Recommendation Systems with Non-stationary Transformer

Gangmin Li
HeXie Management Research Centre
College of Industry-Entrepreneurs (CIE),
Xi'an Jiaotong-Liverpool University
Suzhou, China
gangmin.li@xjtlu.edu.cn

Yuchen Liu
Department of Computing
Xi'an Jiaotong-Liverpool University
Suzhou, China
yuchen.liu21@student.xjtlu.edu.cn

***Abstract*— Recommendation systems rely on an accurate user model to understand users' needs to make a personal recommendation. Traditional user modeling uses users' past behaviors during a "supply-meets-demand" interaction. This approach failed to capture the dynamic and emergence of new items and the shifting of user interests. The recommendation systems, built based on this user model trap users in their previous interests and make recommendations without counting their interest shift. We propose a new approach that integrates a non-stationary transformer into a recommendation system to capture the temporal dynamics of supplies and shifting user interests. Our experiments demonstrate the framework's superiority over benchmark models. The empirical results confirm the efficacy of our proposed framework and significant performance enhancements for recommendations.**

Keywords—Non-stationary Transformer, Recommendation Systems, Deep Learning, Reinforcement Learning, User-centric Systems

I. INTRODUCTION (*HEADING 1*)

In today's rapidly evolving environment, recommending relevant information and products to consumers poses a challenge for any recommendation system. There are many efforts to overcome inaccurate user modeling in recommendation platforms from online marketplaces to news aggregation sites, and even short video platforms. These efforts focused on improving algorithms to meet their needs-and-supply matching requirements. These algorithms adopt advanced technologies such as deep learning. Many algorithms use deep learning to figure out what users are interested in such as Deep Factorisation Machines (DeepFM) [1], Wide & Deep [2], Deep Interest Evolution Networks (DIEN) [3], and Behaviour Sequence Transformer (BST) [4]. Other algorithms use reinforcement learning for sequential product recommendations, such as Exact-k [5] and traditional deep reinforcement learning algorithms: Double Deep Q-Network (DDQN) [6], Proximal Policy Optimisation (PPO) [7], Deep Deterministic Policy Gradient (DDPG) [8], aim to capture the dynamic elements related with new products, user interests shifting and to maximize matches between user needs and product supplies.

Traditionally, machine learning-based algorithms standardize data into a normalized, quasi-normal distribution. These algorithms are designed to enhance computational efficiency and accurate matchings between expected outcomes with the results produced by the algorithms. It inadvertently ignores the non-stationary characteristics of data that particularly represent new products. Models based on these algorithms trained on stable datasets often perform well only within those controlled environments. The introduction of non-stationary variables or a change in context can abruptly degrade models' performance and fail to capture and update user interest change.

This paper reports our effort to capture and preserve the intrinsic value of non-stationary data representing products and user interests in recommendation systems. By introducing Non-stationary Transformers into recommendation systems, we aim to enhance recommendation accuracy and robustness when facing dynamic and changing supplies. Empirical results from our experiments in deep learning contexts affirm the superiority of our proposed approach underscoring its potential applicability across various recommendation systems.

The paper is organized in the following: Section 2 reviews related work on recommendation systems, spotlighting various methodologies currently employed to use non-stationary data. Section 3, presents the structure of our model, detailing its integration with a deep learning-based model as an example. Section 4 presents our primary analytical discourse on our experiments. We validate our model's efficacy in predicting click-through rates within deep learning contexts. We demonstrate our model's superior performance on test datasets through comparative analysis. Section 5 concludes our current discourse, reflecting on the achievements while acknowledging its limitations and proposing future works.

II. RELATED WORK

Traditional recommendation system models have been influenced by the advancement of machine learning to deep learning and reinforcement learning as they are fast advanced in natural language processing (NLP) and that has attracted global attention. They undoubtedly influence the recommendation system development as well.

