جامعة بيرزيت

BIRZEIT UNIVERSITY

Faculty of Engineering & Technology

Department of Electrical & Computer Engineering

Accelerated DLRM-based E-commerce Recommendation System

Prepared By:

Ibraheem Alyan 1201180

Mohammad Abu-Shelbaia 1200198

Nidal Zabade 1200153

Supervised By:

Dr. Ahmed Shawahna

A graduation project report submitted to Electrical and Computer Engineering Department as part of B.Sc. in Computer Engineering degree requirements fulfilments.

Birzeit

December 20, 2023

Abstract

The project aims to design and develop a cutting-edge accelerated e-commerce deep learning recommendation system. The goal is to deliver a production-ready solution, with automated data injection and training pipelines, and a simple RESTful application programming interface (API) as a final interface. The project will have a special focus on scalability and performance.

This report discusses different types of recommendation systems and then compares them to deep learning recommendation model (DLRM) based systems in terms of different metrics and features, such as their accuracy, scalability, and performance. Furthermore, it compares existing solutions and their aspects, in addition to discussing possible technolgies and architectures to use in the system.

المستخلص

يهدف المشروع إلى تصميم وتطوير نظام توصية لمنصات التجارة الإلكترونية باستعمال التعلم الآلي العميق. الهدف النهائي هو تقديم حلول صالحة لبيئة التشغيل، تتم فيها أتمتة عمليات إدخال البيانات و تدريب نماذج التعلم اللآلي وواجهة برمجة تطبيقات RESTful API كواجهة نهائية. سيركز المشروع بشكل خاص على قابلية التوسع والأداء.

يناقش هذا التقرير أنواعًا مختلفة من أنظمة التوصيات ويقارنها بالأنظمة القائمة على نماذج التوصية بالتعلم اللآلي العميق (DLRM) من حيث المقاييس والميزات المختلفة. علاوة على ذلك، فهو يقارن الحلول المتوفرة حالياً و مزاياها، كما ويناقش التقنيات والبنى الممكن استعمالها في تطوير النظام.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Contents

1.1 Motivation

The exponential growth of e-commerce has introduced an enormous amount of choice, where consumers face overwhelming product options. To address this challenge, personalized recommendation systems [1] have become essential for enhancing the shopping experience and increasing the conversion rate for any e-commerce platform.

In contrast to conventional collaborative filtering [2], content-based [3], or popularity-based recommendation systems, our AI-based solution offers distinct advantages. Firstly, AI makes it possible to provide per-user personalized recommendations, which are tailored to their unique preferences and behaviors, enhancing user engagement and satisfaction. AI systems can also intelligently recommend comparable or complementary products or content to increase revenue through cross-selling. Furthermore, AI takes into account the impressions and interactions of users with items, allowing for a more dynamic and accurate understanding of user preferences. Using AI leads to improved recommendation accuracy and relevancy, leading to increased conversion rates and business growth.

Statistics from different use cases of recommendation systems:

- An intelligent recommender system delivers on average a 22.66% lift in conversions rates [4] for web products.
- IKEA experienced a 30% increase in click-through rate, 2% surge in average order value [5] using Google Recommendations AI [6].
- Lotte Mart experienced a 1.7x increase in new product purchases [7] using Amazon Personalize [8].

In summary, the project's motivation is elevating the e-commerce experience, driving business success, and harnessing cutting-edge AI technologies to create a recommendation system that is both high-performing and scalable.

1.2 Problem Statement

The process of building the solution is mainly two parts:

- First, designing a personalized recommendation system that covers what traditional collaborative filtering, content-based, or popularity-based systems cannot achieve.
- Second, deploying and automating the solution, including, data cleaning, data storage, and model deployment processes, and ensuring a production-ready and scalable system.

1.3 Report Organization

The rest of the report is organized as follows. Chapter 2 provides background information about recommendation systems. Furthermore, the chapter also discusses the different types of recommendation systems. Chapter 3 presents a literature review of the existing recommendation systems and libraries. In Chapter 4, the proposed framework is discussed and describe its main functional components together with their implementation details. Chapter 5 will demonstrate the effectiveness of the proposed framework through extensive experiments on state-of-the-art architectures and benchmark datasets. Finally, we will conclude and describe the future directions of this report in Chapter 6.

