Concept Drift Detection using Transformer Autoencoder

Mario Luca Bernardi
Dept. of Engineering
University of Sannio
Benevento, Italy
bernardi@unisannio.it

Marta Cimitile
Dept. of Law and Digital Society
UnitelmaSapienza University
Rome, Italy
marta.cimitile@unitelmasapienza.it

Anna Vacca
Dept. of Law and Digital Society
UnitelmaSapienza University
Rome, Italy
anna.vacca@unitelmasapienza.it

***Abstract*—Applying machine learning to fully distributed environments is always becoming more crucial in several real-life contexts. In networked environments, data models can evolve dynamically over time subject to continuous changes known as concept drifts. Detecting when concept drift occurs is essential for various drift-handling techniques and plays a significant role in many scenarios. However, while drift-handling methods exist, an efficient solution for detecting drift in large-scale networks remains unknown. This study proposes a concept drift detection approach allowing to capture temporal variations in the data, enabling a more robust identification of data changes. Specifically, our method detects local models exhibiting a concept drifts by analyzing deviations in learned representations over time. We validate our approach using a real-world urban traffic dataset, demonstrating its effectiveness in identifying concept drift in real scenarios. The results show that the proposed approach successfully identifies sudden and gradual drifts, respectively achieving an F1-score of 0.94 in sudden drift detection and an F1-score of 0.92 in gradual drift detection.**

***Index Terms*—concept drift detection, transformer autoencoder, drift detection**

I. Introduction

Using Machine Learning (ML) approaches in distributed scenarios has become increasingly common in daily life, enabling automated decision-making and self-learning mechanisms [27] in a safer and more privacy-preserving environment. Given the dynamic and heterogeneous nature of data and data sources [10] in distributed environments, the performance of ML models suffers from concept drifts, that can invalidate the data model [8], [22] and cause their performance degradation. Concept drifts are defined as unforeseeable changes in data over time [15] so that the statistical properties of the variable, which an ML model aims to predict, and continually change in a not predictable way. To mitigate the impact of concept drift on the performance of predictive models, several studies have been proposed in the last years [6]. However, according to [6], most existing strategies are conceived in centralized scenarios and show reduced performance when applied and adapted to distributed settings [11]. To overrun this limitation, more recently, specific strategies have been defined for the concept drift detection in distributed settings [9], [11], [22]. For example, authors in [11] propose a method called FedNN employing Weight Normalization and Adaptive Group Normalization suited for concept drift heterogeneity. They show that this approach outperforms the same state-of-the-art Federated Learning (FL) methods.

This study introduces a Concept Drift Detection strategy for distributed environments. The novelty proposed is the adoption of a Transformer Autoencoder to i) identify concept drifts and ii) identify the type of concept drift (sudden drift, gradual drift).The proposed autoencoder is trained on a baseline describing different concept drifts.The proposed approach is evaluated on a real-world dataset representing an urban traffic scenario. The proposed assessment is independent from the underlying model, for example a Convolutional Recurrent Neural Network (DCRNN) [12] can be used.

The document is structured as follows: Section II reviews the most relevant studies on concept drift detection. Section III covers fundamental concepts related to concept drift. Section IV provides a detailed description of the proposed approach. Section V presents the empirical validation, while Section VI discusses the obtained results. Finally, Section VII concludes the work with a summary and discussion.

II. Related Work

Concept drift detection allows the identification of the specific time instant or interval when changes occur in the properties of a monitored object [2], [18]. The drift identification allows for an update of the prior knowledge of a system and drives the learning models to properly react to the upcoming changes [5]. For this reason, in the last years, several concept drift detection approaches have been proposed [2], [24]. However, several existing strategies aim to detect concept drift in centralized settings and offer limited performance when applied in distributed scenarios. Starting from these considerations, some centralized algorithms are

