IJCI V11-1(2024) 77–86
International Journal of Computers and Information
(IJCI)
Available online at *https://ijci.journals.ekb.eg/*

Long Short-Term Memory and Gated Recurrent Unit for Automated Deep Learning Prediction

Esam Elgohary[a], Mohamed Galal[b], Mostafa Aref [c], Mona Gharib*[,d]
[a]*Associate Professor of Information systems, Institute of National Planning, Cairo, Egypt*
[b]*Predictive Analytics department, National Bank of Egypt, Cairo, Egypt*
[c]*Computer Science Department, Faculty of Computer Science and Information Systems, Ain Shams University, Cairo, Egypt*
[d]*Department of Mathematics, Faculty of Science, Zagazig University, 44519 Zagazig, Egypt*

**Corresponding author: Mona Gharib*
Email: monagharib@yahoo.com, monagharib@zu.edu.eg

Abstract

Recommender systems are nowadays an effective strategy to overcome the exponential growth of online products and services. The recommender systems assist customers to overcome over-choice-related problems and improve their satisfaction. This research presents an automated deep learning-based service for a personalized recommender system in the retail industry. This service automates deep learning data modeling processes regardless of the business case. It predicts the next best offer to a given customer based on the provided customers' behavior. Both Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms are used in parallel to train over the clients' data until the service chooses the best model performance to deploy. Finally, two case studies are presented to show the service performance in two different business cases. The first case study achieves a micro–Area Under Curve (AUC) score of 0.84 on a supermarket dataset, while the second one achieves 0.95 micro-AUC on the restaurant dataset.

Keywords: : Predictive models; Deep learning; Automated Prediction; Recommender systems.

1. Introduction

Global expansion of e-commerce led to a wide selection of products and services. Here comes the Recommender systems (RS) role that analyzes client behavior and recommends products or services. RS increases cross-selling and improves customer loyalty [1]. On the other hand, traditional RS systems lack accuracy and scalability issues where each business case needs a personalized prediction model.

Recently, Deep learning-based recommender systems (DLRS) applied for more complicated tasks to improve the quality of the recommendations. Recurrent neural network (RNNs) is the best solution for customer recommendation systems since their prediction is based upon the time series method and the past user actions. A Long Short-Term Memory (LSTM) unit and a Gated Recurrent Unit (GRU) are two types of RNNs that can predict the next best offer based on previous customers' purchases.

This research comes as part of Customer Loyalty Intelligent Personalization (CLIP). CLIP is a personalized advisory system, which applies the most recent machine learning techniques. The paper proposes an automatic evaluation that focuses on sequence modeling built over business datasets. It will use both LSTM and GRU to develop the next best offer RS.This system is configured to be easily trained and deployed regardless of the business case. This paper sections are: Section1 is the introduction section that gives a short brief on the problem. Section 2 outlined a related work. Section 3 illustrated background, Section 4 proposes the enhanced deep-learning approach in recommender systems. Section 5 shows the applied case studies that include two real business cases' performance on the proposed recommender system, and finally, Section 6 is the conclusion.

2. Related work

Different researches have discussed the personalized RS service which assists customers to find preferred products to their preferences in E-commerce. On the other hand, Recommender systems have been applied in many fields like E-commerce, health, E-learning, music, social networks, etc. This section briefly explores recent applied papers and algorithms in recommender systems.

