Received: December 7, 2024. Revised: February 1, 2025.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

ESReLU: A Dynamic Activation Function for Enhancing Deep Learning Performance in Recommendations

Syed Irteza Hussain Jafri[1,2]* **Ahmed Khalaf Zager Al saedi[3]** **Abubakar Elsafi[4]**
Ghada Ahmed Abdelguiom[5] **Rozaida Ghazali[1]** **Irfan Javid[2]**

[1]*Faculty of Computer Science and Information Technology,*
Universiti Tun Hussein Onn Malaysia, Batu Pahat, Johor, Malaysia
[2]*Department of Computer Sciences & Information Technology, Faculty of Basic and Applied Sciences,*
University of Poonch, Rawalakot, Pakistan
[3]*Department of physics, Faculty of Science, Missan University, Iraq*
[4]*Department of Software Engineering, College of Computer Science and Engineering,*
University of Jeddah, Jeddah, Saudi Arabia
[5]*National Health Insurance Fund (NHIF), Sudan*
* Corresponding author's Email: irtezasyed@upr.edu.pk

Abstract: Deep learning has seen a tremendous increase in the recent past. A variety of deep learning architectures have been proposed for various kind of tasks. Activation functions (AF) play a critical role in a deep learning model with the ability to learn abstract properties through nonlinear transformations. This study proposes an adaptive AF, ExtendedSigmoidReLU (ESReLU), to improve model's performance. The proposed AF combines the strengths of Sigmoid and ReLU AFs, dynamically adjusting to varying inputs using a tunable parameter, that allows ESReLU to transition between traditional sigmoid and ReLU functions seamlessly. To ensure continuous and smooth behaviour during training, sigmoid introduces strong gradient flow, whereas ReLU introduces nonlinearity and overcomes sparsity. Experiments conducted on various datasets using a collaborative filtering model demonstrate that ESReLU outperforms traditional functions, achieving faster convergence and performance improvements of up to ~3% on accuracy. These results indicate that ESReLU has significant potential to enhance recommendation performance, especially in data-intensive applications.

Keywords: Activation functions, Collaborative filtering, Deep learning, Recommendation, Recommender systems.

1. Introduction

The amount of data generated and consumed online has increased significantly due to the widespread use of the internet and the explosive growth of the digital information. The data outburst has resulted in a large volumetric data posing numerous challenges to the users in getting the desired information in an effective and efficient way. Deep learning-based recommender systems (DLRS) provide efficient and effective recommendations to the users [1].

Neural networks are composed of numerous layers of neurons, or nodes, that are used to classify and predict data when the network receives input data. An input layer, an output layer, and one or more hidden layers are present. Every layer has nodes, and each node has a weight that is taken into account when information is processed from one layer to the next. Design and selection of a particular AF has great impact on the performance, convergence rate and the predictive ability of the deep learning model [2]. Current study presents a novel adaptive AF, ExtendedSigmoidReLU (ESReLU), to improve the performance and convergence of the DLRS.

The proposed ESReLU AF creates an adaptive and dynamic mechanism by combining the strengths of Sigmoid and ReLU AFs and optimizing gradient flow and learning behavior using configurable parameters. This adaptability enables ESReLU to address issues such as vanishing gradients and dead neurons more effectively than typical AFs. Unlike static activations, ESReLU adjusts dynamically to changing input distributions, resulting in faster convergence and greater accuracy. Experimental findings show that ESReLU routinely outperforms popular AFs such as ReLU, Mish, and Swish variants, attaining up to 3% greater accuracy on datasets such as MovieLens 25M while being computationally efficient.

The rest of the article is organized as follows. Related work is covered in section 2. Section 3 covers the proposed activation function. Experimental results and discussion is covered in section 4, and finally section 5 covers the conclusion and future directions.

2. Related work

A neural network should be able to execute increasingly complex tasks, such as modelling complex data types including photos, videos, audio, voice, text, etc., in addition to learning and computing linear functions. For neural networks to exhibit non-linearity they need activation functions (AFs) [3, 23]. Our ultimate goal is to extract knowledge, thus, to do this, we employ artificial neural network techniques and AFs to make sense of complex, high dimensional, and nonlinear datasets where the model has numerous hidden layers. These AFs are responsible for the decision whether a neuron in the network can activate or not. Over time, different AFs have been suggested in the literature, each with unique characteristics that persuade the network's behaviour. One popular example is Sigmoid, which is widely used in learning models [4] for its suitability in classification tasks due to its output range between 0 and 1. However, the Sigmoid function is known to encounter vanishing issues that hinder its convergence and effectiveness. The Rectified Linear Unit (ReLU) has gained popularity for its adaptability, efficiency, and simplicity [5]. ReLU is more effective than other functions because it activates a subset of neurons at a time rather than activating all of them at once. The weights and biases in a neural network are not updated during the back-propagation step of training when the gradient value is 0.