A. Deep Learning-based Recommendation System

The evolution of machine learning towards deep learning has significantly influenced the development of recommendation system. Traditionally grounded in machine learning and statistical algorithms, such as Collaborative Filtering (CF), Alternating Least Squares (ALS), and Factorisation Machines (FM), the field has witnessed the emergence of deep learning-based extensions that enhance these foundational models. The Factorisation-machine-supported Neural Network (FNN) represents one of FM's earliest deep learning expansions, laying the groundwork for subsequent improvements [9]. For instance, the Wide & Deep model merges the basic linear regression model with a Multi-Layer Perceptron (MLP), thereby harnessing memorization and generalization [2]. Building on these advancements, Deep Factorization Machines optimize the network structure further, integrating FM and deep neural networks to improve prediction accuracy [1]. Attentional Factorization Machines introduce an attention mechanism into the network, enabling the model to focus on relevant features dynamically [2]. Additionally, Graph factorization machines [10] incorporate graph neural network modules, enhancing the model's ability to leverage complex relational data. Parallel to these developments, deep learning-based sequence recommendation algorithms have also evolved, primarily leveraging Recurrent Neural Networks (RNNs). The Deep Interest Network (DIN) [11] enhances basic sequence models with an attention mechanism, focusing on capturing evolving user interests. This concept is further extended by the Deep Interest Evolution Network (DIEN) [3], which divides the model into layers for user behavior sequences, interest extraction, and interest evolution, addressing the dynamic nature of user preferences. The introduction of the transformer model into the development of the Behaviour Sequence Transformer (BST) [4], which employs the transformer encoder to integrate user historical interaction with user and item features. Furthermore, BERT4Rec [12] adapts the BERT model's capabilities for recommendation systems, showcasing the adaptability of deep learning innovations in this domain. These models represent the synergy between the advancements in deep learning techniques and their application in recommendation systems, leading to a new era of personalized and dynamic recommendations.

B. *Works with non-stationary data*

There are efforts to integrate non-stationary data into recommendation systems. Ye et al. introduce an adaptive case that employs a novel pruning algorithm for large-scale recommendation systems grappling with non-stationary data, effectively balancing model adaptability and computational efficiency [13]. Huleihel, Pal, and Shayevitz extend collaborative filtering techniques to accommodate the temporal variability in user preferences, enhancing recommendations' relevance and personalization [14]. Wu, Iyer and Wang propose a two-tiered hierarchical bandit algorithm to navigate the exploration-exploitation trade-off in environments characterized by non-stationarity and delayed feedback, facilitating more timely and contextually appropriate recommendations [15]. Chandak et al. address the challenge of delayed feedback in such settings with a stochastic, non-stationary bandit model that leverages intermediate observations to refine learning processes and decision-making [16]. Liu used non-stationary data for sentiment analysis and dynamic pricing [18, 19].

Despite the notable advancements, applications tend to be constrained by the specific contexts for which they were developed. Our research aims to bridge this gap by reinstating the non-stationary attributes of data within a more universally applicable model that any recommendation system can use. By integrating our model within deep learning the performance of traditional models that rely on stationary data processing can be improved. This enhancement not only underscores the robustness of the model but also its versatility across a broad spectrum of complex data scenarios.

III. METHODOLOGY

In this section, we will introduce the architecture of our new e-commerce recommendation system. The core concept is to add a non-stationary transformer as an extra layer. We will begin with a brief introduction to the non-stationary transformer, and then present our modified foundational structure. Finally, we demonstrate how it is integrated into a deep learning-based recommendation system, and its versatility and applicability.

A. *Modified Non-stationary Transformer*

The non-stationary transformer was initially introduced by Liu [17] to explore stationarity in time series forecasting. The architecture of the non-stationary transformers is shown in Figure 1. Where series rationalization is adopted as a wrapper on the base model to normalize each incoming series and de-normalize the output. De-stationary attention replaces the original Attention mechanism to approximate attention learned from un-stationarized series, which rescales current temporal dependency weights with learned de-stationary factors τ, Δ.