Initially, a survey [2] provided a comprehensive review of algorithms and techniques on recommender systems. It introduced a comparative study between Content-Based Filtering (CBF), Collaborative Filtering (CF), Demographic-Based Filtering (DBF), hybrid filtering, and Knowledge-Based Filtering (KBF) algorithms. The Content Filtering (CF) algorithms used the user-item interactions [3-5]. This approach had low scalability, data sparsity, data acquisition, and high computational time problems. Its strengths were better relevance, recommendation quality result, and considering the user experiences [2]. The CBF approach employed the features of users and items [6-8]. It improved the quality of recommendations. CBF also solved the cold-start problem and the complex e-commerce personalized recommendations. However, it needs to consider more parameters to increase processing resources, storage need, and response time [2]. KBF approach required the outward knowledge foundations, user requirements, and restrictions [9-11]. KBF problems were like a cold-start problem, early-rater problem, and fake neighborhoods. Although its advantages brought customer satisfaction and, in some cases, boosted sales and enhanced the recommendation quality. Low scalability and high operation cost problems still showed up [2]. Demographic RSs used demographic information of the user based on special mapping for ratings or purchasing [12-14]. The significant problem with DBF methods was that some individuals with similar demographic features might have different interests [2]. A hybrid approach [15,16] is the combination of the pre-mentioned methods. Its results are more accurate. However, it required more processing times and higher computational costs [2]. [17] presented a hybrid approach. It combined the simplicity of collaborative filtering with ontology-based recommenders. Its evaluation demonstrated higher quality than collaborative filtering. That article chose the k-nearest neighbors of the active user, to which KNN would be applied. However, it satisfied the active user's preferences.

Recently, deep learning has stellar performance and attractive learning feature representations in various research fields such as computer vision and natural language processing. Deep learning has a pervasive influence when applied to information retrieval and recommender systems research. Deep learning-based recommender systems are presented in different architectures multi-layer percepton, Autoencoder, Convolutional neural network, Recurrent Neural Network, and Attention Models [18]. Their strengths are implied in Nonlinear transformation, representation learning, sequence modeling, and flexibility. Their limitation is summarized in Interpretability, sufficient data, and extensive hyper-parameters tuning.

For example, [19] proposed a personalized Fashion Recommender system. It generated users' recommendations based on an input image. This research used neural networks to generate fashion recommendations from the DeepFashion dataset. It could be described as data-driven, visually related, and simple effective recommendation systems. The proposed approach used a two-stage phase. Initially, it extracted the features of the image using CNN classifier. Then, it allowed the customers to upload any random fashion image from any E-commerce website and later generated similar ones. It promoted greater recommendation accuracy and improved the overall experience of fashion exploration for direct and indirect consumers alike.

Another example [20] presented the design of iPrescribe. It used an ensemble of deep learning and machine learning algorithms for prediction. It compared its performance using different real-time streaming technology stacks. This study achieved a 90th percentile recommendation latency of 38 milliseconds. The prediction model had been built as an ensemble of XGBoost and a deep learning LSTM network. Their models' accuracy was AUC=0.67 and F-score=0.3843 for these two e-commerce data sets.

Briefly, related research showed a lack of accuracy with non-domain specific models and the efficiency of deep learning methods in solving recommender systems problems. Consequently, this research proposed an enhanced deep learning-based recommender system service. It automates data modeling processes based on the client's behavior. Then, it can be dynamically adapted to different e-commerce cases without data scientist interference. It builds two recurrent networks LSTM and GRU. The service auto-evaluates the two networks. Lastly, the best performance model is deployed on the client's side. The proposed system will be detailed in the following sections with two practical case studies.

3. Background

This research trains and evaluates two recurrent neural networks (RNN); a Long Short-Term Memory (LSTM) unit, and a Gated Recurrent Unit (GRU). Why RNN? Because RNN has two advantages. First, significant features are maintained, because each unit can remember a specific feature in the input stream for a long series of steps. Second, it effectively creates shortcut paths that bypass multiple temporal steps. These shortcuts allow the error to be back-propagated easily without too quickly vanishing, thus reducing the difficulty due to vanishing gradients. As a result, these algorithms don't fade information. They keep the relevant information and pass it down to the next step. Thus, they avoid vanishing gradients problems. Though, if LSTM and GRU models are trained carefully, they perform exceptionally well in complex scenarios [21].

