Adapt but Do Not Forget: Towards Enhancing Drift Handling in 6G Networks

Mazene Ameur*, *Member, IEEE*, Bouziane Brik‡, *Senior Member, IEEE,* and Adlen Ksentini*, *Senior Member, IEEE*
**Communication Systems Department, EURECOM, Sophia Antipolis, France.*
‡Computer Science Department, University of Sharjah, Sharjah, UAE.
Emails: firstname.lastname@eurecom.fr, bbrik@sharjah.ac.ae

***Abstract*—The advent of 6G represents a paradigm shift where Artificial Intelligence and Machine Learning AI/ML becoming the pulsating heart of network innovation. However, these AI/ML models face a significant challenge known as Model Drift, caused by changes in data distributions and feature relationships over time, leading to degraded performance. While numerous drift adaptation strategies have been explored including model retraining, incremental learning, and ensemble approaches, they often neglect the critical issue of *"Catastrophic Forgetting" (CF)*. The latter refers to the phenomenon where neural networks, upon learning new information (post-drift), lose the ability to retain previously learned knowledge (pre-drift). This challenge is particularly pronounced in the time-series forecasting tasks inherent to telecommunications networks, where sequential data dependencies amplify the forgetting issue. To address this limitation, we propose a novel LSTM-EWC model that synergistically combines the time-series processing strengths of Long Short-Term Memory (LSTM) networks with the CF mitigation capabilities of Elastic Weight Consolidation (EWC). The proposed framework empowers LSTM models to retain essential pre-drift knowledge, while effectively adapting to post-drift changes, enabling robust and continuous learning in dynamic environments. Experimental evaluations on a real-world telecommunications dataset highlight the efficacy of our approach, demonstrating up to 51% improvement in knowledge retention over state-of-the-art CF baseline methods.**

***Index Terms*—Catastrophic Forgetting, Drift Adaptation, Elastic Weight Consolidation, Trustworthy AI, 6G Networks.**

I. Introduction

With 6G standards and specifications still taking shape, experts unanimously anticipate it will mark a revolutionary leap in technology. Central to this transformation are Artificial Intelligence (AI) and Machine Learning (ML) models, which serve as critical enablers for the seamless operation of these networks [1]. However, the dynamic nature of 6G environments poses significant challenges like model drift, which leads to performance degradation in AI/ML models [3]. This degradation can cause severe issues, including security vulnerabilities and QoS breaches. To address these, effective drift detection and handling mechanisms are crucial for maintaining reliable model performance in 6G networks [4]. Numerous drift adaptation methods have emerged in various fields, including Internet of Things (IoT) [2], healthcare [3], and 5G networks [4]. One common approach involves complete model retraining, which is straightforward but computationally expensive and time-consuming, thereby limiting its practicality [2]. In contrast, incremental learning techniques, such as those proposed by Yang et al. [5], continuously update the model's knowledge without complete retraining. Additionally, adaptive online learning techniques [6] adjust model parameters or strategies dynamically, while ensemble methods [7] handle different drift manifestations across applications.

A. Motivation & Research Gap

Despite the effectiveness of the aforementioned drift handling methods, they often overlook "Catastrophic Forgetting (CF)", a significant challenge in neural network-based models [8]. CF involves the abrupt loss of previously learned information when models learn new tasks, which is particularly problematic in 6G networks where knowledge continuity is crucial. This issue undermines model performance and compromises drift adaptation techniques, detracting from their goal of mitigating drift [10]. Indeed, addressing CF is essential to maintain a balance between learning new patterns and preserving existing knowledge, ensuring model reliability in dynamic applications. Numerous approaches have been proposed to address CF in continual learning [8]. Notable methods include Elastic Weight Consolidation (EWC) [16] which safeguards critical model parameters by modulating their plasticity based on their relevance to prior tasks. Additionally, replay-based approaches, like rehearsal [8] and generative replay [10], revisit stored data or generate synthetic samples to strengthen earlier knowledge retention. These methods enhance a model's capacity to maintain previously acquired knowledge while adapting to new information. Notably, the majority of these methods are narrowly focused on specific use cases, predominantly centered around imagery datasets.