adapted to the distributed context showing improved performance. The study proposed in [5] introduces an adaptation of the well-known Federated Averaging (FedAvg) [17] algorithm for distributed learning under concept drift. The method has been tested on an ad-hoc dataset describing the activities of 10 different users when they use their mobile devices. The results show the improved performance of the approach with respect to the traditional FedAvg. Another study proposing an adaptation of the FedAvg algorithm is described in [10]. Even if these strategies show improved performance, they fail to incorporate the distributed nature of large-scale and distributed networks [10]. However, in these settings, traditional drift detection techniques are difficult to implement due to the absence of direct access to centralized data. Furthermore, the variation in data distribution across networked clients adds complexity to the application and effectiveness of these methods [19]. To overcome this limitation, new approaches are proposed in recent years. FedRepo [22] is a horizontal federated learning [13] concept drift detection methodology based on Random Forest (RF) regression models. The objective is to train and continuously adjust a repository of federated models that is linked to a group of similar clients collaboratively. The evaluation of the approach on a real dataset shows its capability to handle adequately concept drifts. Authors in [11] introduce FedNN combining Weight Normalization (WN) and Adaptive Group Normalization (AGN) to reduce heterogeneity in concept drift data. Other studies have considered the differences between online and offline model training to identify anomalies using concept drifts. In fact, since often the training of a model with time series takes place offline and may not be corrected in dynamic environments where normal behavior changes over time, the authors in [20] have proposed a model based on Recurrent Neural Networks (RNN), which is updated incrementally with new data coming in. The model uses RNN to make multi-step time series predictions and analyzes forecast errors to identify anomalies and points of change.

Differently from the above solutions, we used an approach based on transformer autoencoder suitable to identify concept drifts thanks to an improved capability to identify the set of complex patterns of signal variability over time.

III. Background: Concept Drift

The phenomenon of concept drift refers to the changes in the statistical properties of a target domain over time [16]. It can occur due to evolving patterns, external influences, or unforeseen shifts in data distribution, making it essential to detect and adapt models accordingly. Concept drift can be categorized into the following types [7], [14]:

- Sudden (Abrupt) Drift: The data distribution changes abruptly at a specific point in time.
- Gradual Drift: The data distribution changes slowly over time, with both old and new patterns coexisting for a period before the new pattern dominates.
- Incremental Drift: The change happens progressively in small steps rather than all at once, making it more challenging to detect.
- Recurring (Seasonal) Drift: The data distribution follows a cyclical pattern, repeating after a certain period.
- Virtual Drift: The input data distribution changes, but the relationship between input and output remains unchanged. This does not necessarily impact model performance directly but may indicate potential future shifts.

Our study primarily examines sudden and gradual concept drift, since the other types of concept drifts are often reducible to combinations of these two fundamental types.

IV. The approach

The proposed architecture aims to identify concept drifts according to their types and dynamics in a FL context. The architecture of the proposed Concept Drift Detection approach is described in Figure 1.

The figure shows the distributed learning scenario where there is a fixed number of local clients (Client 1, .., Client n). As described on the right side of the figure, each client generates local models according to the local data stream. The models are then transferred to the Global Model Manager (left side of the figure), where they are combined and aggregated. The Global Model Manager can be implemented using different models since the proposed approach is agnostic from the underlying model.

At the client level, the Local Model Manager monitors the training of the local models. Moreover, the Drift Detector component examines the models training data to detect concept drifts. The local detection is performed using a transformer-based autoencoder [25], [26] that processes the data by reconstructing input patterns and measuring deviations through reconstruction errors. These errors are then assessed against an adaptive threshold to identify drifted nodes. Considering the left side of the figure, there are all the components that are responsible for aggregating the local models into the global model. However, the Aggregation Manager ensures the filtering of all the models that exhibit adversarial drift. We assume that the sudden drifts can be filtered out while the other drifts can be included. According to the definition reported in Section III, we consider a drift as sudden concept drift if the local data stream deviates abruptly from the expected dynamics. However, the filtering rules can change according to the context parameters. After the filtering, the Aggregation Manager aggregates the local models using a custom FL strategy called DQFed described in [3]. The obtained