The LSTM (Long Short-Term Memory) has three gates input, output, and forget gate. The input gate guides how much to keep for cell state, the forget gate coordinates how much to eliminate from the current cell, and the output gate manages how much should be exposed to the following layers. In the LSTM unit, the output gate controls the amount of memory content used by other units in the network. LSTM may lead to better results with gigantic data. While the GRU (Gated Recurrent Unit) has two gates reset and update. The reset gate is between the previous and the next activations to forget the previous state. The update gate adjusts how much of the activation to use in updating the cell state. GRU has two values at the output (output and hidden). GRU is less complex and faster to compute than LSTM. It has fewer parameters and thus may train a bit faster or need fewer data to generalize. GRUs exposes its full content without any control. The GRU does not need a memory to control the flow of information like the LSTM. It can directly make use of the all-hidden states without any control [22].Consequently, both GRU and LSTM avoid the exploding and vanishing gradient problems. They both keep long-term dependencies effectively while learning. The GRU uses less training parameters and less memory, while LSTM is more accurate on a larger dataset. This research evaluation demonstrated the superiority of the gated units; both the LSTM unit and GRU. However, there is no concrete conclusion on which of the two gating units was better.

4. The Enhanced Deep Learning Based Evaluation Approach

Customers' data represent their behavior towards products or services. The sequence of customers' purchases shows their behavior and interests. To predict the next best offer for each customer, a complex machine learning model is the solution especially sequence modeling algorithms. Therefore, the built model could have the ability to analyze and predict the best offer for the organization's customers.

Here is come deep-based learning role, where LSTM and GRU are applied. Fig. 1 displays the data flow of the proposed approach's essential processes. This workflow is repeated for each business case because the prediction model changes based on customers' behavior. Firstly, data is cleansed, prepared, and divided into three sets: train, validate, and test. The recurrent networks' hyper-parameters are initialized then two recurrent models are built independently. Later, the training and validation datasets are used to train and validate these built models until the best-fitted model is saved. This process is essential to train the model on the customers' behavior since it could vary from one case to another. Lastly, the test datasets evaluate the saved model to save micro-F1 performance for a given business case with unseen data. The final deployed model could recommend the most preferred categories for the business customers.

4.1. Data Preparation and Splitting

This research proposed an automated deep learning-based model. It focused on learning customer behavior to predict more than one preferred category of products. It begins to capture customers' data into the data preparation pipeline. This pipeline includes data cleansing, data preprocessing, and feature selection called data preparation. This pipeline should handle two crucial points: skewed data and one visit customer manipulation. The skewed data includes removing samples of any category or class less than 2% because classification problems require a reasonable number of cases in each category. Manipulation for one visit customer handles customers who only came one visit, so there is no sequence for that customer to predict based on it. Therefore, our model labels these visits. After that, the cleaned data is divided into three basic sets: training, validation, and testing. The best splitting percentage on customer data is 60 for training, 20 for validation, and 20 for testing. The following subsections illustrate the building, training, and evaluation processes in detail.

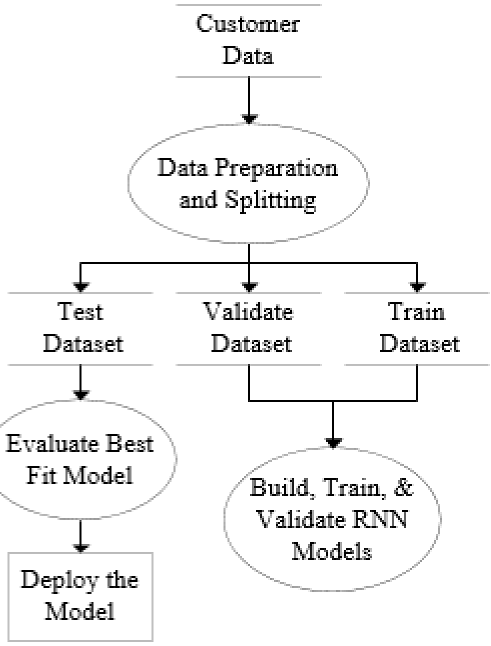

Fig.1. The proposed RS approach data flow

4.2. Build, Train, Validate, and Evaluate LSTM and GRU Models

The proposed architecture of both LSTM and GRU consists of three successive layers of the recurrent unit, followed by two dense layers, then the decision layer as shown in Fig. 2. Both LSTM and GRU include various hyper parameters. The effective hyper-parameters of numbers of recurrent neurons, batch size, and dense size are adapted on client behavior. The number of recurrent neurons and the dense size i or j in Figure 2 have a range of values [16, 32, 64, or 128], and the batch size has that range [10, 50, 100, 150, or 200]. The decision layer size varies according to the number of product categories N of the customer data. These hyper-parameters can form many combinations with variant values. These values will affect differently on the built model performance.