Inspired by this architecture, Our Non-stationary Transformer is simpler and its projector and encoder are described as follows:

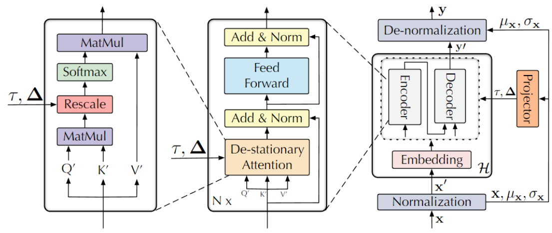

Fig. 1. Non-stationary Transformers architecture quoated from Liu [17] under CC BY-NC-ND 4.0.

Projector: Our projector is designed to detect and adapt to the evolving patterns within sequential datasets that describes item, user and interactions. The adaptation process commences with:

$$\mathbf{X}_{\text{reduced}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t \tag{1}$$

where $\mathbf{X}_{\text{reduced}}$ represents the dimensionally reduced data obtained through averaging over the temporal dimension T. Following this, the data is processed through a series of transformation layers, each comprising a dense neural network structure with Leaky ReLU activation, encapsulated as:

$$\mathbf{Y} = \text{Leaky ReLU}(\mathbf{W}_{\text{hidden}} \bullet \mathbf{X}_{\text{reduced}} + \mathbf{b}_{\text{hidden}}) \tag{2}$$

where $\mathbf{W}_{\text{hidden}}$ and $\mathbf{b}_{\text{hidden}}$ denote the weights and biases of the hidden layers, respectively. Then the final output, incorporating the essence of the non-stationary features, is rendered through:

$$\mathbf{Z} = \tanh(\mathbf{Y}) \tag{3}$$

where tanh represents the hyperbolic tangent function, encapsulating the detected non-stationary aspects.

Transformer Encoder: Our transformer encoder adds a self-attention mechanism specifically tailored for analysing complex sequential data. Integral to this encoder are the dynamic elements scale_learner and shift_learner. These elements are crucial for adapting to changes in data over time, with the scale_learner adjusting the significance of different temporal features and the shift_learner accommodating shifts in the data patterns or distributions. Together, they ensure the model's attention mechanism remains attuned to the evolving characteristics of the sequential data, as expressed by:

where σ_{enc} and μ_{enc} denote the standard deviation and

$$\log \tau = \text{scale\_learner}(x_{\text{raw}}, \sigma_{\text{enc}}),$$
$$\Delta = \text{shift\_learner}(x_{\text{raw}}, \mu_{\text{enc}}) \tag{4}$$

mean of the input sequences, respectively. The adapted attention mechanism in our dynamic structure is formulated to accommodate the intricacies of non-stationary data. Specifically, Q′, K′, and V′ represent stabilised versions of the queries, keys, and values obtained from the original dataset.

$$\text{Attn}(Q', K', V', \tau, \Delta) = \text{Softmax}\left(\frac{\tau \odot (Q'K'^{\top}) + \Delta}{\sqrt{d_k}}\right)V' \tag{5}$$

The attention function is represented as:

In (5), the operation $\tau \odot (Q'K'^{\top})$ effectively scales the dot product of the queries Q′ and keys K′ with the scaling factor τ, which is designed to adjust for time-varying aspects of the data. The term Δ introduces a shift to these scaled scores, further tailoring the attention scores to the non-stationary characteristics of the dataset. The normalization factor $\sqrt{d_k}$, where d_k denotes the dimensionality of the keys, ensures that the scaled dot products maintain a consistent variance, promoting stable gradients throughout the model. The softmax function is then applied to the resulting scores, converting them into a probability distribution. This step ensures that each value in the interval (0, 1) and the entire vector sums to 1. Finally, the attention scores are applied to the values V′ through a weighted sum. This multiplication aggregates the information across all values, weighted by their relevance as determined by the attention scores, culminating in the output of the attention mechanism. This output serves as a contextually enriched representation that synthesizes the most relative information from the input data, adjusted for both the temporal dynamics and the non-stationary features inherent in the dataset. To ensure the stability and prevent overfitting of the model, a combination of layer normalization and dropout is applied to the attention output:

$$X_{\text{final}} = \text{LayerNorm}(\mathbf{X} + \text{Dropout}(\text{Attn}(Q', K', V', \tau, \Delta))) \tag{6}$$

With this novel approach to handling non-stationary data through adaptive learning and dynamic adjustments, this architecture can be applied in designing transformer-based recommendation systems for complex and evolving datasets. To demonstrate this capability, we integrate this non-stationary transformer into an exemplary recommendation system in the next section.

