tunis El Manar
Tunis,Tunisia
namedssia@iau.edu.sa

Minyar Sassi Hidri
Computer Department
Deanship of Preparatory Year and Supporting Studies
Imam Abdulrahman Bin Faisal University
Dammam, Saudi Arabia
mmsassi@iau.edu.sa

Ali Frihida
RISC Laboratory
National Engineering School of Tunis
University of Tunis El Manar
Tunis,Tunisia
ali.frihida@enit.utm.tn

***Abstract*—Text stream classification is the task of assigning labels to a sequence of text data as it arrives in real time. This is a challenging task because the data is often incomplete and noisy, and the labels may change over time. In this paper, we review the state-of-the-art in text stream classification models.We discuss the different types of models that have been proposed, as well as the strengths and weaknesses of each approach. We also identify some of the open challenges in text stream classification, and suggest directions for future research.**

***Index Terms*—Natural language processing, Text stream classification, Incremental approach, Deep learning , Active learning.**

I. Introduction

With the technology revolution and the widely used of the internet, the amount of data became increasingly exhibited and has led to the popularity of data streams. Learning from this data for many real applications presents unprecedented challenges. Classify this data stream is a challenging task due to the fact that the data is constantly changing and can contain a variety of different types of data.

Text stream classification is one of the hottest topics in machine learning (ML) since it involves automatically classifying text. It's a process of assigning a category or label to a piece of text. Text stream classification models can be trained using a variety of machine learning algorithms, such as support vector machines, decision trees, and random forests. However, these algorithms are often not able to keep up with the speed of the data stream.

Text stream classification has a wide range of applications, including:

- Sentiment analysis: Identifying the sentiment of text, such as positive, negative, or neutral.
- Topic detection: Identifying the main topics of a news stream.
- Fraud detection: Identifying fraudulent transactions and other types of fraudulent activity cute.
- Spam filtering: Identifying and filtering out spam emails.
- Customer service: Categorizing customer inquiries and routing them to the appropriate department.

In recent years, there has been a growing interest in developing deep learning models for text stream classification. Deep learning models have been shown to be very effective at learning complex patterns from data and can be more robust to concept drift than traditional machine learning algorithms.

In [1], they used the Convolutional Neural Networks (CNNs) for sentiment analysis tasks with great success. Recurrent Neural Networks (RNNs) have also been used for text classification [2].

Additionally, attention-based models such as Transformer networks have been used for document categorization. Moreover a recent paper by proposed an ensemble-based incremental learning method that combines CNNs and a Support Vector Machines (SVM).

Recently, an incremental transfer learning method for image classification tasks has been proposed. The authors found that their method achieved better performance than traditional incremental learning methods on several benchmark data-sets [3]. In the same context, proposed an end-to-end DL model for incremental classification tasks based on recurrent neural networks (RNNs) [4].

In streaming and evolving scenarios, constructing a training model can be challenging. As Data stream is exposed to the issue of concept drift, many other problems ensue in data stream classification, which is specific to the problem at hand. These issues are reliant on the type of application in which streaming classification is used.

This literature review will examine the state-of-the-art in text stream classification models. This paper gives a review of incremental approaches in section II. Text stream-based approaches are presented in section III. In section IV, we will present active learning-based approaches. Finally, we offer in Section V conclusions and directions for future work.

II. Incremental-based approaches

Incremental-based approach is done by updating the model incrementally as new data is received, rather than retraining the model from scratch each time. Which is well-suited for applications where the text data is constantly changing, such as social media monitoring or customer service chat-bots.

There are a number of different incremental-based approaches [5]. One common approach is to use a sliding

2769-5654/23/$31.00 ©2023 IEEE
DOI 10.1109/CSCI62032.2023.00044

window model. In a sliding window model, the model is trained on a subset of the most recent data. As new data is received, the oldest data in the window is discarded and the model is retrained on the new window of data. This approach is simple and effective, but it can be computationally expensive for large data sets. Another approach is to use an online learning algorithm. Online learning algorithms are able to update the model incrementally as new data is received, without having to retrain the model from scratch.

Incremental training involves feeding new data to the model and updating its parameters accordingly. This can be done online or offline. Online incremental training updates the model parameters after each new data point is processed, while offline incremental training batches up new data and updates the model parameters periodically.Incremental classification methods based on DL use a variety of techniques to update the model. These methods can be broadly divided into three categories: (1) Approaches with fine-tuning and (2) Approaches with knowledge distillation.