This paper's main contribution is the automation of data modeling processes to facilitate building and deploying prediction models. Previously, there were two built recurrent networks: LSTM and GRU. Their hyperparameter values are modified on several trials to decide the best achievement on the training data. The validation data is used to nominate the best fit model. Fig. 3 shows each trial flow of processes during the training and validation of RNN models. At each trial, hyper-parameter values are assigned to adjust each recurrent neural network. Then, the training dataset trained the two recurrent networks. The models' performance is examined using validating dataset and then recorded for each trial and only the best-fitted model is saved. Eventually, the best hyper-parameter combination is noticed when training and validation data have neither over-fitting nor under-fitting behavior and with steady error loss and accuracy score.

Evaluation is applied to the best-fitted model from the pre-mentioned training and validation phase using the testing dataset as shown in Fig. 4. The micro-F1 score [23, 24] is calculated and saved to record the model's performance over unseen data. The developed recommender system could predict more than one category or offer for one customer. Lastly, the best RS prediction model is chosen for deployment on the customer side.

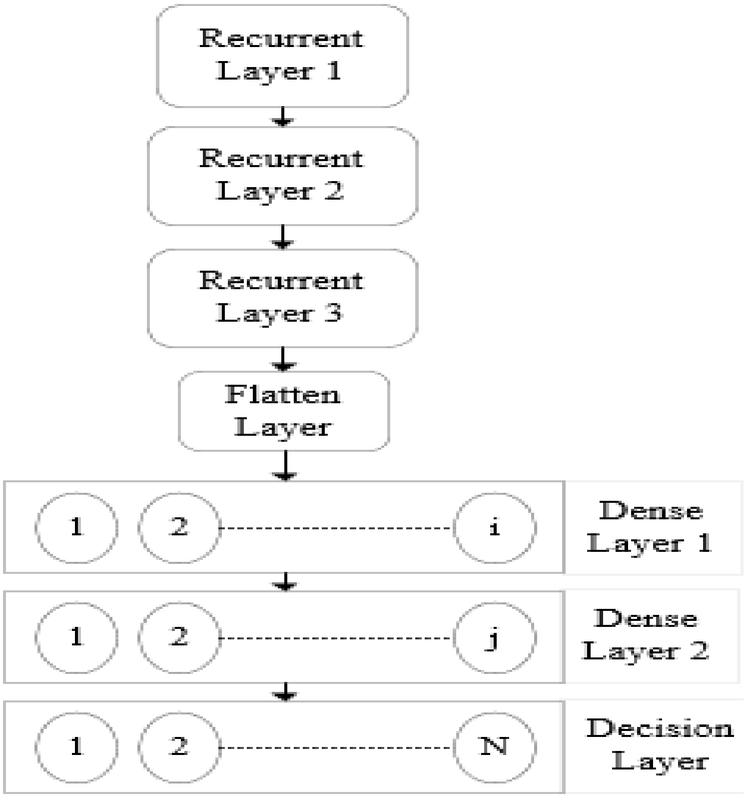

Fig. 2. The proposed RNNs architecture

5. Case studies I and II

There are two case studies applied. They show how the proposed recommender system performed efficiently on various business datasets to predict the next best categories. As previously mentioned, both LSTM and GRU networks trained on customer purchases sequences to recommend the next best categories or classes. The first case study works on a supermarket. The second one uses a restaurant dataset. Each dataset was initially preprocessed and cleaned, then applied to two parallel LSTM and GRU networks that are automatically trained and evaluated on the given customer purchases. In the end, the highest performance model of these two networks is chosen for deployment. The highest performance model is evaluated using loss, micro-F1 score, and AUC [23, 24] metrics. The micro-F1 is preferred in evaluation since the customer can have more than one offer or recommended category.