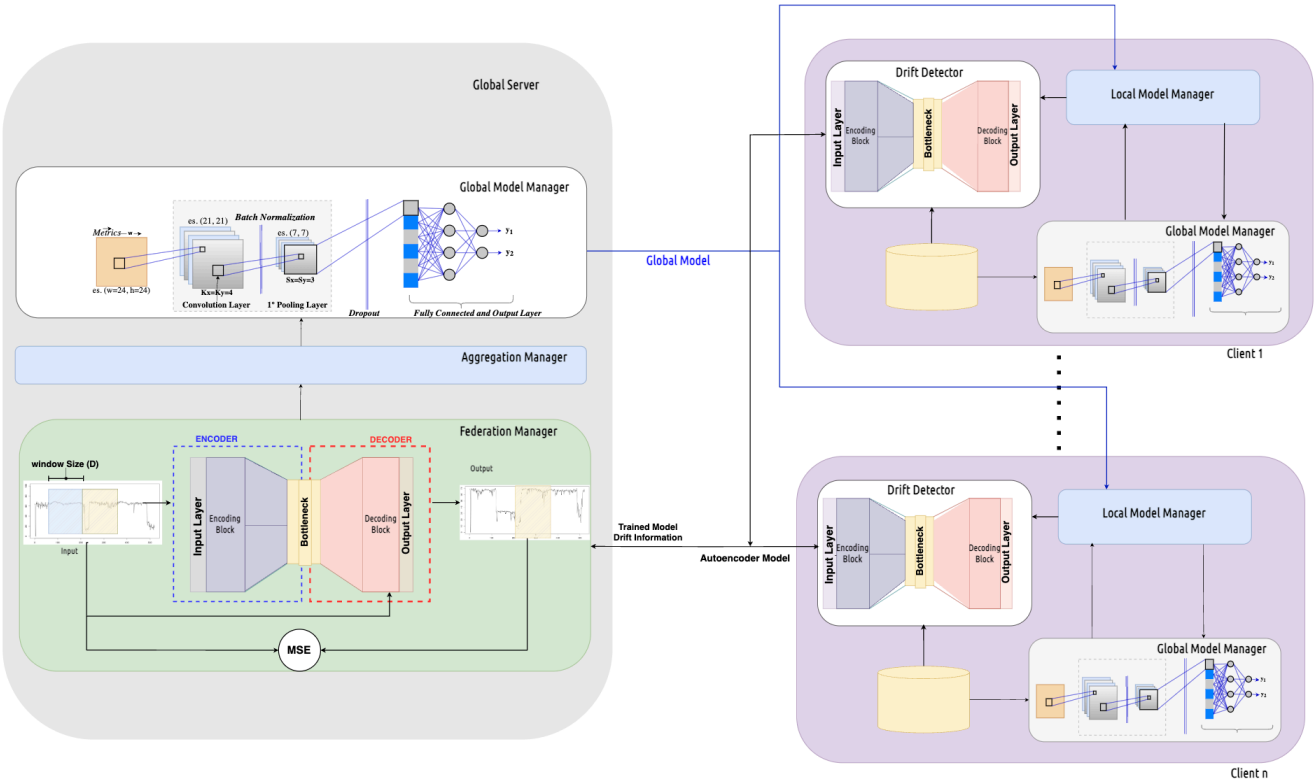


Fig. 1: The concept drift detection architecture.

global model is then sent to the Global Model Manager, which redistributes it to the clients. The autoencoder component allows to detection concept drifts from the analysis of the local data stream [1].

We used a Transformer-based network [25] in order to identify complex patterns over long-running sequences. By analyzing the relationships between running sequences, the Transformer can identify anomalous trends that deviate from expected patterns in the data. Additionally, its attention mechanisms enable it to focus on specific parts of the input sequence that are particularly relevant for detecting drifts.

The transformer autoencoder is composed of an encoder and a decoder component. The encoder transforms the input data, while the decoder generates the data distribution from the encoded representation.