A. Fine-tuning based approaches

Incremental learning with fine-tuning approaches [6] based on DL is a recent approach to ML that has been gaining traction in the field. This approach combines the advantages of both incremental learning and DL, allowing for more efficient and accurate training of models. It involves fine-tuning a pre-trained DL model on new data while preserving the weights of the previously trained model. Examples include incremental learning for image classification [7], incremental learning for object detection [5], and incremental learning for natural language processing (NLP) [8]–[10]. For example, in a recent study [11], incremental learning with fine-tuning based on DL approach was used to improve the accuracy of medical diagnosis tasks. The results showed that this approach was able to achieve better performance than traditional methods.

Recent work has demonstrated that incremental learning with fine-tuning can significantly improve the accuracy of DL models, even when the data is limited or noisy. In particular, incremental learning with fine-tuning has been used to improve the accuracy classification tasks [12]. For example, researchers have used this approach to improve the accuracy of image classification tasks on ImageNet by up to 8%.

Similarly, researchers have used incremental learning with fine-tuning to improve the accuracy of NLP tasks such as sentiment analysis and question answering. In these cases, incremental learning with fine-tuning was able to improve the accuracy by up to 10%. However, the results demonstrate that this approach still suffers from high divergence.

Overall, incremental learning with fine-tuning based on DL is a powerful technique for improving the accuracy of DL models.

B. Knowledge distillation-based approaches

Incremental classification with Knowledge Distillation (KD) [13] methods is an emerging research topic in the field of ML. KD is a technique that allows a deep learning model to learn from new data without forgetting previously learned information. KD technique can be used to transfer knowledge from a pre-trained model to a new model, allowing the new model to perform better than if it had been trained from scratch. In recent years, KD has been applied to various tasks such as image classification [14], [15], object detection [16], and NLP [17].

There are different approaches that have been proposed for incremental learning with KD. The first approach [18] for incremental learning with KD is based on a distillation of the pre-trained model's weights into a smaller network. This approach has been used successfully in image classification tasks such as CIFAR-10 and ImageNet. The main advantage of this approach is that it allows for fast training times due to the smaller size of the network. However, it can suffer from overfitting if not properly regularized or if too much information is distilled into the smaller network.

In [16], another approach has been used successfully in object detection tasks such as PASCAL VOC and MS COCO datasets. This approach allows for more accurate predictions due to the larger size of the network which can capture more complex features than a smaller network would be able to do. However, this approach can suffer from slow training times due to its larger size and complexity.

There are also approaches that combine both distillation techniques into one framework, such as Knowledge Distillation Networks (KDNets). KDNets have been used successfully in image classification tasks such as CIFAR-10 and ImageNet datasets. The main advantage of KDNets is that they allow for both fast training times due to their small size and more accurate predictions due to their larger size which can capture more complex features than a smaller network would be able to do.

Overall, incremental learning with Knowledge Distillation methods has shown promising results in various tasks such as image classification, object detection, and text classification [19].

Further research should focus on improving existing approaches by exploring different architectures or combining different techniques into one unified framework in order to achieve better performance on these tasks while still maintaining fast training times and low memory usage requirements.

III. TEXT STREAM-BASED APPROACHES

Text stream classification is a process of automatically assigning labels to text streams, such as tweets, news articles, blog posts, and other online content. This process is used to identify the topics of the text streams and to classify them into different categories. Text stream classification can be used for a variety of purposes, such as identifying trends in public opinion or sentiment analysis. It can also be used to detect spam or malicious content. In addition, text stream classification can be used to improve search engine results by providing more accurate and relevant results.

Data streams are different from static data in that they are often inexhaustible, speed flow, and are subject to concept drift, making the mining of data streams extremely difficult and having many challenges such as Data Scarcity , Noise and Concept drift.

Due to these numerous challenges, text stream classification become a rich field in the literature. The achievement in this field can be grouped into the following three categories according to the user model [20]:

- **Sliding window (or Chunk-based) approaches**: They work by dividing the data stream into a series of windows. Each window contains a fixed number of data points. The model is trained on the data in the current window and then used to classify the data in the next window.
- **Ensemble approaches**: They involve combining multiple models to create a more accurate and robust model.
- **Online learning approaches**: They allow the model to be updated as new data is received. This makes it possible for the model to adapt to changing data streams.