B. *Integrate non-stationary transformer into a Recommendation System*

We integrate our non-stationary transformer into a deep learning-based recommendation system that adopts the BST algorithm. This integration is just for testing and illustration purposes. It involves replacing the conventional transformer layers with our non-stationary transformer to capture temporal dynamics and distributional shifts in user behavior sequences, Figure 2 is the illustration of the integration. It consists of three layers: Embedding Layer, Non-stationary Transformer Layer

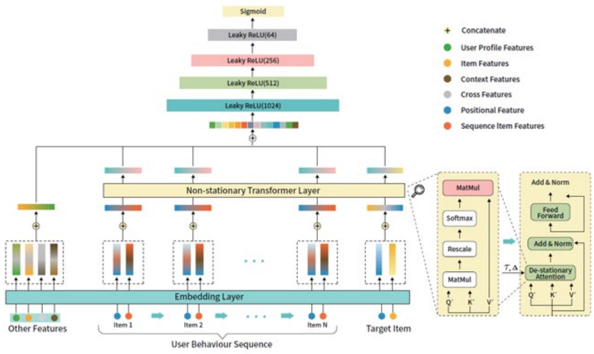

Fig. 2. Illustration of the Enhanced BST Model with Non-stationary Tranformer Integration.

Embedding Layer: The embedding layer initiates the adaptation process by transforming the multifaceted input data into compact, low-dimensional vector representations. The input data is categorized into three principal segments: (1) The core component comprises sequences of user-item interactions, encapsulating the dynamic interplay between user's selections among multiple alternative items over time. (2) Auxiliary features encompass a broad spectrum of attributes, including user demographics, product specifications, and contextual information, enriching the model's understanding beyond mere interaction patterns. (3) The target item features primarily focus on characteristics of new or prospective items that are subjects of prediction. Each of these segments undergoes a distinct embedding process, resulting in specialized embeddings that collectively form a comprehensive representation of the multifaceted input within our model. This embedding strategy is crucial for capturing the nuanced relationships and attributes inherent in user behavioral sequences, auxiliary features, and target items. In addition, to preserve the sequential essence of user interactions, we assign temporal values based on the chronological distance between items evolving in a user's interactions and the moment of selection to reflect the temporal sequence of user behaviors.

Non-stationary Transformer Layer: We introduced a non-stationary transformer layer to replace the traditional transformer. This replacement improves the model's ability to adapt temporal variations and data distribution shifts, enabling a deeper understanding of complex interaction-item relationships and user interaction patterns within a dynamically changing context.

Multiple-Perceptron layer: The final part of our architecture is the Multiple-Perceptron layer. It is a series of the Leaky ReLU functions designed for the binary classification predicting user clicks or product scores. This final ensemble leverages the enriched features processed through the non-stationary transformer layers, which can be used to produce precise andcontext-aware recommendations.

By adding the non-stationary transformer into the structure of the BST algorithm, our approach retains the original model's capability to process user behavior sequences. It significantly enhances the adaptability and predictive accuracy of user interaction. It is hoped that this novel integration will bring a significant improvement in deep learning-based recommendation systems, and result in superior performance in navigating the complexities of dynamic user behavior patterns

IV. EXPERIMENTS

To test the effectiveness of the integration, we have conducted tests to compare the performance of the proposed integration with the models without a non-stationary transformer. In our experimental analysis, we utilize a distinct dataset tailored to the specific for our study.

A. Tenrec Dataset: QK-video

Derived from Tencent's renowned recommendation platforms, QQ BOW (QB) and QQ KAN (QK), the QK-video dataset focuses on video recommendations, encapsulating a vast range of user interactions including clicks, likes, shares, and follows. The dataset has over 5 million users and 3.75 million items, resulting in a staggering 142 million clicks, alongside significant volumes of likes, shares, and follows. This extensive dataset, with its diverse feature set covering user demographics and item categories, is anonymized to ensure user privacy. Including both positive and negative feedback provides a holistic view of user preferences, which is pivotal for refining deep learning models within recommendation systems.