Along the training process, the autoencoder learns an efficient representation of the initial dataset.

In this step, the input dataset is split into sliding windows (the window size is D) as described in the figure 1.

For each window, the MSE (Mean Squared Error) reconstruction error is minimized by optimizing the weights of the hidden states in both the encoding and decoding layers of the autoencoder.

The MSE is computed according to the following formula:

$$MSE = |X - (f_{dec} \circ f_{enc})X| \tag{1}$$

where, given a D-dimensional data stream of N elements $\{X_1, X_2, ..., X_N\}$, in which $X_i^j = \{X_i^1, X_i^2, ..., X_i^D\}$ for timestamp i, the encoder and decoding functions have the following dimensions:

$$f_{enc} : R^{S \times D} \rightarrow R^H \qquad f_{dec} : R^H \rightarrow R^{S \times D} \tag{2}$$

where H is the size encoded layer.

The MSE is evaluated using a fixed error threshold. According to the analysis of distribution of reconstructing errors, we consider different thresholds to distinguish the concept drift type. Hence, we assumed that there is a type of concept drift when the MSE stream overruns the corresponding threshold.

The encoder consists of a set of embedding layers, a positional encoding layer, and $N_E = 4$ stacked encoder layers. The embedding layer transforms the time series data into a D-dimensional vector using a fully connected network. Positional encoding plays a crucial role in capturing the dynamics of the time series data. The extracted hidden features are then passed to the encoder block, which consists of two layers: a self-attention mechanism and a positional fully connected network. Each layer uses a residual net-like structure and a normalization layer. The obtained features are then input into the reconstruction network.

The transformer's reconstruction network leverages the global features encoded by the encoder to reconstruct

the input. The Decoder(·) function follows the standard transformer decoder block, as defined in [23].

V. Experiment Description

The empirical validation aims to evaluate the effectiveness of the concept drift detection approach. More precisely, we investigate the effectiveness of the proposed approach in detecting gradual and sudden concept drifts. The set of proposed experiments is conducted by applying the proposed approach to a traffic dataset evaluating its performance in detecting a set of introduced gradual and concept drifts.

The following subsections propose a short description of the adopted dataset and the setting of the experiments.

A. The dataset

In this study, we used our approach on a dataset obtained starting from METR-La [21], a real-world and large-scale dataset[1].

METR-La dataset contains traffic information extracted by loop detectors posed on highways in Los Angeles County [4]. The data was collected from March 2012 to June 2012 using 207 sensors (they will correspond to the nodes of our network). This dataset is then modified by randomly introducing sudden and gradual drifts. Especially, starting from the initial dataset, 30 experiments are conducted each one introducing a different number of gradual and sudden drifts.

Figure 2 shows the number of drifted stations across the conducted experimental runs (from 1 to 30).

Each experiment injects concept drifts into varying numbers of nodes, affecting different parts of the network. The figure shows fluctuations in drifted nodes due to randomized injection, ensuring the detection approach is tested under diverse conditions.

Specifically, for a given station, a data window is individuated starting from an initial data point, and a transformation function is applied to introduce the drift.

In the case of sudden drift, the adopted transformation function is the following:

$$x_t = x_t \cdot \lambda, \quad \text{for } t_d \leq t < t_e \tag{3}$$

where λ represents the drift multiplier applied to the affected data. Instead, concerning gradual drift, the adopted transformation function is the following:

$$x_t = x_t \cdot (1 + \lambda(t - t_d)), \quad \text{for } t_d \leq t < t_e \tag{4}$$

where x_t represents the data value at time t; t_d and t_e are the drift starting and ending points; λ is the drift increment.

[1]https://graphmining.ai/temporal_datasets/METR-LA.zip

B. Experimental setting

The validation of the proposed approach in the traffic scenario is performed using the original METR-LA dataset to train the proposed transformer autoencoder. Specifically, starting from the initial dataset a training set (containing of 80% of the baseline dataset), and a test set (containing of the remaining 20% of the baseline dataset) were used.