Figure 1 shows the stream classification approaches and highlights the adopted architectures and the implemented algorithms [21]–[23] .

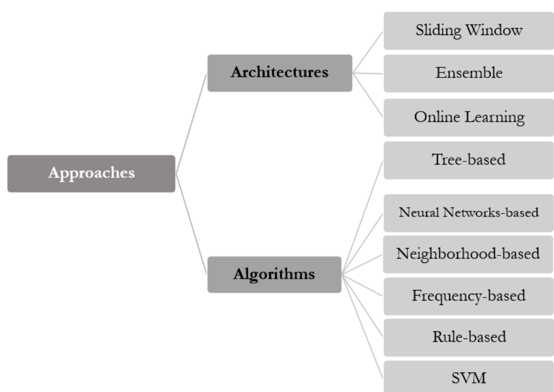

Fig. 1. Stream classification approaches.

Chunk-based training is faster than online training at first because it gives the model more data to start with. This helps the model learn more quickly. However a concept drift may occur later in the learning process, and it is possible that it will be missed. With online learning a latter problem does not affect the performance of the model; however, the model may suffer from slower initial learning performance. The latter is slower and less efficient than chunk-based learning.

In [24], they proposed a learning ensemble based on concept drifting data streams. They used both online and block-based methods. The mean square error is used to determine the weight of the classifier and the replacement strategy to adjust the problem of concept drift when occurs.

Many studies have been conducted on Decision Tree (DT) models for data stream classification.

The Density functional theory verification (DFTV) [25] and the variant concept-adapting very fast decision tree (CVFDT) [25] approaches are based on the DT. They used a sliding window to handle the concept drift proposed in [26].

In [27], the authors propose a model to classify imbalanced data streams based on Neural Networks (NNs). There are other works that use ensembles of classifiers, such as Extreme Learning Machine (ELM) [28]. In data stream classification, ensembles of classifiers facilitate adaptation to changing distributions of data, thus making them suitable for building data stream classifiers.

Concept drift detection is used in all of the approaches above and in most of the literature [29]. Using this process, we can identify when the underlying concept of data has changed, which can be caused by changes in the environment or by the addition of new data over time. It helps to ensure that models are up-to-date and accurate, or simply replace outdated models with new ones.

TABLE I
DRIFT DETECTION METHOD.

Concept Drift Type	Method
Gradual	Circle [30], R-RBF with drift [31], RHG [25], Rotating checkerboard [32], CUSUM [33], ADWIN [32].
Abrupt	Gauss [30], Mixed [30], RTG [25], SEA [34], Sine [30], STAGGER [30],WaveForm [35].
Incremental	RHG [25], RHG with drift [31], MetaL [36], AUE [36].

Many existing methods aim to detect this change [37]. In [38], the authors treated the drift detection concept with STAGGER method.

Table I represents a synthetic of the most used concept drift method detection and the type of change that detects.

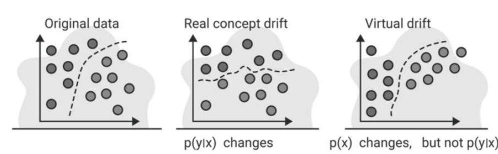

Fig. 2. Drift classification method according to subject.

The concept drift can be grouped into two categories according to the subject [39]:

- Real subject: it concerns changes for all examples but it could be also a sub-concept change which refers to changes in the conditional distribution of the output.
- Virtual: virtual concept drift when data stream contains noise and consider it as concept drift.

Figure 2 shows the Drift classification method according to subject.

Another categorization of concept drift according to the type [39] or their evolution over time (see Fig. 3): Abrupt(sudden) when the change occurs suddenly in the distribution of the data.

Incremental concept drift can be difficult to detect because the changes are often small and become slightly more and more inaccurate over time, Gradual concept drift can be difficult to detect because it occurs over a long period of time and can be hard to distinguish from normal fluctuations in the data and Recurring concept drift occur either cyclically or randomly, depending on their context.

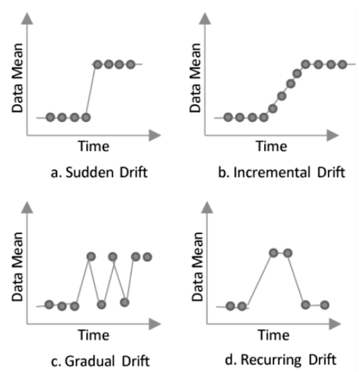

Fig. 3. Concept Drift according Type.