The first case study is on a supermarket dataset, and it consists of 1659 records. The supermarket recommender system is supposed to predict with four main categories. It has a 0.8 as minimum loss (see Fig. 5) and reaches the best micro-F1 score of 0.65 on the validation dataset as displayed in Fig. 6. The micro-AUC is 0.84 as shown in Fig. 7, while micro-AUC for each category ranges from 0.66 to 0.88. Best performance achieved by LSTM neural network with the recurrent neuron of 64 and large batch sizes 150 or 200. Both loss and micro-F1 score graphs show stable progress, which means the model is not poorly generalized or more tailored to predict the training data only.

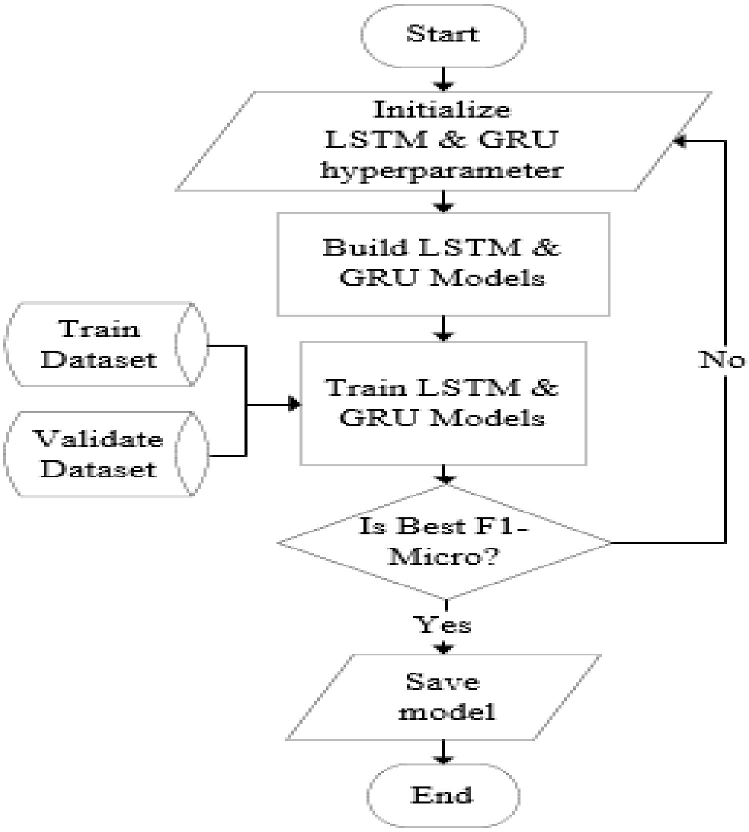

Fig. 3. Auto-training RS models flow chart

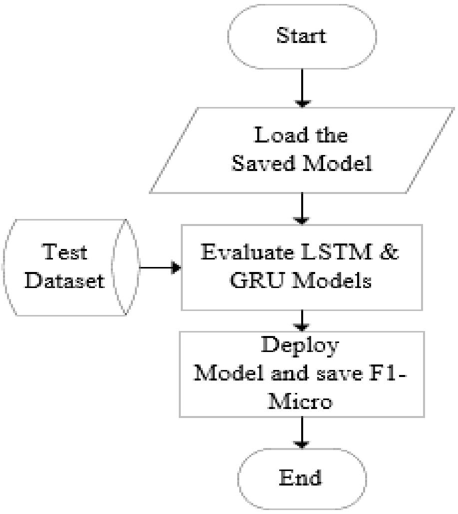

Fig. 4. Auto-evaluation RS models flow chart

The second case study uses a restaurant dataset and includes 668 records. Its recommender system is applied to predict within ten categories. The minimum loss achieved is 0.25 as in Fig. 8 through 300 epochs, and the steadiness micro-F1 score reaches 0.95 as shown in Fig. 9. The micro-AUC is 0.95 as shown in Fig. 10. Also, the LSTM recommender model shows the best performance with 64 recurrent neurons and large batch sizes. All previous graphs of loss and micro-F1 show steadiness and best-fitted learning during training and validation. Consequently, the choice of working with both LSTM and GRU boosted the system performance regardless business case. Table 1 compares the proposed system results with related background results. The proposed system surpasses other state-of-art approaches on reliability and automation with different business cases.