Successively, at each run, the transformer autoencoder is used in the modified METR-LA dataset to perform the traffic detection.

According to the approach described in Section IV the MSE reconstruction error over time windows is computed to identify the concept drifts. Referring to x_i as the input data and $\hat{x}_i$ as the reconstructed output the MSE is computed using the following formula:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{5}$$

where n is the total number of samples in the data.

The obtained MSE is then evaluated according to the fixed thresholds to determine if in the monitored window there is a sudden or gradual drift. The threshold T_R is computed using the following formula:

$$T_R = \mu + k * \delta \tag{6}$$

μ represents the mean, δ is computed as the standard deviation of historical errors and k is empirically determined. We considered different values for k in order to optimize the performance of the proposed approach. Specifically, T_R assumed values included in the range $(0, 2)$.

The performance of the proposed approach has been evaluated using very known metrics: precision, recall, F1-score. Precision represents the ratio between the number of right-identified concept drifts and the total number of wrongly and right-identified concept drifts. The recall is calculated by dividing the number of correctly identified concept drifts by the total number of concept drifts in the monitored data stream. Finally, the F1-score is the harmonic mean of Precision and Recall.

VI. Results and Discussion

In this section, the results of the proposed experiments are described and discussed.

Fig. 3a reports the F1-score distribution for gradual drift detection across the experimental runs. The distribution is obtained using the original dataset as the golden standard and in the figure, the obtained values across different thresholds are reported. Similarly, Fig. 3b shows the F1-score distribution obtained in sudden drift detection across different threshold values. The figures show that for gradual drift detection, the best median value of the F1-score is equal to 0.92. This value is obtained by fixing the threshold in the range [0.4,0.5). For

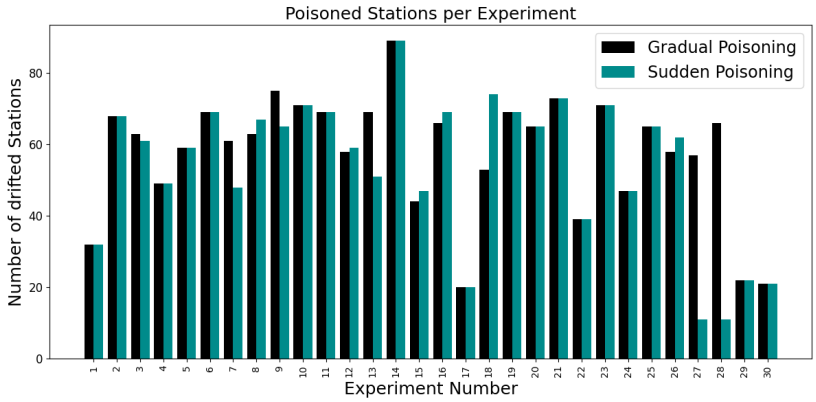
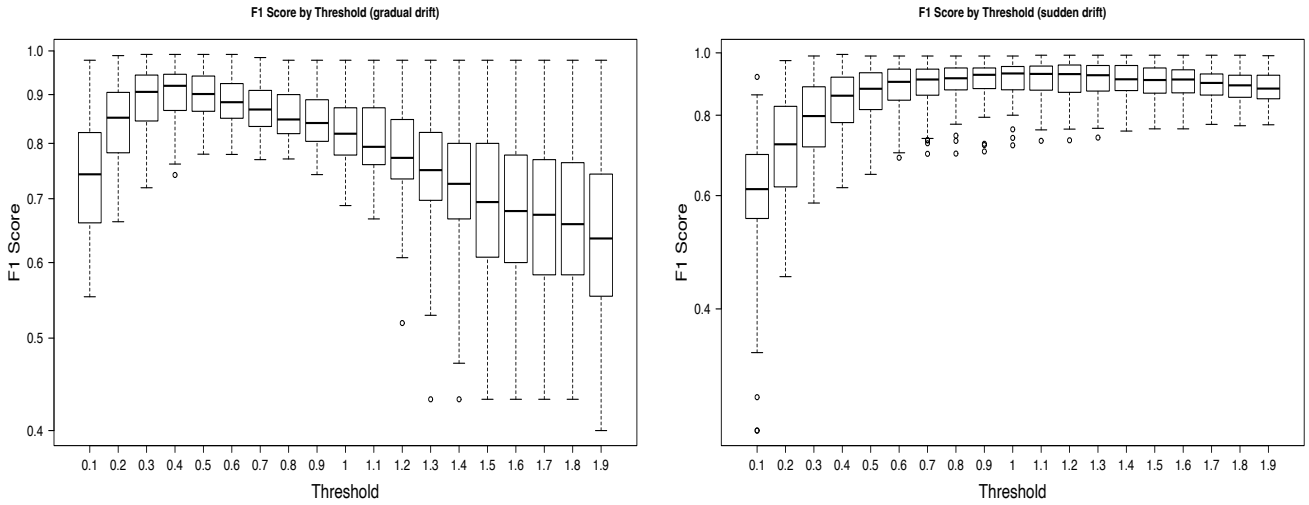