Stream classification is a form of adaptive learning,in terms of performance measurements, monitoring, displaying, and interpreting them over time is essential [40]–[42]. The drift detection concept is not the only metric to evaluate the model performance, there are others parameters that must be checked to evaluate the model performance such as test procedure [43], [44], decision metrics [43]–[46], statistical tests [44], and ensemble evaluation [44], [47] (see Fig. 4).

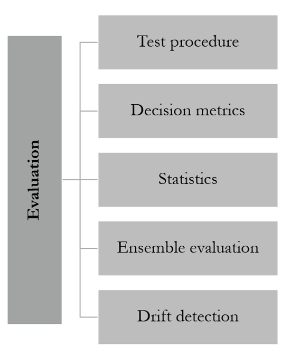

Fig. 4. Stream classification model: Evaluation.

In [48], they list and compare data sets used in previous studies based on their frequency of use.

IV. Active learning-based approaches

No matter what classification algorithm is used, training text must be correctly labeled. In many applications, manually labeling training examples is prohibitively expensive. Time, resources, and specialized annotators are needed to perform labeling tasks. In this case, the active learning (AL) approach is the suitable solution.

AL is a type of machine learning technique that focuses on decreasing the mandatory amount of annotation to train an accuracy model [49].

The process of AL begins with a model trained with a small amount of labeled training data. After the primary iterations, the model starts training from annotated data. The strategy to train the model in each iteration selects the most relevant data points from a large pool of unlabeled data. This allows the model to learn more efficiently and accurately, as it only presents the most relevant and informative data points. How to choose this relevant sample in each iteration?

Figure 5 shows the overview of the AL process.

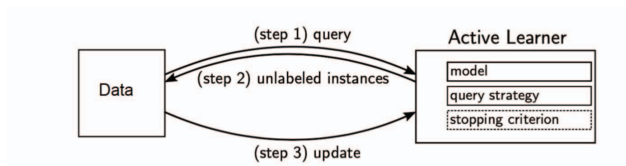

Fig. 5. Active learning process.

Within AL, query strategies are used to select the most informative data points from a large pool of unlabeled data. This allows the model to learn more efficiently and quickly, as it only needs to focus on the most relevant data points. Query strategies can be based on uncertainty, diversity, or other criteria.

A. Sampling selection random-based strategy

As with passive learning (standard learning), this strategy adopts samples for training purposes from the large pool of available unlabeled data.

In [50] and [51], they used random sampling to evaluate the effectiveness of their approaches in terms of accuracy.

B. Sampling selection uncertainty-based Strategy

Based on the AL literature, uncertain select samples are the most popular used. The uncertainty sampling strategy applies a probabilistic scoring function to sort the instances. A high level of uncertainty indicates that the instance's label was given in error. A general uncertainty sampling variant uses Least Confidence (LC) and Entropy [51].

The lasts are used in single and ensemble models and evaluated against random sampling (chance) as a baseline.

Many approaches are built based on this strategy for example, in [52], they proposed an approach called DBAL (Deep Bayesian Active Learning) to calculate the uncertainty according to the probabilistic distribution using the Monte Carlo Dropout method, which uses the neural network regularization technique known as a dropout. In [53] and [50], they computed the gradients of losses to get the sample uncertainty.

C. Sampling selection diversity-based strategy

Within this strategy, the learner requests a small diverse set of unlabeled instances according to the local interpretations [54].

In [51], the authors proposed an approach, called ALDEN (DivErse iNterpretations), based on DNN (Deep Neural Network) inspired by the local piece-wise interpretability of DNNs. They introduced the linearly separable regions of samples to the problem of deep active learning.

In the same context, ReLU (Rectified Linear Unit) [55], Maxout [56], and CORESET (an Active Learning for Convolutional Neural Networks) [54] proposed approaches showed that a deep model with piece-wise linear activation can be regarded as a set of numerous local linear classifiers.

D. Sampling selection ensemble-based strategy

In case when the approach combines two or more query strategies it's called *Ensemble Strategy*.

In [57], they grouped EGL (Expected Gradient Length) and CORESET in their approach called BADGE (Batch Active learning by Diverse Gradient Embeddings) to trade-off between uncertainty and diversity without the need for any hand-tuned hyperparameters. Table II gives a summary of the different techniques used in the sampling selected strategy.