B. Novelty

Despite numerous efforts to address CF and drift handling independently [11], a mature and widely adopted solution for mitigating CF within drift adaptation methods for 6G applications remains lacking. In the broader domain of continual learning, an examination of existing literature reveals that most proposed methodologies are predominantly confined to conventional image datasets (e.g. MNIST, PermutedMNIST [12]) and lack exploration of networking applications relevant to 5G and beyond. ***To the best of our knowledge, there has been minimal to no focus on addressing these challenges in dynamic, time-series datasets, particularly in the context of next-generation networks (i.e. 6G networks).*** This gap highlights the pressing need for research into CF and model drift adaptation, which are crucial for the reliability of AI models in dynamic networks.

C. Key Contributions

The key contributions of this paper can be summarized as follows:

- Shedding light and modeling the often-overlooked challenge of CF in time-series neural network models (i.e. LSTMs) when handling drifts while demonstrating their inherent susceptibility to CF.
- Introducing, for the first time, a novel CF-aware drift adaptation approach that integrates the temporal capabilities of LSTM with EWC to address CF in existing drift adaptation methods for 6G time-series data.
- The evaluation of the proposed LSTM-EWC method on a realistic telecommunication dataset compellingly highlights its effectiveness in mitigating knowledge loss across three distinct applications. Notably, the method achieves up to a 51% performance improvement over CF baseline approaches, underscoring its superior capability to preserve learned knowledge while adapting to new tasks.

The remainder of this paper is structured as follows: Section II presents the proposed LSTM-EWC system modeling and design. In Section III, we present the experiment setup, detailing the dataset and model implementation. Section IV presents the performance results of various experiments. Finally, Section V concludes and wraps up the paper.

II. The Proposed LSTM-EWC Method

Our method tackles the challenge of CF in LSTM networks when adapting to drifts in dynamic and evolving scenarios. For starters, LSTMs are well-suited for time series analysis [14], but their ability to adapt to changes in sequential tasks is hindered by CF, where learning new information often disrupts previously acquired knowledge [13]. By integrating EWC (i.e. regularization technique) into LSTM-based models, we ensure that adaptation to new tasks does not come at the expense of previously learned insights, making the method particularly effective in dynamic environments.

EWC achieves this by identifying critical model parameters using the Fisher Information Matrix (FIM), which quantifies their importance based on prior tasks [8], [16]. Changes to these crucial parameters are restricted during training, preventing the model from "forgetting" earlier tasks. As shown in Fig. 1, this approach ensures that parameter updates remain within shared low-error regions across tasks, allowing for efficient adaptation to evolving tasks while preserving historical insights. By leveraging EWC's ability to constrain destructive updates, LSTM-EWC strikes a balance between stability and adaptability [16], enabling consistent and reliable performance across sequential tasks in environments characterized by temporal variability and drift.

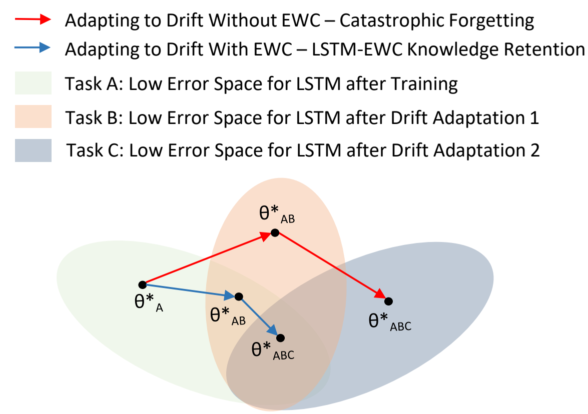

Fig. 1: Overlap of potential LSTM model parameter configurations θ^* during drift adaptation, with and without EWC.

A. System Model

This section presents the formulation of our LSTM-EWC system, meticulously designed to address CF in drift adaptation scenarios for 6G throughput forecasting.

*1) **Modeling the Input Data**:* The input data for the throughput prediction task is represented as a time series dataset $\mathcal{D} = \left\{ \boldsymbol{X} \in \mathbb{R}^{T \times d} \right\}$, where T is the number of time steps, and d is the number of features at each time step.