Chapter 2

Background

Contents

2.1 Recommendation Systems

A recommendation system is an artificial intelligence or AI algorithm, usually associated with machine learning, that uses Big Data to suggest or recommend additional products to consumers. These can be based on various criteria, including past purchases, search history, demographic information, and other factors. [2]

Recommender systems undergo training to understand the preferences, earlier decisions, and attributes of the user and products using their past interactions which include impressions, clicks, purchases, and ratings. Recommender systems are usually used by content and product providers to suggest items to users that they may like based on their profiles and preferences.

2.2 Types of Recommendation Systems

2.2.1 Context Filtering

Context Filtering is a technique that uses the contextual information of the user by framing the recommendation problem as a contextual multi-armed bandit problem and using the contextual information to learn the user's preferences.

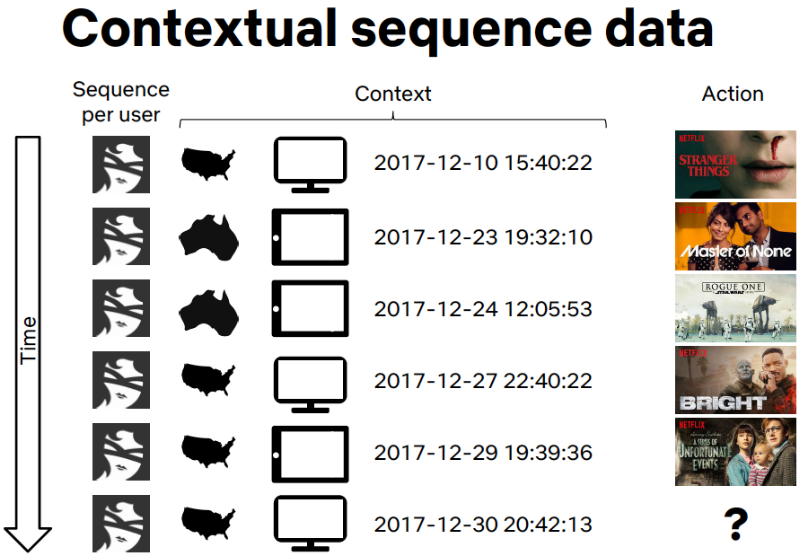


Figure 2.1: Context Filtering Diagram [2]

2.2.2 Variational Autoencoder for Collaborative Filtering

This model consists of two parts: an encoder and a decoder. The encoder takes the user's preferences as input and encodes them into a latent space. The decoder takes the latent space as input and decodes it into the item's features.

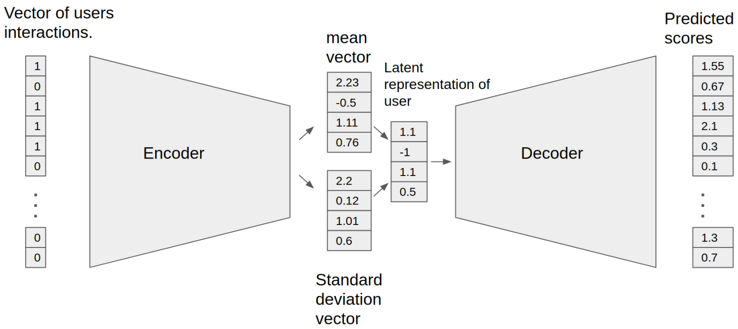


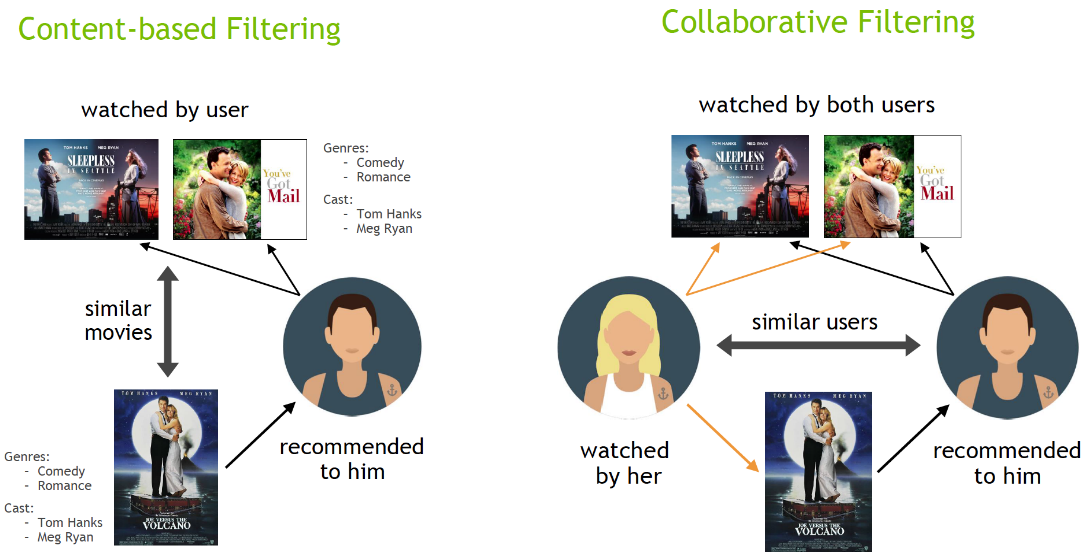
Figure 2.2: Variational Autoencoder for Collaborative Filtering Structure [2]