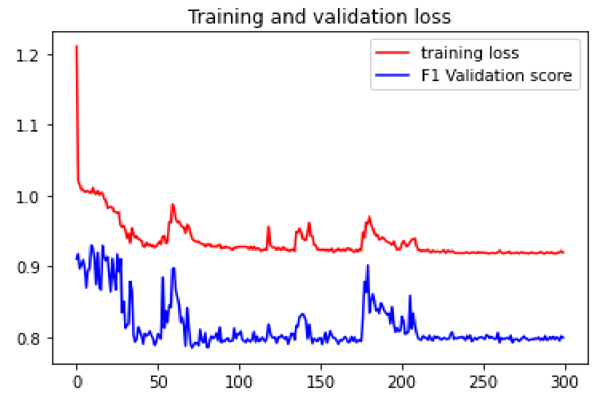

Fig. 5. Loss graph on supermarket dataset during 300 epochs (X-AXIS).

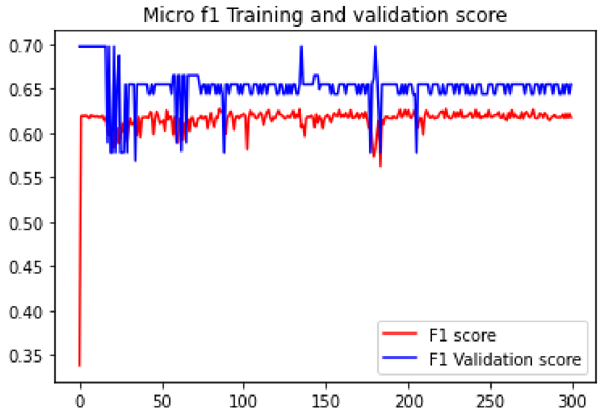

Fig. 6. Supermarket dataset micro-f1 score graphs in 300 epochs (X-AXIS).

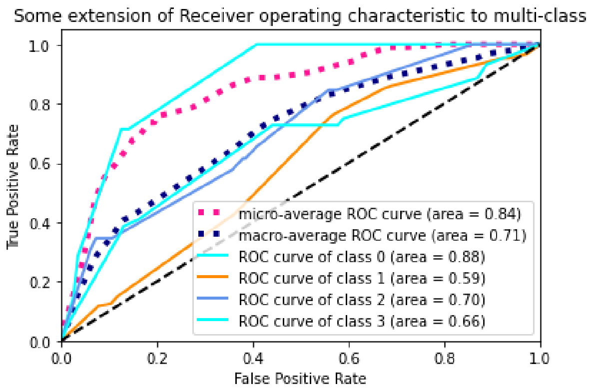

Fig. 7. Supermarket dataset AUC graphs to recommend between 4 categories

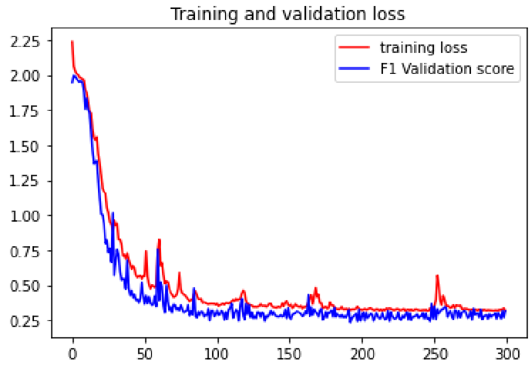

Fig. 8. Loss graph on restaurant dataset in 300 epochs (x-axis represents epochs numbers)