Each row $\boldsymbol{X}_t = [x_{t1}, x_{t2}, \ldots, x_{td}]$ corresponds to the feature vector at time step t. For ease of reference, we write it in block form:

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{T1} & x_{T2} & \cdots & x_{Td} \end{bmatrix}. \tag{1}$$

The corresponding target outputs are represented as a vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_T]^\top$, where y_t represents the download bitrate at time t.

*2) **LSTM Model**:* The LSTM model is designed to process sequential data and learn temporal dependencies. The LSTM computes a sequence of hidden states $\boldsymbol{h}_t$ and cell states $\boldsymbol{c}_t$ iteratively using the following equations [12]:

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{U}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right), \quad \text{(forget gate)} \tag{2}$$

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right), \quad \text{(input gate)} \tag{3}$$

$$\tilde{\boldsymbol{c}}_t = \tanh\left(\boldsymbol{W}_c \boldsymbol{x}_t + \boldsymbol{U}_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c\right), \quad \text{(candidate cell state)} \tag{4}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t, \quad \text{(cell state update)} \tag{5}$$

$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{U}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right), \quad \text{(output gate)} \tag{6}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t), \quad \text{(hidden state update)} \tag{7}$$

Here:

- $\boldsymbol{W}_*$, $\boldsymbol{U}_*$ are weight matrices, and $\boldsymbol{b}_*$ are bias vectors for each gate.
- $\sigma(\cdot)$ is the sigmoid activation function, and $\odot$ denotes element-wise multiplication.

The final output at each time step t is computed as:

$$\hat{y}_t = \boldsymbol{W}_y \boldsymbol{h}_t + \boldsymbol{b}_y, \tag{8}$$

where $\boldsymbol{W}_y$ and $\boldsymbol{b}_y$ are the weights and bias for the output layer.

The loss function used to train the LSTM is the Mean Squared Error (MSE), defined as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{T}\sum_{t=1}^{T}\left(y_t - \hat{y}_t\right)^2, \tag{9}$$

where $\hat{y}_t$ is the predicted download rate, and y_t is the actual value.

*3) **EWC Modeling**:* As discussed previously, EWC prevents CF by preserving important parameters learned from previous data. The importance of each parameter is quantified using the FIM. The general EWC objective function is defined as [16]:

$$\mathcal{L}_{\text{EWC}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{task}}(\boldsymbol{\theta}) + \frac{\lambda}{2}\sum_i F_i(\theta_i - \theta_i^*)^2, \tag{10}$$

where:

- λ is a regularization coefficient.
- F_i is the Fisher Information for parameter θ_i.
- θ_i^* are the parameters optimized for a previous task.

The FIM F is defined as [16]:

$$F_i = \mathbb{E}_{\boldsymbol{x},y}\left[\left(\frac{\partial \log p(y|\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_i}\right)^2\right]. \tag{11}$$

Here, F_i represents the Fisher Information corresponding to parameter θ_i. The expectation $\mathbb{E}_{\boldsymbol{x},y}$ is computed over the data points $\boldsymbol{x}$ and their corresponding labels y. Intuitively, F_i captures the sensitivity of the likelihood function $\log p(y|\boldsymbol{x};\boldsymbol{\theta})$ with respect to θ_i. Parameters with higher F_i values are considered more critical for the previously learned tasks.

*4) **LSTM-EWC Model**:* The proposed LSTM-EWC model combines the LSTM loss function with the EWC regularization term. The loss function is given by:

$$\mathcal{L}_{\text{LSTM-EWC}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{MSE}}(\boldsymbol{\theta}) + \frac{\lambda}{2}\sum_i F_i(\theta_i - \theta_i^*)^2, \tag{12}$$

where:

- $\mathcal{L}_{\text{MSE}}$ is the LSTM's time series prediction loss.
- The second term enforces the EWC regularization.