In our experiments, we divided the Tenrec video dataset into three subsets: 70% for training, 15% for validation, and 15% for testing. Table I provides a summary of these splits.

TABLE I. SUMMARY OF THE TENREC VIDEO DATASET SPLITS

Set	Records	Users	Items	Video Categories	User Features	Seq Interactions
Train	84,239,614	998,993	2,027,521	3	3	10
Validate	18,051,346	985,099	1,104,613	3	3	10
Test	18,051,346	985,315	1,104,678	3	3	10

We trained two models on the training set: the baseline BST model and our enhanced version, which incorporated the non-stationary transformer. As depicted in Figure 3.

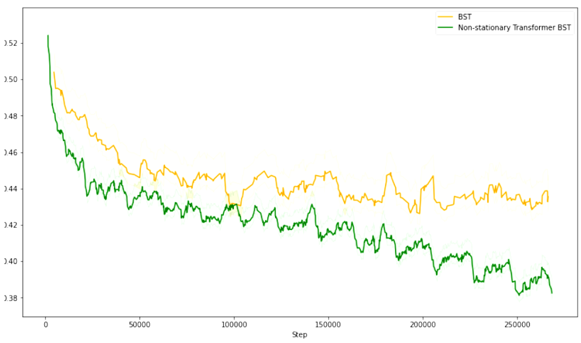

Fig. 3. Different Models Loss among Train Set

From Fig. 3 we can see the baseline BST model's loss gradually converges to approximately 0.43, while the Non-stationary Transformer BST model demonstrates a more significant loss reduction, converging around 0.39. This notable difference indicates that our non-stationary transformer BST model achieves a lower train loss overall, suggesting an enhanced learning efficiency. Then, we plan to extend this comparative analysis to the test set to validate the models' performance and ascertain whether the lower train loss translates to improved prediction accuracy on unseen data.

During the validation, we conducted trials to determine the optimal batch size, considering the immense scale of the Tenrec video dataset, which contains over a hundred million records. Guided by Tenrec's introduction, which suggests larger batch sizes for datasets of this magnitude, we experimented with batch sizes in powers of two: 1024, 2048, 4096, and 8192. Fig. 4 indicates that larger batch sizes facilitated a more rapid decrease in loss. However, upon evaluating the largest batch size of 8192, we observed the loss diminishing to near zero, indicating a potential risk of severe overfitting. Consequently, based on these observations, we identified 4096 as the most suitable batch size for our experiments, balancing efficient learning with the need to avoid overfitting.

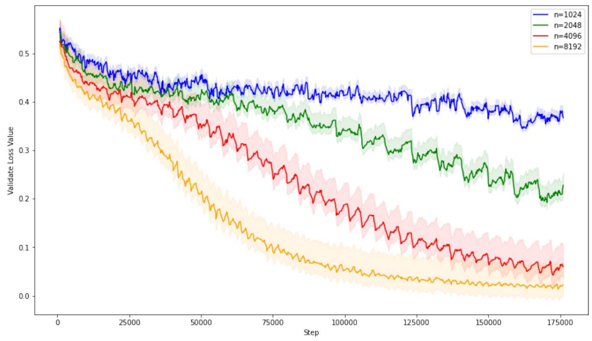

Fig. 4. Different Batch Sizes Loss among Validate Set

B. Comparison

To compare performance, we selected several performance metrics in the test, including Logloss, AUC, and F1 score. We compare the models' effectiveness with Tenrec data's baseline models, such as Wide & Deep [2], DeepFM [1], NFM [20], and DCN [21].