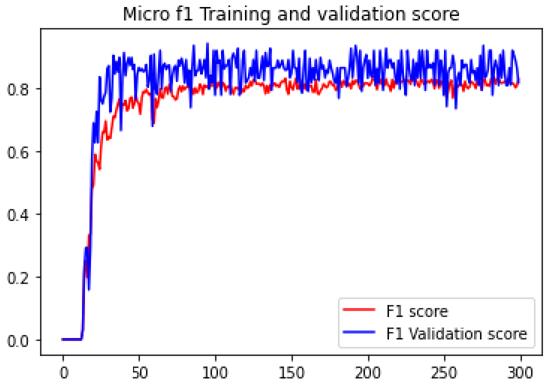

Fig. 9. Micro-F1 score graph on restaurant dataset in 300 epochs (X-Axis represents epochs number)

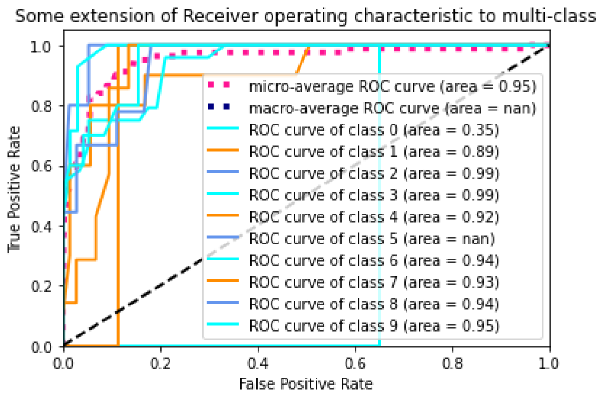

Fig. 10. Restaurant dataset AUC graphs to recommend between 10 categories

Table 1: *Comparing proposed work and state of art approaches*

Paper / Data	F-score	AUC
[20]	0.40	-
[21]	0.38	0.67
Supermarket	0.65 (micro)	0.84 (micro)
Restaurant	0.80 (micro)	0.95 (micro)

6. Conclusion

This paper came as part of implementing a customer loyalty intelligent personalized system called CLIP. The paper presented an enhancement on deep-learning-based recommender systems. They used two kinds of recurrent neural networks, which are LSTM and GRUs. These networks are accurate and effective due to the ability to keep learning important features even in a long data sequence. The proposed system enhanced the RS performance and outperformed other RS techniques by reliability and automobility in the retail business industry. Eventually, two case studies were shown to illustrate RS system performance on two clients' datasets to recommend more than one category for each customer. These datasets were from different retail cases, one for a supermarket and the other is for a restaurant. The supermarket dataset of 1659 records achieved a micro-F1 score of 0.65 to recommend between four categories, while the restaurant dataset of 668 records achieved a micro-F1 score of 0.80 to predict recommendation within ten categories. The future work implies enhancing running time and enhancing performance with increasing data scaling.