TABLE II
OVERVIEW OF ACTIVE LEARNING QUERY STRATEGY.

Query strategy	Technique	Approach
Uncertainty	Entropy [58] [59] [52] [60]	NB, SVM, BALD, FTZ, ULMFiT
	Monte Carlo Dropout [58] [61]	BERT, CREGEX
	Expected Gradient Length [50]	EGL-Word
	Least Confidence [62]	LSTM-CRF
Diversity	Last layer [54]	CORESET
	Gradient Backpropagation [51]	ALDEN
	Piece-wise linear activation functions [56] [55]	Maxout, ReLU
Ensemble	Uncertainty + Diversity [57] [63]	BADGE, QUIRE.

V. CONCLUSION

This paper highlights a wink on the state of the art of text stream classification. We start focusing on the deep neural approaches which are widely used in this field. Moreover, this work gives a brief summary of the approaches used in text stream classification such as the DT, static, and deep models. We state the most popular techniques on the concept of drift detection.

As a solution to the unlabeled data, this literature review focused on Active learning which can resolve this problem. We closed this paper with an overview of techniques used for the sampling query strategy.

In future directions, we aim to integrate and exploit new training approaches based on the exploitation of old incremental learning algorithms in order to optimize the DL architecture.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[2] M.-S. Kim, B.-Y. Lim, K. Lee, and H.-Y. Kwon, "Effective model update for adaptive classification of text streams in a distributed learning environment," *Sensors*, vol. 22, no. 23, p. 9298, 2022.

[3] J. Wang, Z. Mo, H. Zhang, and Q. Miao, "Ensemble diagnosis method based on transfer learning and incremental learning towards mechanical big data," *Measurement*, vol. 155, p. 107517, 2020.

[4] M. Muzammel, H. Salam, and A. Othmani, "End-to-end multimodal clinical depression recognition using deep neural networks: A comparative analysis," *Computer Methods and Programs in Biomedicine*, vol. 211, p. 106433, 2021.

[5] J. Leo and J. K. Kalita, "Incremental deep neural network learning using classification confidence thresholding," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, pp. 7706–7716, 2021.

[6] P. Peng and J. Wang, "How to fine-tune deep neural networks in few-shot learning?" *arXiv preprint arXiv:2012.00204*, 2020.

[7] W. Wang, D. Liang, Q. Chen, Y. Iwamoto, X.-H. Han, Q. Zhang, H. Hu, L. Lin, and Y.-W. Chen, "Medical image classification using deep learning," *Deep learning in healthcare: paradigms and applications*, pp. 33–51, 2020.

[8] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[9] H.-J. Dai, C.-H. Su, Y.-Q. Lee, Y.-C. Zhang, C.-K. Wang, C.-J. Kuo, and C.-S. Wu, "Deep learning-based natural language processing for screening psychiatric patients," *Frontiers in psychiatry*, vol. 11, p. 533949, 2021.

[10] S. A. Alsaif, M. Sassi Hidri, I. Ferjani, H. Eleraky, and A. Hidri, "NLP-based bi-directional recommendation system: Towards recommending jobs to job seekers and resumes to recruiters," *Big Data Cogn. Comput.*, vol. 6, no. 4, p. 147, 2022.

[11] F. Ramzan, M. U. G. Khan, A. Rehmat, S. Iqbal, T. Saba, A. Rehman, and Z. Mehmood, "A deep learning approach for automated diagnosis and multi-class classification of alzheimer's disease stages using resting-state fmri and residual neural networks," *Journal of medical systems*, vol. 44, pp. 1–16, 2020.

[12] T. Gao, A. Fisch, and D. Chen, "Making pre-trained language models better few-shot learners," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Association for Computational Linguistics, 2021, pp. 3816–3830.

[13] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, vol. abs/1503.02531, 2015.

[14] M. Kang, J. Park, and B. Han, "Class-incremental learning by knowledge distillation with adaptive feature consolidation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 16 071–16 080.

[15] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 233–248.

[16] L. Chen, C. Yu, and L. Chen, "A new knowledge distillation for incremental object detection," *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2019.

[17] A. Alkhulaifi, F. Alsahli, and I. Ahmad, "Knowledge distillation in deep learning and its applications," *PeerJ Computer Science*, vol. 7, p. e474, 2021.