2.2.3 Collaborative Filtering

A technique that filters out products that a customer might like based on reactions by similar users. It functions by clustering customers into smaller sets with similar interests. Then it uses the items they show interest in to create a ranked suggestions list. The idea behind this technique is that individuals who have previously agreed will continue to do so in the future.

2.2.4 Content Filtering

A technique that uses the features of items a user has interacted with to recommend more items with similar features. This technique is based on the idea that if a customer shows interest in a particular product, he will also be interested in a similar product.



(a) Content Filtering

(b) Collaborative Filtering Diagram

Figure 2.3: Nvidia Glossary Diagram [2]

2.2.5 Hybrid Recommendation Systems

Combine the advantages of multiple types of recommendation algorithms to create a more comprehensive recommending system.

2.2.6 Neural Collaborative Filtering

NCF is a technique that uses neural networks to learn the customer's preferences and recommend items. It uses one neural network that learns the customer's preferences and another neural network that learns the item's features. The two networks are then combined to create a recommendation.

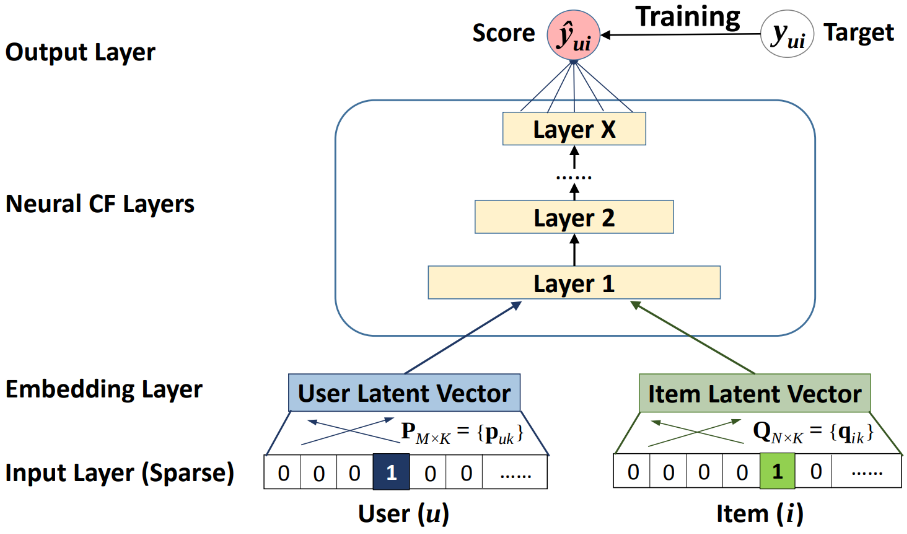


Figure 2.4: Neural Collaborative Filtering [2]

2.2.7 Contextual Sequence Learning

Contextual Sequence Learning is a technique that takes into account the context and sequence of the user's action, it usually uses a recurrent neural network (RNN). An example use case is session-based recommendations, RNNs predict the next items based on user event sequences, mirroring word embedding in NLP.

2.2.8 Wide And Deep

Wide & Deep is a technique that uses a wide neural network to learn the preferences of the customer and utilizes another deep neural network to learn the products's features. The wide model is a generalized linear model (GLM) and the deep model is built from a dense neural network (DNN).