Therefore, the final LSTM-EWC loss is presented as:

$$\mathcal{L}_{\text{LSTM-EWC}}(\boldsymbol{\theta}) = \frac{1}{T}\sum_{t=1}^{T}\left(y_t - \hat{y}_t\right)^2 + \frac{\lambda}{2}\sum_i F_i(\theta_i - \theta_i^*)^2, \tag{13}$$

The EWC regularizer ensures convergence to an optimal parameter set $\boldsymbol{\theta}$, that balances performance across tasks if λ is sufficiently large [8], [16].

B. System Design & Use case

As discussed previously, the proposed LSTM-EWC method is designed to mitigate CF in LSTM models, particularly for sequential tasks like throughput forecasting. The primary objective is to prevent the loss of previously acquired knowledge when adapting to data drift. As detailed in Algorithm 1, for each sequence in the time series forecasting task, the MSE loss (eq. 9) is calculated to optimize the LSTM for the regression objective (i.e. Throughput forecasting in our case). For all tasks after the first, an additional EWC loss term (eq. 10) is computed using the FIM (eq. 11), which quantifies the importance of each network parameter for prior tasks. The total loss (eq. 13) which is a combination of the MSE loss and the EWC penalty, guides the network updates. The EWC mechanism ensures that only less critical parameters (those with lower Fisher Information values) are adjusted significantly during training, effectively preserving the functionality of neurons that are critical to previously learned tasks.

Algorithm 1: LSTM-EWC Drift Adaptation

Input: Dataset $\mathcal{D} = \left\{\boldsymbol{X} \in \mathbb{R}^{T\times d}\right\}$, learning rate α, EWC strength λ, Fisher Information F.
Output: LSTM parameters θ_t for each task t

$\theta_0 \leftarrow$ Initialize;
for $t = 1, \ldots, T$ **do**
  $\theta_t \leftarrow \theta_{t-1}$;
  foreach *Sequence* $\mathcal{S} \subset \mathcal{D}_t$ **do**
    $\mathcal{L}_{\text{MSE}} = \frac{1}{T}\sum_{t=1}^{T}\left(y_t - \hat{y}_t\right)^2$;
    if $t > 1$ **then**
      $\mathcal{L}_{\text{EWC}}^t = \sum_i \frac{\lambda}{2} F_{t-1,i}\left[(\mu_{t-1,i} - \theta_{t,i})^2\right]$;
    else
      $\mathcal{L}_{\text{EWC}}^t = 0$;
    $\mathcal{L}_{\text{LSTM-EWC}} = \mathcal{L}_{\text{MSE}} + \frac{\lambda}{2}\sum_i F_i(\theta_i - \theta_i^*)^2$;
    $\theta_t \leftarrow \theta_t - \alpha\nabla_{\theta_t}\mathcal{L}_{\text{LSTM-EWC}}^t$;
  $\boldsymbol{\theta}_t^* \leftarrow \boldsymbol{\theta}_t$; // Store model parameters
  $F_i = \mathbb{E}_{\boldsymbol{x},y}\left[\left(\frac{\partial \log p(y|\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_i}\right)^2\right]$; //Compute FIM

III. Performance Evaluation

A. Datasets

In this study, we leveraged a real-world, publicly available dataset from an Irish mobile operator, analyzing throughput, channel, and context information [15]. The dataset captures several timely throughput traces of three applications (i.e. Download file and 2 Streaming applications) with two mobility patterns (Static, and Mobile). In our experiments, we refer to the various applications by the following names for simplicity: **Application 1** refers to the Download Application in static mobility mode. **Application 2** refers to the Netflix streaming service in drive mobility mode. **Application 3** refers to the Amazon Prime streaming service in drive mobility mode. Due