References

[1] J. B. Schafer, J. Konstan, and J. Riedi, Recommender systems in e- commerce, in Proc. 1st ACM Conf. Electron. Commerce (EC), 1999, pp. 158–166.
[2] P. M. Alamdari, N. J. Navimipour, M. Hosseinzadeh, A. A. Safaei , A. Darwesh: A Systematic Study on the Recommender Systems in the E-Commerce, IEEE Access Vol 8, pp. 115694 – 115716, 2020.
[3] J. B. Schafer, D. Frankowski, J. Herlocker, S. Sen: ‘‘Collaborative filtering recommender systems,’’ in The Adaptive Web: Methods and Strategies of Web Personalization, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Germany: Springer, 2007, pp. 291–324.
[4] P. H. Aditya, I. Budi, Q. Munajat: A comparative analysis of memory-based and model-based collaborative filtering on the implementation of recommender system for e-commerce in Indonesia: A case study PT X, in Proc. Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS), pp. 303–308, Oct. 2016.
[5] J. Cao, H. Hu, T. Luo, J. Wang, M. Huang, K. Wang, Z. Wu, X. Zhang: Distributed design and implementation of SVD++ algorithm for e-commerce personalized recommender system, in Proc. Nat. Conf. Embedded Syst. Technol. Springer, pp. 30–44, 2015.
[6] P. Lops, M. de Gemmis, G. Semeraro: ‘‘Content-based recommender systems: State of the art and trends,’’ in Recommender Systems Handbook, F. Ricci, L. Rokach, B. Shapira, P. B. Kantor, Eds. Boston, MA, USA: Springer, 2011, pp. 73–105.
[7] Z. Gao, Z. Li, K. Niu: Solutions for problems of existing e-commerce recommendation system, in Proc. IEEE Int. Conf. Consum. Electron. (ICCE-TW), pp. 1–2, May 2016.
[8] C. Chen, D. Wang, Y. Ding: User actions and timestamp based personalized recommendation for e-commerce system, in Proc. IEEE Int. Conf. Comput. Inf. Technol. (CIT), pp. 183–189, Dec. 2016.
[9] S. Trewin: Knowledge-based recommender systems, Encyclopedia Library Inf. Sci., 69(32):180-2000.
[10] Y. Luo: An analysis of Web mining-based recommender systems for e- commerce, in Proc. 2nd Int. Conf. Comput. Appl. Syst. Model. Paris, France: Atlantis Press, pp. 167–170, 2012.
[11] W. Xiaosen: The design and realization of personalized e-commerce recommendation system, in Proc. Int. Conf. Social Sci. Technol. Educ. Paris, France: Atlantis Press, pp. 292–295, 2015.
[12] M. Y. H. Al-Shamri: User profiling approaches for demographic recommender systems, Knowl.-Based Syst., Vol. 100, pp. 175–187, May 2016.
[13] Y. Sung Cho, K. Ho Ryu: Implementation of personalized recommendation system using demographic data and RFM method in e-commerce, in Proc. 4th IEEE Int. Conf. Manage. Innov. Technol. (ICMIT), pp. 475–479, Sep. 2008.
[14] Y. Fan, Y. Shen, J. Mai: Study of the model of e-commerce personalized recommendation system based on data mining, in Proc. Int. Symp. Electron. Commerce Secur., pp. 647–651, 2008.
[15] V. N. Zakharov S. A. Philippov: Clustering of goods and user profiles for personalizing in E-commerce recommender systems based on real implicit data, in Data Analytics and Management in Data Intensive Domains. Cham, Switzerland: Springer, pp. 178–191, 2017.
[16] S. Saini, S. Saumya, J. P. Singh: ‘‘Sequential purchase recommendation system for E-commerce sites,’’ in Computer Information Systems and Industrial Management. Cham, Switzerland: Springer, pp. 366–375, 2017.
[17] M. Guia, R. Silva, J. Bernardino: A Hybrid Ontology-Based Recommendation System in e-Commerce, Algorithms, Vol. 12, No. 11:239, 2019.
[18] S. Zhang, L. Yao, A. Sun, Y. Tay: Deep Learning based Recommender System: A Survey and New Perspectives, ACM Computing Surveys, Vol. 52, No. 1, 2019
[19] I. Goodfellow, Y. Bengio, A. Courville: “Deep Learning”, MIT Press. 2016
[20] R. Singhal, G. Shroff, M. Kumar, S. Roy, S. Kadarkar, R. Virk, S. Verma, V. Tewari: Fast Online ’Next Best Offers’ using Deep Learning, COMAD-CoDS, 2019.
[21] R. Dey & F. Salem: Gate-variants of Gated Recurrent Unit (GRU) neural networks. 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), 2017, Doi:10.1109/MWSCAS.2017.8053243, 2017.

[22] J. Chung, C. Gulcehre, K. Cho, Y. Bengio: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, Computer Science, December 2014.
[23] J. A. Swets: ROC Analysis Applied to the Evaluation of Medical Imaging Techniques, Invest. Radiol., Vol. 14, No. 2, pp. 109-121, 1979.
[24] J. A. Hanley: Receiver Operating Characteristic (ROC) Methodology: The State-of-the-Art, Crit. Rev. Diagn. Imaging, Vol. 29, No. 3, pp. 307-335, 1989.