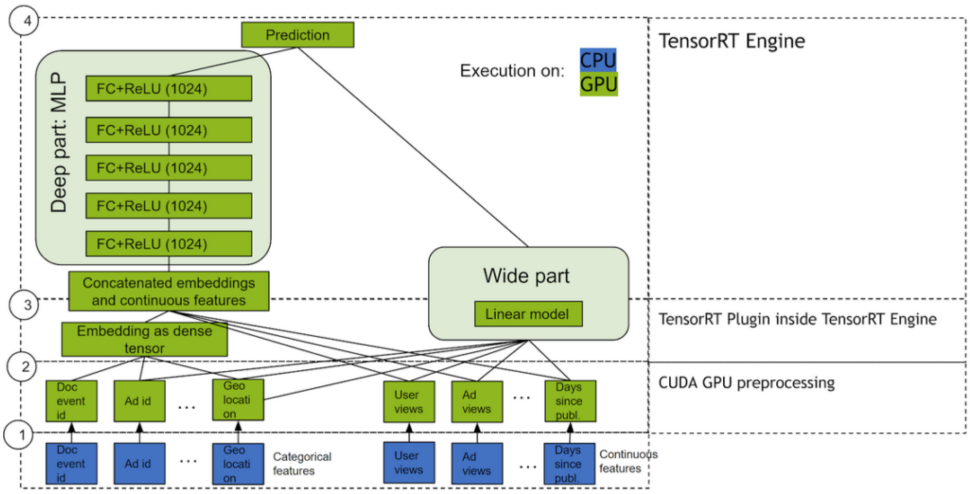


Figure 2.5: Wide Deep Structure [2]

2.2.9 DLRM

Deep Learning Recommendation Model (DLRM) is a technique that uses a deep neural network to handle categorical and numerical features. Each categorical feature is represented as a one-hot vector and each numerical feature is represented as a dense vector, both fed into multi-layer perceptron (MLP) layers. The output of the MLP layers is then fed into a dot product layer to compute the inner product of the feature vectors. The output of the dot product layer is then fed into a sigmoid layer to compute the probability of the user liking the item.

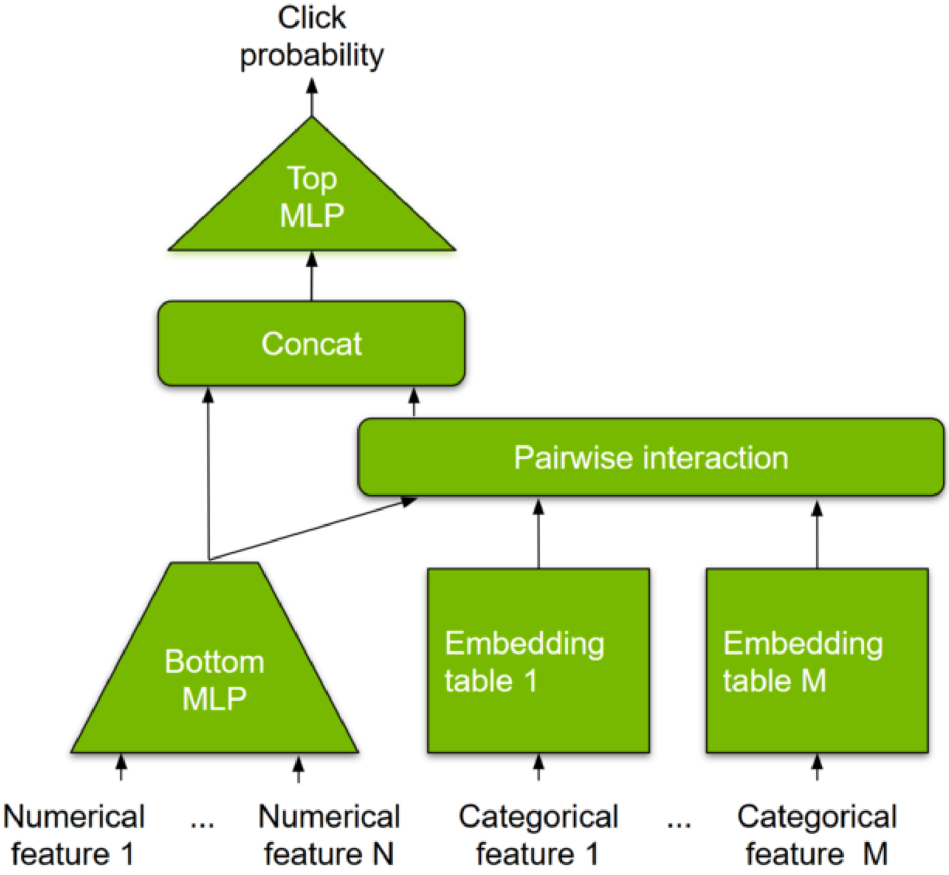


Figure 2.6: DLRM Structure [2]