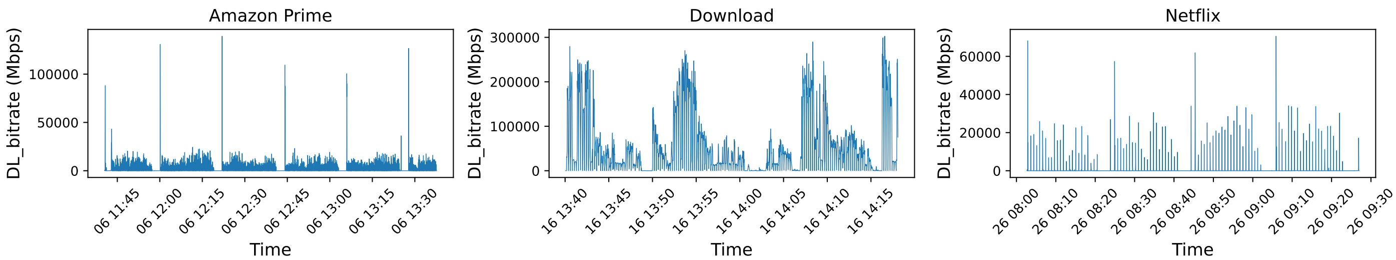

Fig. 2: Downlink bitrate temporal pattern study of the three applications (Amazon Prime streaming, Download File, and Netflix streaming) used in our experiments.

to page constraints, we kindly refer readers to the dataset paper [15] for detailed exploration.

B. Experimental Setup

To conduct our experiments, we implemented a baseline LSTM model and extended it by integrating various CF methods, including EWC (Our method) and rehearsal strategies [8]. The LSTM architecture comprised three layers with a total of 30,651 trainable parameters, designed to capture temporal dependencies within the dataset. For the LSTM-EWC model, the EWC penalty was set to $\lambda = 100$. The FIM was estimated using the first 1000 samples from each application type, and the coreset size was set to 500. Both the LSTM and LSTM-EWC models were trained for 100 epochs with a sequence length of 10. The average test accuracies across these tasks were computed, with results averaged over three runs. All experiments were executed on a single machine equipped with an 8-core 3.2 GHz AMD CPU, 16 GB of RAM, and a 4GB NVIDIA GeForce RTX 3050 Ti GPU.

IV. Experimental Results

In this section, we comprehensively examine the experimental results, beginning with dataset temporal analysis and LSTM model validation. We then delve into drift detection and adaptation, extensively studying the impact of CF on drift adaptation methods. Finally, we evaluate the proposed LSTM-EWC model, comparing its effectiveness in mitigating CF against existing solutions.

1) Dataset time-series Patterns: Fig. 2 highlights the time-series nature of the dataset for three applications: Amazon Prime, Download, and Netflix. Each application exhibits distinct temporal traffic patterns, with Amazon Prime showing periodic bursts, Download demonstrating highly volatile and bursty behavior, and Netflix displaying steady traffic interspersed with moderate spikes. This emphasizes the variability in traffic demands over time for different applications.

2) LSTM Model Validation: The comparison between the actual and predicted downlink bitrate over time, using an LSTM model for time series data, is illustrated in Fig. 3. Overall, the model demonstrates strong performance, as the predicted values (dashed blue line) closely follow the patterns and fluctuations

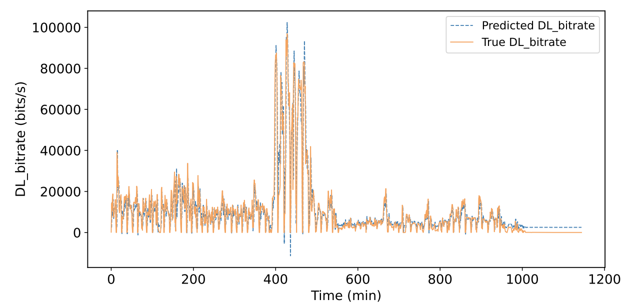

Fig. 3: LSTM model validation: Predicted vs Actual accuracy plot.

of the actual data (solid brown line) throughout the time interval. This alignment indicates the model's ability to effectively capture and generalize the variations in DL_bitrate (download bitrate).

3) Drift Detection: For this experiment, We simulated both data and concept drifts by changing data distribution and application context, such as training on static mobility download data and then testing on streaming data with driving mobility conditions. Fig. 4 shows the RMSE (Root Mean Squared Error) for two applications: Application 1 (brown) and Application 2 (blue), with an RMSE drift baseline (red dashed line) for reference. Initially, the LSTM model performs well for Application 1, as indicated by consistently low RMSE values, staying well below the baseline. However, once the model is exposed to Application 2 (after the vertical dashed line), its performance deteriorates significantly, with RMSE values spiking and fluctuating far above the baseline. This sharp decline in performance signals a clear drift, suggesting that the model, trained or fine-tuned on Application 1, fails to generalize effectively to Application 2 due to differences in underlying data characteristics. This highlights the need for model retraining or adaptation to handle the drift between the applications.