TABLE II. PERFORMANCE COMPARISON OF DIFFERENT MODELS

Model	Logloss	AUC	F1 score
BST	0.4808 (+8.31%)	0.7921 (-0.81%)	0.7386 (-2.79%)
NsT-BST	**0.4439**	**0.7986**	**0.7598**
Wide & Deep	0.51 (+14.89%)	0.7919 (-0.84%)	0.7225 (-4.91%)
DeepFM	0.508 (+14.51%)	0.793 (-0.70%)	0.7463 (-1.78%)
NFM	0.508 (+14.44%)	0.7957 (-0.36%)	0.7512 (-1.13%)
DCN	0.509 (+14.71%)	0.7927 (-0.74%)	0.7261 (-4.44%)

As Table II illustrates, our Non-stationary Transformer BST(NsT-BST) model achieved superior performance across all metrics. It outperformed the baseline BST model with a improvement of 8.31% in Logloss, an increase of 0.81% in the AUC, and a 2.79% rise in the F1 score. Moreover, compared with other benchmark models, our Non-stationary Transformer BST model demonstrated a clear advantage, yielding the lowest Logloss and the highest scores in both AUC and F1 metrics. Notably, it surpassed the Wide & Deep model by a substantial margin, with improvements of 14.89% in Logloss, 0.84% in AUC, and a significant 4.91% in the F1 score. Similar outperformance trends were observed against the DeepFM, NFM, and DCN models across all metrics. These results show the exceptional ability of the Non-stationary Transformer BST model to effectively predict click-through rates, showcasing its strength in handling the complex and dynamic nature of the data inherent in recommendation systems. Integrating the Non-stationary Transformer within the BST framework enhanced its learning efficiency on the training data. It solidified its robustness and accuracy, making it a superior model for the CTR prediction task.

V. Discussion

Our experiments in deep learning-related models that are frequently used in recommendation systems have revealed the enhanced performance of the Non-stationary Transformer BST model over the baseline BST. Notably, the Non-stationary Transformer BST model achieved lower Logloss and higher AUC and F1 scores, indicating superior predictive accuracy and classification quality. This improvement suggests that accommodating the non-stationary aspects of user interaction data can significantly enhance the effectiveness of recommendation systems. Moreover, the observed benefits were consistent across various baseline models, emphasising the robustness and generalisability of our proposed approach.

However, it is our desire to firmly approve the integration of the Non-stationary Transformer into recommendation systems can capture the change of the user interest while new and dynamic items are presented. Due to the difficulty of finding suitable target systems for comparison. We only stopped at the level of model's performance where the comparison can be made. However, our experiments highlighted the feasibility and possibility of considering temporal dynamics and non-stationarity in user, item and interaction data when designing algorithms for recommendation systems. The findings advocate for a paradigm shift toward models that accommodate data dynamics and its evolving nature. The superior performance of the Non-stationary Transformer enhanced models suggests that such architectures could be easily build critical for recommendation systems.

Future research could explore the scalability of Non-stationary Transformer models in even larger datasets and their adaptability across different domains. Additionally, investigating the interpretability of these models could yield further insights into the nature of the complex patterns they learn, potentially guiding the design of even more effective recommendation systems.

VI. Conclusions

This study has presented an examination of integrating Non-stationary Transformer for recommendation systems with a hope that it can capture the change of user, item and interactions data that we believe are the fundamental reasons for user interests shifting when new items emerged. We tested the integration of Non-stationary Transformer with a commonly used BST model. Our primary test demonstrated considerable superiority over the baseline BST and other benchmark models. The results indicated that the integration of Non-stationary Transformer can capture the non-stationary nature of user, item and interaction data. Those temporal dynamics that traditional models often overlook. Despite its strengths, our study also revealed some potential issues such as the increase of the computational complexity of Non-stationary Transformer models. It is significantly higher than that of more simplistic models, which may present scalability challenges in larger or more dynamic environments. For the moment, our priority is to embed Non-stationary Transformer into more deep learning-based recommendation models and test them for recommendation tasks and prove the effectiveness of capturing user interests shifting.

Acknowledgment

This work is partially supported by the XJTLU HeXie Management Research Centre and XJTLU-REF-21-01-002 and XJTLU Key Program Special Fund (KSF-A-17).