Chapter 3

Requirements And Literature Review

Contents

3.1 Functional Requirements

The system should provide a RESTful API as the final interface to be used by the front-end application. The API provides endpoints that allow inserting customers, products, and interactions. In addition to endpoints for retrieving the recommendations for a given customer.

3.2 System Requirements

In order for the system to be useful it has to meet the following specifications:

3.2.1 Scalability

Scalability implies that it has to be cloud-native, the inference system should apply proper load balancing across multi-node, multi-model deployments.

3.2.2 Real-time Predictions

To be usable in any website or application, the system should be able to provide real-time predictions, and suggestions, with a few milliseconds latency.

To fulfill this requirement, trained models should run on optimized inference servers or services, the suggested deployment plan is to use **Nvidia Triton** [1]. inference server [9], integrated with **Amazon SageMaker** model deployment [10] as infrastructure.

3.2.3 Near Real-time Training

This implies continuous training and deployment of the model which requires the automation of training and deployment.

3.2.4 Elasticity And Optimization

Elasticity is vital for keeping up with traffic spikes and declines while optimizing infrastructure costs. To achieve this, the system should be able to scale up and down based on the traffic and load.

3.2.5 Security

Like any other system, the system has to be immune to security threats by implementing best practices at every level in the deployment and design.

For example, rate-limiting requests to interaction injection endpoints, using attestation when possible, and limiting access to user and product create, read, update, and delete (CRUD) operations.

3.3 Related Work

There are many open-source and paid solutions that provide recommendation systems and libraries, this section discusses some of them.

3.3.1 LightFM

A Python library that enables the classic matrix factorization techniques to include metadata about both items and users, incorporating both content and collaborative information into the recommendation process (hybrid)[11].

Its approach is described with more depth in the LightFM paper [12].

3.3.2 Rexy

Rexy [13] is a Python library that provides a general-purpose recommendation system framework. It is flexible and can be adapted to a variety of data schemas [13].

[1] Nvidia Triton Inference Server, part of the Nvidia AI platform and available with Nvidia AI Enterprise, is open-source software that standardizes AI model deployment and execution across every workload [9]

3.3.3 Gorse

Gorse [14] is an open-source recommender system engine implemented in Go that provides a scalable and flexible recommendation system framework. It supports a variety of algorithms, including collaborative filtering, content-based filtering, and deep learning [13].

3.3.4 AWS Personalize

According to Amazon Web Services (AWS), developers can utilize Amazon Personalize [8] to rapidly create and implement customized recommendations and sophisticated user segmentation on a large scale through machine learning (ML). This service is adaptable to individual requirements, enabling the delivery of personalized customer experiences at the optimal moment and location[2].

3.3.5 Google Recommendations AI

As outlined by Google on their cloud marketplace[3], Recommendations AI [6] allows individuals to develop a fully personalized recommendation system utilizing state-of-the-art deep learning ML models, without requiring expertise in ML or recommendation systems.

3.3.6 Nvidia Merlin

Nidia defines Merlin [16] as "an open source library providing end-to-end GPU-accelerated recommender systems, from feature engineering and preprocessing to training deep learning models and running inference in production." [4].

The frameworks, discussed in more depth later, provide many components including:

- Merlin Models [18]
- Merlin NVTabular [19]
- Merlin HugeCTR [20]
- Merlin Transformers4Rec [21]
- Merlin SOK (SparseOperationsKit)
- Merlin Distributed Embeddings (DE)
- Merlin Systems [22]

Making it a very customizable and extensible solution.