4) Effects of Catastrophic Forgetting on Drift Adaptation Methods: After the performance drop caused by concept/data drift when the model was exposed to application 2 data, we applied a drift handling technique that fine-tunes the LSTM model with new data. As depicted in Fig. 5, we can observe the

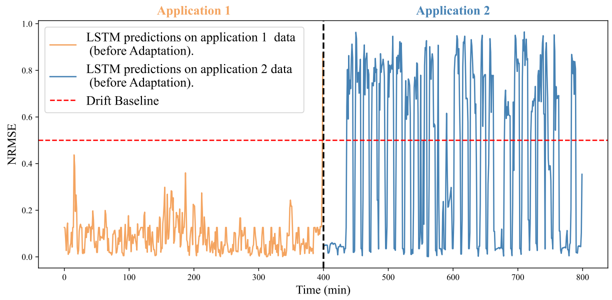

Fig. 4: Performance of the LSTM Model when first exposed to application 2 data.

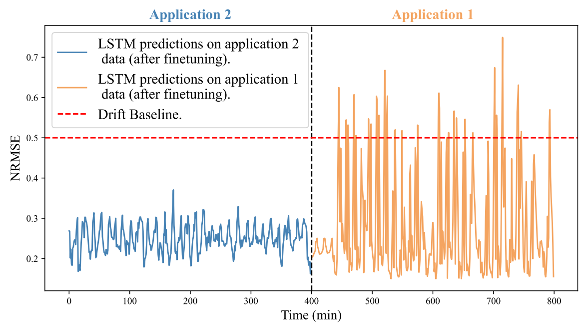

Fig. 5: Performance of the LSTM model after finetuning with application 2 data.

model's performance after fine-tuning it on application 2 data and then re-exposing it to application 1 data. The plot reveals a significant decline in the model's performance when tested on application 1 data after fine-tuning on application 2 data. This performance drop clearly showcases that the knowledge the model previously acquired from application 1 was forgotten. Such a decline emphasizes the susceptibility of current drift adaptation methods in LSTM models to CF, which is particularly problematic in dynamic environments. This phenomenon is particularly problematic in dynamic environments where models must continually adapt to new data without sacrificing previously gained insights.

5) Effectiveness of the Proposed LSTM-EWC: To address the challenges of CF in LSTM drift adaptation methods, we tested the proposed LSTM-EWC model under the same conditions as the previous experiment. Hence, after detecting the drift, we fine-tuned the LSTM-EWC model by applying EWC to the LSTM, as detailed in Algorithm 1. Fig. 6 illustrates the performance of the LSTM-EWC model after fine-tuning on application 2 data. Subsequently, we re-exposed the model to application 1 data. Initially, before applying EWC (Fig. 5), the model's performance on application 1 data dropped significantly. This indicated a substantial loss of previously acquired knowledge due to drift. However, after fine-tuning the model with EWC, the performance on application 1 data was markedly improved even

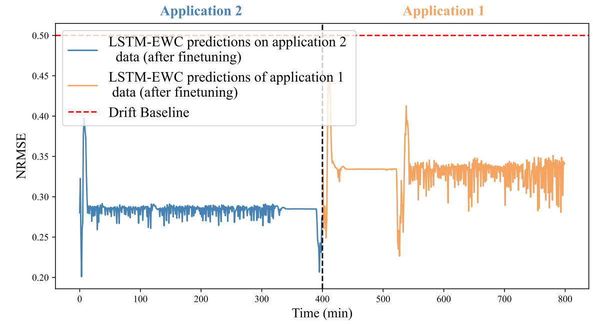

Fig. 6: Performance of the proposed LSTM-EWC model after finetuning with application 2 data.