References

[1] Guo, H., Tang, R., Ye, Y., Li, Z., He, X. "DeepFM: a factorization-machine based neural network for CTR prediction." In Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 1725–1731.

[2] Cheng et al. "Wide deep learning for recommender systems". In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 2016, pp. 7–10.

[3] Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., Zhu, X., Gai, K. "Deep interest evolution network for click-through rate prediction." In Proceedings of the AAAI Conference on Artificial Intelligence, Volume 33, Number 01, 2019, pp. 5941–5948.

[4] Chen, Q., Zhao, H., Li, W., Huang, P., Ou, W. "Behavior sequence transformer for e-commerce recommendation in Alibaba." In Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data, 2019, pp. 1–4.

[5] Gong, Y., Zhu, Y., Duan, L., Liu, Q., Guan, Z., Sun, F., Ou, W., Zhu, K.Q. "Exact-k recommendation via maximal clique optimization." In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 617–626.

[6] Van Hasselt, H., Guez, A., Silver, D. "Deep reinforcement learning with double q-learning." In Proceedings of the AAAI Conference on Artificial Intelligence, Volume 30, Number 1, 2016.

[7] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O. "Proximal policy optimization algorithms." arXiv:1707.06347, 2017. unpublished.

[8] Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D. Continuous control with deep reinforcement learning. arXiv:1509.02971, 2015. unpublished.

[9] Zhang,W., Du, T.,Wang, J. Deep Learning over Multi-field Categorical Data: –A Case Study on User Response Prediction. In Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38, Springer, 2016, pp. 45–57.

[10] Li, Z., Wu, S., Cui, Z., Zhang, X. "GraphFM: Graph factorization machines for feature interaction modeling". arXiv:2105.11866 2021. unpublished.

[11] Zhou, G. et al. "Deep interest network for click-through rate prediction". In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1059–1068.

[12] Sun, F. et al. "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer". In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 1441–1450.

[13] Ye, M et al "Adaptive dense-to-sparse paradigm for pruning online recommendation system with non-stationary data". arXiv:2010.08655 2020. unpublished.

[14] Huleihel, W., Pal, S., Shayevitz, O. "Learning user preferences in non-stationary environments". In Proceedings of the International Conference on Artificial Intelligence and Statistics, 2021, PMLR: pp. 1432–1440. 604

[15] Wu, Q., Iyer, N., Wang, H. Learning contextual bandits in a non-stationary environment. In Proceedings of The 41st International ACM SIGIR Conference on Research Development in Information Retrieval, 2018, pp. 495–504.

[16] Chandak, Y., Theocharous, G., Shankar, S., White, M., Mahadevan, S., Thomas, P. "Optimizing for the future in non-stationary MDPs". In

Proceedings of the International Conference on Machine Learning, 2020, PMLR: pp. 1414–1425.

[17] Liu, Y., Wu, H., Wang, J., Long, M. "Non-stationary transformers: Exploring the stationarity in time series forecasting". Advances in Neural Information Processing Systems 2022, 35, 9881–9893.

[18] Liu, Y.; Mikriukov, D.; Tjahyadi, O.C.; Li, G.; Payne, T.R.; Yue, Y.; Siddique, K.; Man, K.L. "Revolutionising Financial Portfolio Management: The Non-Stationary Transformer's Fusion of Macroeconomic Indicators and Sentiment Analysis in a Deep Reinforcement Learning Framework." Applied Sciences 2023, 14(1), 274.

[19] Liu, Y.; Man, K.L.; Li, G.; Payne, T.R.; Yue, Y. "Evaluating and Selecting Deep Reinforcement Learning Models for Optimal Dynamic Pricing: A Systematic Comparison of PPO, DDPG, and SAC." In Proceedings of the 2024 8th International Conference on Control Engineering and Artificial Intelligence, 2024; pp. 215–219.

[20] He, X., Chua, T.-S. "Neural factorization machines for sparse predictive analytics". In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp. 355–364.

[21] Wang, R., Fu, B., Fu, G., Wang, M. "Deep & Cross Network for Ad Click Predictions". In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2017, pp. 1–7.