[18] U. Asif, J. Tang, and S. Harrer, "Ensemble knowledge distillation for learning improved and efficient networks," *arXiv preprint arXiv:1909.08097*, 2019.

[19] J. Chen, Z. Feng, and W. Jiang, "Analysis and comparison of chinese news text classification methods based on deep learning," *Highlights in Science, Engineering and Technology*, vol. 16, pp. 146–154, 2022.

[20] S. Bakhshi, P. Ghahramanian, H. Bonab, and F. Can, “A broad ensemble learning system for drifting stream classification,” *IEEE Access*, 2023.

[21] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy, “A survey of classification methods in data streams,” *Data Streams: models and algorithms*, pp. 39–59, 2007.

[22] A. Hidri, A. Selmi, and M. Sassi Hidri, “Discovery of frequent patterns of episodes within a time window for alarm management systems,” *IEEE Access*, vol. 8, pp. 11 061–11 073, 2020.

[23] M. Ammar, A. Hidri, , and M. Sassi Hidri, “Time-sensitive clustering evolving textual data streams,” *Int. J. Comput. Appl. Technol.*, vol. 63, no. 1/2, pp. 25–40, 2020. [Online]. Available: https://doi.org/10.1504/IJCAT.2020.107900

[24] D. Brzezinski and J. Stefanowski, “Combining block-based and online methods in learning ensembles from concept drifting data streams,” *Information Sciences*, vol. 265, pp. 50–67, 2014.

[25] G. Hulten, L. Spencer, and P. Domingos, “Mining time-changing data streams,” in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 97–106.

[26] ——, “Mining time-changing data streams,” in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’01. New York, NY, USA: Association for Computing Machinery, 2001, p. 97–106.

[27] A. Ghazikhani, R. Monsefi, and H. Sadoghi Yazdi, “Online neural network model for non-stationary and imbalanced data stream classification,” *International Journal of Machine Learning and Cybernetics*, vol. 5, pp. 51–62, 2014.

[28] R. Yang, S. Xu, and L. Feng, “An ensemble extreme learning machine for data stream classification,” *Algorithms*, vol. 11, no. 7, p. 107, 2018.

[29] A. Tsymbal, “The problem of concept drift: definitions and related work,” *Computer Science Department, Trinity College Dublin*, vol. 106, no. 2, p. 58, 2004.

[30] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, “Learning with drift detection,” in *Advances in Artificial Intelligence–SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings 17*. Springer, 2004, pp. 286–295.

[31] A. Bifet, R. Gavalda, G. Holmes, and B. Pfahringer, *Machine learning for data streams: with practical examples in MOA*. MIT press, 2018.

[32] R. Elwell and R. Polikar, “Incremental learning of concept drift in nonstationary environments,” *IEEE Transactions on Neural Networks*, vol. 22, no. 10, pp. 1517–1531, 2011.

[33] S. Muthukrishnan, E. Van Den Berg, and Y. Wu, “Sequential change detection on data streams,” in *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*. IEEE, 2007, pp. 551–550.

[34] W. N. Street and Y. Kim, “A streaming ensemble algorithm (sea) for large-scale classification,” in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 377–382.

[35] T. Wang, D. Trugman, and Y. Lin, “Seismogen: Seismic waveform synthesis using gan with application to seismic data augmentation,” *Journal of Geophysical Research: Solid Earth*, vol. 126, no. 4, p. e2020JB020077, 2021.

[36] D. Brzezinski and J. Stefanowski, “Reacting to different types of concept drift: The accuracy updated ensemble algorithm,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 81–94, 2013.

[37] S. V. Kadam, “A survey on classification of concept drift with stream data,” 2019.

[38] L. Clever, J. S. Pohl, J. Bossek, P. Kerschke, and H. Trautmann, “Process-oriented stream classification pipeline: A literature review,” *Applied Sciences*, vol. 12, no. 18, p. 9094, 2022.

[39] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” *ACM computing surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014.

[40] I. Ferjani, M. Sassi Hidri, and A. Frihida, “SiNoptiC: swarm intelligence optimisation of convolutional neural network architectures for text classification,” *Int. J. Comput. Appl. Technol.*, vol. 68, no. 1, pp. 82–100, 2022.