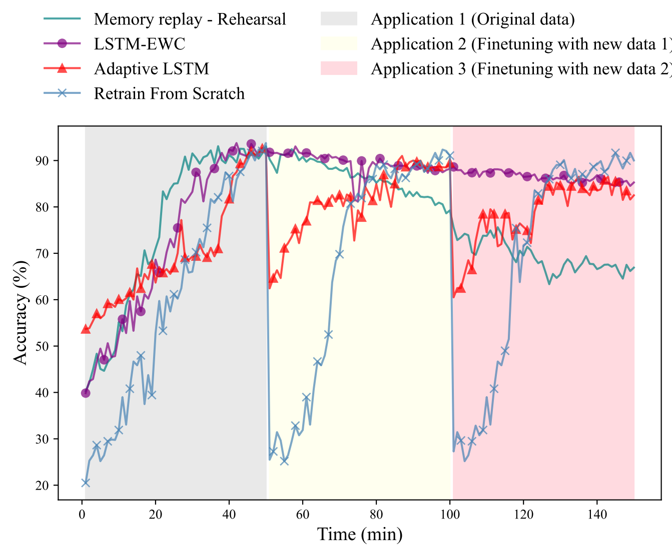

Fig. 7: Comparative Analysis with/without Applying EWC to the LSTM with the three applications.

after being fine-tuned on application 2 data as Fig. 6 depicts. This outcome highlights the LSTM-EWC model's capability to continue performing well on previously seen data, despite minor decreases in performance. The model remains well within acceptable thresholds, showcasing that it can retain knowledge from previous tasks. Therefore, the LSTM-EWC model presents a promising solution for current LSTM-based drift adaptation methods that suffer from CF.

A. *Comparative Analysis*

We conducted extensive experiments to evaluate the LSTM-EWC CF-aware drift adaptation method, comparing it against state-of-the-art approaches including (i) Adaptive LSTMs, (ii) Replay Memory, and (iii) Complete Retraining, originally from the imagery domain and adapted to our context [8]. These methods were tested across three applications: Download, Netflix, and Amazon Prime. To assess a broad spectrum of scenarios, we tested the applications in both static and drift modes, encompassing various drift patterns. To improve visualization and readability, we converted NRMSE into an accuracy-like percentage using Accuracy $= (1 - \text{NRMSE}) \times 100\%$. As shown in Fig. 7, the experimental results offer a detailed comparison of the performance of our LSTM-EWC method against the aforementioned state-of-the-art approaches. The background colors (gray, yellow, and pink) indicate transitions between three different tasks (Application 1, 2, and 3 respectively). The LSTM-EWC method consistently outperforms other CF-mitigating solutions across different applications. Notably, the LSTM-EWC model maintains an impressive accuracy of around 89% while exposed to dataset 2 (task 2) and over 85% when exposed to the third dataset (task 3). In contrast, alternative methods experience significant declines in accuracy when transitioning between tasks. For example, the retrain approach and the Adaptive LSTM see accuracy drops of up to 25% and 61%, respectively, due to complete retraining and the constant fluctuation adjustment of model weights. Furthermore, although the rehearsal method performs well in the second application, these methods start to falter in the third application. This decline is due to the replay buffer accumulating a larger pool of values, causing the model to lose accuracy. In summary, these findings highlight the resilience and effectiveness of our LSTM-EWC method in addressing CF. Across all tested applications, the LSTM-EWC model consistently outperformed alternative methods, demonstrating its ability to adapt to new data while retaining knowledge from previous tasks.

B. Discussion & Lessons Learned

The outcomes of our experiments provide meaningful insights and practical lessons for applying EWC to LSTM models. From these results, we can derive the following conclusions:

- **Susceptibility to Catastrophic Forgetting:** Indeed, drift adaptation methods for LSTM are susceptible to CF, which can lead to performance degradation.
- **Resilience to Catastrophic Forgetting:** The LSTM-EWC model demonstrated superior performance in retaining knowledge from previous tasks while adapting to new ones.
- **Applicability to 6G Networks** LSTM-EWC's ability to adapt to new data while retaining prior knowledge makes it a viable solution for dynamic and evolving time series environments, such as 6G networks.