[2] AWS description of the service [8]

[3] Google Cloud Marketplace Recommendations AI Page[15]

[4] Nvidia Merlin Repository [17]

Table 3.1: Comparison of Recommendation Solutions

System	**LightFM**
License	Apache 2.0
Algorithm Type	Matrix Factorization
Hardware Utilization	CPU
Deployment Readiness	Library (Additional Components Needed)
Notes	-
System	**Rexy**
License	MIT
Algorithm Type	Matrix Factorization
Hardware Utilization	CPU
Deployment Readiness	Library (Additional Components Needed)
Notes	-
System	**Gorse**
License	Apache 2.0
Algorithm Type	Matrix Factorization
Hardware Utilization	CPU
Deployment Readiness	Single-node-learning multi-node-inference cluster
Notes	Unreliable and has many bugs
System	**AWS Personalize**
License	Proprietary
Algorithm Type	DLRM
Hardware Utilization	-
Deployment Readiness	A lot of customization required
Notes	High customization, predictions, and training fees
System	**Google Recommendations AI**
License	Proprietary
Algorithm Type	DLRM
Hardware Utilization	-
Deployment Readiness	End-to-End service
Notes	High predictions, and training fees
System	**Nvidia Merlin**
License	Apache 2.0
Algorithm Type	Multiple Options
Hardware Utilization	Optimized for Nvidia GPUs
Deployment Readiness	Recommendation pipelines components
Notes	Very customizable

Bibliography

[1] Raghavendra C K, Srikantaiah K.C, Venugopal K. R, "Personalized Recommendation Systems (PRES): A Comprehensive Study and Research Issues," *International Journal of Modern Education and Computer Science (IJMECS)*, vol. 10, no. 10, pp. 11–21, 2018. DOI: 10.5815/ijmecs.2018.10.02.

[2] Nvidia Glossary, "Recommendation System." `https://www.nvidia.com/en-us/glossary/data-science/recommendation-system/`. Accessed: 2023-12-19.

[3] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The adaptive web: methods and strategies of web personalization*, pp. 325–341, Springer, 2007.

[4] Salesforce Marketing Cloud, "Predictive intelligence benchmark report." `https://brandcdn.exacttarget.com/sites/exacttarget/files/deliverables/etmc-predictiveintelligencebenchmarkreport.pdf`, 2014.

[5] Google Cloud Summit, "Ikea's approach to building a powerful recommendations engine." by Google Cloud Events `https://www.youtube.com/watch?v=PyjC0wRRtBg`, 2021.

[6] Google Cloud, "Google Recommendations AI." `https://cloud.google.com/recommendations`. Accessed: 2023-11-9.

[7] Sungoh Park and Kyoungtae Hwang, "Increasing customer engagement and loyalty with personalized coupon recommendations using Amazon Personalize," *AWS Machine Learning Blog*, 2020.

[8] AWS, "AWS Personalize." `https://aws.amazon.com/personalize/`, accessed: 2023-10-4.

[9] Nvidia, "Nvidia Triton Inference Server." `https://developer.nvidia.com/triton-inference-server`, accessed: 2023-10-4.

[10] AWS, "Amazon SageMaker Model Deployment." `https://aws.amazon.com/sagemaker/deploy/`, accessed: 2023-10-4.

[11] "LightFM." `https://making.lyst.com/lightfm/docs/home.html`, accessed: 2023-10-8.

[12] M. Kula, "Metadata embeddings for user and item cold-start recommendations," 2015.

[13] "Rexy." `https://github.com/kasraavand/Rexy`, accessed: 2023-10-8.

[14] "Gorse." https://github.com/gorse-io/gorse, accessed: 2023-10-8.

[15] "Google Cloud Marketplace - Recommendation AI." https://console.cloud.google.com/marketplace/product/google/recommendationengine.googleapis.com, accessed: 2023-12-20.

[16] Nvidia, "Merlin." https://developer.nvidia.com/merlin, accessed: 2023-10-6.

[17] Nvidia, "Merlin Repository." https://github.com/NVIDIA-Merlin/Merlin, accessed: 2023-10-6.

[18] Nvidia, "Merlin Models Repository." https://github.com/NVIDIA-Merlin/models, accessed: 2023-10-6.

[19] Nvidia, "Merlin NVTabular." https://developer.nvidia.com/nvidia-merlin/nvtabular, accessed: 2023-10-6.

[20] Nvidia, "Merlin HugeCTR." https://developer.nvidia.com/nvidia-merlin/hugectr, accessed: 2023-10-6.

[21] Nvidia, "Merlin Transformers4Rec." https://github.com/NVIDIA-Merlin/Transformers4Rec, accessed: 2023-10-6.

[22] Nvidia, "Merlin Systems Repository." https://github.com/NVIDIA-Merlin/systems, accessed: 2023-10-6.