Fig. 2: Number of drifted stations (gradual and sudden drifts) for experimental run.



(a) Gradual drift

(b) Sudden drift

Fig. 3: F1-score distributions for gradual drift and sudden drift by threshold T_{mse}.

sudden drifts, the best F1-score is 0.94 obtained using a threshold in the range [0.9,1.3). Combining these results, we observed that fixing the threshold to 0.65 optimizes the performance of the proposed approach when both the drift types are considered to obtain a best F1-score of 0.89.

Notice that the distribution of the F1-score, in all the considered threshold ranges, has an interquartile included in the interval $[0.7, 0.9]$ for gradual drift and in the interval $[0.82, 0.93]$ for sudden drifts. This demonstrates that the drift detection is almost stable in both conditions and for the sudden drift, it is more stable allowing less constrained selection.

VII. Conclusions and Future Work

This paper introduces a novel drift detection approach in the context of federated learning based on the adoption of a transformer autoencoder. The validation of the proposed approach is performed on a large extended real-world traffic dataset. The results of the experiments show the effectiveness of the approach in detecting both sudden and gradual drifts. Specially, we observed F1-score of 0.89 in the best configuration setting (threshold of 0.65). The experiments also show that the drift detection approach is almost stable when different setting conditions are used.

Future work aims to extended the proposed validation on a large number of scenarios, several datasets and different configuration settings. Moreover, in the future work, new types of concept drifts will be considered in order to explore the capability of the proposed approach to detect more complex drifts.

Furthermore, the proposed approach will be validated on datasets with natural variations (e.g., seasonal traffic variations) to strengthen its generalizability. Finally, it will be analyzed how false drift detections could impact real-world systems, such as traffic management.

References

[1] Dor Bank, Noam Koenigstein, and Raja Giryes. *Autoencoders*, pages 353–374. Springer International Publishing, Cham, 2023.

[2] Firas Bayram, Bestoun S. Ahmed, and Andreas Kassler. From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems*, 245:108632, 2022.

[3] Mario Luca Bernardi, Marta Cimitile, and Muhammad Usman. Dqfed: A federated learning strategy for non-iid data based on a quality-driven perspective. In *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2024, Yokohama, Japan, June 30 - July 5, 2024*, pages 1–8. IEEE, 2024.

[4] Gabriele Borg and Charlie Abela. Graph based traffic analysis and delay prediction. *arXiv e-prints*, pages arXiv–2410, 2024.

[5] Fernando E. Casado, Dylan Lema, Marcos F. Criado, Roberto Iglesias, Carlos V. Regueiro, and Senén Barro. Concept drift detection and adaptation for federated and continual learning. *Multimedia Tools and Applications*, 81(3):3397–3419, 2022.