V. Conclusion

In this paper, we unveil the critical yet underexplored challenge of CF in drift adaptation techniques, with a focused examination of time series LSTM models. This issue jeopardizes the robustness of AI models deployed in dynamic networks such as 6G. To address this challenge, we introduced a novel drift adaptation approach known as LSTM-EWC, designed to retain knowledge from previous tasks. This method leverages the temporal modeling capabilities of LSTMs with the robust regularization offered by EWC. Extensive experiments with a real-world public dataset demonstrated the effectiveness of the proposed LSTM-EWC model. The findings revealed that LSTM-EWC significantly enhances CF mitigation in dynamic time-series datasets, preserving knowledge by up to 51% compared to existing state-of-the-art CF methods.

For future work, we aim to extend our approach to a broader range of 6G use cases and investigate other strategies for mitigating CF in drift adaptation methods.

Acknowledgment

This work was supported by the European Union's Horizon Program under 6G-DALI (Grant Agreement No. 101192750) and SUNRISE-6G projects (Grant No. 101139257).

References

[1] M. Shadab, and L. Liu. "Revolutionizing future connectivity: A contemporary survey on AI-empowered satellite-based non-terrestrial networks in 6G." IEEE Communications Surveys and Tutorials (2024).

[2] Yang, Li, and Abdallah Shami. "A lightweight concept drift detection and adaptation framework for IoT data streams." IEEE Internet of Things Magazine 4.2 (2021): 96-101.

[3] A. Kore et al., "Empirical data drift detection experiments on real-world medical imaging data," Nat. Commun., vol. 15, no. 1, p. 1887, 2024, doi: 10.1038/s41467-024-46142-w.

[4] Manias, Dimitrios Michael, Ali Chouman, and Abdallah Shami. "A model drift detection and adaptation framework for 5g core networks." 2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom). IEEE, 2022.

[5] Yang, Zhe, et al. "A novel concept drift detection method for incremental learning in nonstationary environments." IEEE transactions on neural networks and learning systems 31.1 (2019): 309-320.

[6] G. I. Chaitanya and M. Webb, "Extremely fast decision tree," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018.

[7] H. M. Gomes, J. Read, and A. Bifet, "Streaming random patches for evolving data stream classification," in 2019 IEEE International Conference on Data Mining (ICDM), IEEE, 2019, pp. 240–249.

[8] Kemker, Ronald, et al. "Measuring catastrophic forgetting in neural networks." Proceedings of the AAAI conference on artificial intelligence. Vol. 32. No. 1. 2018.

[9] M. Ameur, B. Brik, and A. Ksentini, "Leveraging LLMs to eXplain DRL decisions for transparent 6G network slicing," in 2024 IEEE 10th International Conference on Network Softwarization (NetSoft), IEEE, 2024.

[10] D. Li et al., "Overcoming catastrophic forgetting during domain adaptation of Seq2seq language generation," in Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Stroudsburg, PA, USA, 2022.

[11] L. De Souza Alexandre, "Incremental learning approach using fuzzy logic to mitigate catastrophic forgetting," in Latinx in AI@ NeurIPS, 2024.

[12] M. Schak and A. Gepperth, "A study on catastrophic forgetting in deep LSTM networks," in Lecture Notes in Computer Science, Cham: Springer International Publishing, 2019, pp. 714–728.

[13] Arora, Gaurav, Afshin Rahimi, and Timothy Baldwin. "Does an lstm forget more than a cnn? an empirical study of catastrophic forgetting in nlp." Proceedings of the 17th Annual Workshop of the Australasian Language Technology Association. 2019.

[14] P. M. Tshakwanda, H. Kumar, S. T. Arzo, and M. Devetsikiotis, "Unveiling the future: A comparative analysis of LSTM and SP-LSTM for network traffic prediction in 6G networks," in ICC 2024 - IEEE International Conference on Communications, IEEE, 2024, pp. 2919–2924.

[15] D. Raca, "Beyond throughput, the next generation: A 5G dataset with channel and context metrics," in Proceedings of the 11th ACM multimedia systems conference, 2020.

[16] Lesort, Timothée. Continual Learning: Tackling Catastrophic Forgetting in Deep Neural Networks with Replay Processes. Diss. Institut polytechnique de Paris, 2020.