[41] S. A. Alsaif, M. Sassi Hidri, H. A. Eleraky, I. Ferjani, and R. Amami, “Learning-based matched representation system for job recommendation,” *Comput.*, vol. 11, no. 11, p. 161, 2022.

[42] M. Sassi Hidri, S. A. Alsaif, and A. Hidri, “Performance analysis of machine learning algorithms on networks intrusion detection,” *Int. J. Comput. Appl. Technol.*, vol. 70, no. 3/4, pp. 285–295, 2022.

[43] S. U. Din, J. Shao, J. Kumar, C. B. Mawuli, S. H. Mahmud, W. Zhang, and Q. Yang, “Data stream classification with novel class detection: a review, comparison and challenges,” *Knowledge and Information Systems*, vol. 63, pp. 2231–2276, 2021.

[44] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, “Ensemble learning for data stream analysis: A survey,” *Information Fusion*, vol. 37, pp. 132–156, 2017.

[45] I. Ferjani, M. Sassi Hidri, and A. Frihida, “Multi-GPU-based convolutional neural networks training for text classification,” in *Intelligent Systems and Applications - Proceedings of the 2021 Intelligent Systems Conference, IntelliSys 2021, Amsterdam, The Netherlands, 2-3 September, 2021, Volume 1*, ser. Lecture Notes in Networks and Systems, K. Arai, Ed., vol. 294. Springer, 2021, pp. 72–84.

[46] S. A. Alsaif, A. Hidri, and M. Sassi Hidri, “Towards inferring influential facebook users,” *Computers*, vol. 10, no. 5, p. 62, 2021.

[47] P. Sidhu and M. Bhatia, “A novel online ensemble approach to handle concept drifting data streams: diversified dynamic weighted majority,” *International Journal of Machine Learning and Cybernetics*, vol. 9, pp. 37–61, 2018.

[48] B. Srivani, N. Sandhya, and B. Padmaja Rani, “Literature review and analysis on big data stream classification techniques,” *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 24, no. 3, pp. 205–215, 2020.

[49] C. Schröder and A. Niekler, “A survey of active learning for text classification using deep neural networks,” *arXiv preprint arXiv:2008.07267*, 2020.

[50] Y. Zhang, M. Lease, and B. Wallace, “Active discriminative text representation learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[51] Q. Liu, Y. Zhu, Z. Liu, Y. Zhang, and S. Wu, “Deep active learning for text classification with diverse interpretations,” in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 3263–3267.

[52] Y. Gal, R. Islam, and Z. Ghahramani, “Deep bayesian active learning with image data,” in *International conference on machine learning*. PMLR, 2017, pp. 1183–1192.

[53] J. Huang, R. Child, V. Rao, H. Liu, S. Satheesh, and A. Coates, “Active learning for speech recognition: the power of gradients,” *arXiv preprint arXiv:1612.03226*, 2016.

[54] O. Sener and S. Savarese, “Active learning for convolutional neural networks: A core-set approach,” *arXiv preprint arXiv:1708.00489*, 2017.

[55] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.

[56] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, “Maxout networks,” in *International conference on machine learning*. PMLR, 2013, pp. 1319–1327.

[57] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal, “Deep batch active learning by diverse, uncertain gradient lower bounds,” *arXiv preprint arXiv:1906.03671*, 2019.

[58] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[59] S. Tong and D. Koller, “Support vector machine active learning with applications to text classification,” *Journal of machine learning research*, vol. 2, no. Nov, pp. 45–66, 2001.

[60] A. Prabhu, C. Dognin, and M. Singh, “Sampling bias in deep active classification: An empirical study,” *arXiv preprint arXiv:1909.09389*, 2019.

[61] C. A. Flores, R. L. Figueroa, and J. E. Pezoa, “Active learning for biomedical text classification based on automatically generated regular expressions,” *IEEE Access*, vol. 9, pp. 38 767–38 777, 2021.

[62] A. Shelmanov, D. Puzyrev, L. Kupriyanova, D. Belyakov, D. Larionov, N. Khromov, O. Kozlova, E. Artemova, D. V. Dylov, and A. Panchenko, “Active learning for sequence tagging with deep pre-trained models and bayesian uncertainty estimates,” *arXiv preprint arXiv:2101.08133*, 2021.

[63] S.-J. Huang, R. Jin, and Z.-H. Zhou, “Active learning by querying informative and representative examples,” *Advances in neural information processing systems*, vol. 23, 2010.