[6] Paulo M. Gonçalves, Silas G.T. de Carvalho Santos, Roberto S.M. Barros, and Davi C.L. Vieira. A comparative study on concept drift detectors. *Expert Systems with Applications*, 41(18):8144–8156, 2014.

[7] Salvatore Greco, Bartolomeo Vacchetti, Daniele Apiletti, Tania Cerquitelli, et al. Driftlens: A concept drift detection tool. In *EDBT*, pages 806–809, 2024.

[8] István Hegedus, Lehel Nyers, and Róbert Ormándi. Detecting concept drift in fully distributed environments. In *2012 IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics*, pages 183–188, 2012.

[9] Fabian Hinder, Valerie Vaquet, and Barbara Hammer. One or two things we know about concept drift—a survey on monitoring in evolving environments. part b: locating and explaining concept drift. *Frontiers in Artificial Intelligence*, 7, 2024.

[10] Ellango Jothimurugesan, Kevin Hsieh, Jianyu Wang, Gauri Joshi, and Phillip B. Gibbons. Federated learning under distributed concept drift, 2023.

[11] Myeongkyun Kang, Soopil Kim, Kyong Hwan Jin, Ehsan Adeli, Kilian M. Pohl, and Sang Hyun Park. Fednn: Federated learning on concept drift data using weight and adaptive group normalizations. *Pattern Recognition*, 149:110230, 2024.

[12] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, 2018.

[13] Dianqi Liu, Liang Bai, Tianyuan Yu, and Aiming Zhang. Towards method of horizontal federated learning: A survey. In *2022 8th International Conference on Big Data and Information Analytics (BigDIA)*, pages 259–266, 2022.

[14] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12):2346–2363, 2018.

[15] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, João Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363, 2019.

[16] Ning Lu, Guangquan Zhang, and Jie Lu. Concept drift detection via competence models. *Artificial Intelligence*, 209:11–28, 2014.

[17] H. B. McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *ArXiv*, abs/1602.05629, 2016.

[18] Russel Pears, Sripirakas Sakthithasan, and Yun Sing Koh. Detecting concept change in dynamic data streams. *Mach. Learn.*, 97(3):259293, December 2014.

[19] Leyla Rahimli, Feras M Awaysheh, Sawsan Al Zubi, and Sadi Alawadi. Federated learning drift detection: An empirical study on the impact of concept and data drift. In *2024 2nd International Conference on Federated Learning Technologies and Applications (FLTA)*, pages 241–250. IEEE, 2024.

[20] Sakti Saurav, Pankaj Malhotra, Vishnu TV, Narendhar Gugulothu, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Online anomaly detection with concept drift adaptation using recurrent neural networks. In *Proceedings of the acm india joint international conference on data science and management of data*, pages 78–87, 2018.

[21] Zheng Shi, Yingjun Zhang, Jingping Wang, Jiahu Qin, Xiaoqian Liu, Hui Yin, and Hua Huang. Dagcrn: Graph convolutional recurrent network for traffic forecasting with dynamic adjacency matrix. *Expert Systems with Applications*, 227:120259, 2023.

[22] Elena Tsiporkova, Michiel De Vis, Sarah Klein, Anna Hristoskova, and Veselka Boeva. Mitigating concept drift in distributed contexts with dynamic repository of federated models. In *2023 IEEE International Conference on Big Data (BigData)*, pages 2690–2699, 2023.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[24] Shuo Wang, Leandro L. Minku, and Xin Yao. A systematic study of online class imbalance learning with concept drift. *IEEE Transactions on Neural Networks and Learning Systems*, 29:4802–4821, 2017.

[25] Wenhao Wu, Weiwei Wang, Xixi Jia, and Xiangchu Feng. Transformer autoencoder for k-means efficient clustering. *Engineering Applications of Artificial Intelligence*, 133:108612, 2024.

[26] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*, 2021.

[27] Fahri Anıl Yerlikaya and Şerif Bahtiyar. Data poisoning attacks against machine learning algorithms. *Expert Syst. Appl.*, 208(C), December 2022.