Different variations of ReLU AF have been proposed in the previous studies. To strengthen the deep networks by accepting the negative inputs as well, LeakyReLU is the one that adds a tiny slope for the negative values. Similarly, the Exponential Linear Unit (ELU), another variation of the ReLU was introduced to enhance training efficiency and convergence rates of models [6]. When x is negative, ELU adds a slope to the parameter. The negative values are defined using a log curve. LeakyReLU aims to fix the drawbacks of conventional ReLU with its adaptive method of permitting a tiny slope to be negative for negative inputs [7]. Although it offers a solution that addresses the dying ReLU issue. its wide range of applications and randomization of its leaky slope factor call for cautious assessment in the recommender system context. The goal of recent work has been to improve the resilience of ReLU by utilizing versions such as Scaled Exponential Linear Units (SELU) [8] and Parametric ReLU (PReLU) [9]. These activations have shown potential in recommendation tasks, where recording complicated user behaviours is critical.

Swish has demonstrated competitive performance when compared to the ReLU AF and helps to improve the model's performance by addressing the dying ReLU issue [10]. It achieves this by adding a smooth varying non-linearity to the model, which improves results and computational efficiency in some applications [11]. TSwish, a Swish variant with an extra threshold, gives AF more control over its convergence behaviour and may be very beneficial in some scenarios [12].

In a similar vein, Parametric Swish, also known as P-Swish, is an additional variant of Swish that provides tunability [13, 17] and offers computational efficiency comparable to that of Swish function. One well-known AF is Tanh, also called Hyperbolic Tangent, which gets its name from range of -1 to 1. When the desired results fall between the negative and positive extremes, Tanh is very helpful. Although Tanh generates zero-centred output to maximize the model's performance, it faces certain limitations [14].

Recommendation systems (RSs) [15, 24], a particular type of deep learning applications, have received substantial attention due to their ability to give personalized and relevant content to consumers in domains such as e-commerce, media, and networking sites. In RSs, selecting an AF is crucial because it affects the model's capacity to identify hidden trends in user-item interactions, capture user preferences, and improve suggestion accuracy. RSs confront distinct obstacles that set them apart from standard deep learning tasks. The need for reasonable exploration-exploitation trade-offs, cold-start problems, and a lack of data are some of these challenges [16]. AFs are essential for resolving these issues and improving the recommendation model's performance. Each functions efficacy varies according to the situation and choice of AF in a RS is often influenced by the features of datasets, model's

Received: December 7, 2024. Revised: February 1, 2025.

architecture, and the intended trade-offs between model expressiveness and training stability [18].

Several factors, including training techniques, hyper-parameter tuning, and the number of hidden layers in a network, must be taken into account for improved performance and fewer inaccurate outputs. One of the most crucial factors to consider is the AF. Selecting an appropriate AF for a given task can be a laborious procedure that necessitates extensive investigation and analysis [19].

Here, we introduce the ESReLU AF, which capitalizes on the latest advancements in AFs and their successful integration in RSs. ESReLU combines the benefits of sigmoid and ReLU components and enables fine-tuning via adaptive parameter (β) with its unique approach to solving the performance issues in deep learning-based RSs. Our test results on the MovieLens25 [20], Netflix Prize [21], and Online Retails [22] datasets show how well-suited ESReLU is to outperform the popular AFs, which shows promise to advance the field of DLRS's use for AF.

3. Proposed activation function

The present study introduces the ESReLU AF, which integrates the benefits of the ReLU and Sigmoid functions, to overcome the shortcomings of the existing functions and increase the deep learning model's rate of convergence. The ESReLU function is presented by the Eq. (1).

$$f(x) = (1 - \beta).sigmoid(x) + \beta.ReLU(x) \quad (1)$$

Where the Sigmoid and ReLU components are balanced via the use of β parameter.

By providing a smooth and even transition, the Sigmoid component makes sure that ESReLU function operates as intended over a wide range of input values. Definition of Sigmoid AF is given in Eq. (2).

$$sigmoid(x) = \frac{1}{1+e^{-x}} \quad (2)$$

For optimization methods like gradient decent to update the model parameters efficiently during the training, the Sigmoid function's differentiability is a prerequisite. By using the characteristics of the Sigmoid function to find complex and intricate correlations in the data, ESReLU seeks to improve the expressiveness and recognition capabilities of the neural networks. Fig. 1 illustrates the architecture of the suggested AF.

Because of the nonlinearity that ReLU imparts, the network can identify and capture traits with positive activations. Using the maximum between zero and the input x, ReLU effectively retains positive values while setting negative values to zero. The nonlinearity of the network allows it to identify and highlight items with positive activations. The ReLU function especially activates the neurons that recognize important patterns in the input data. Eq. (3) provides definition of the ReLU function.

$$ReLU(x) = max(0, x) \quad (3)$$

ReLU effectively promotes sparsity and improves the model's capacity to identify important features. The data is represented sparsely when neurons that receive negative input are set to zero. Sparsity can produce representations that are more effective because the network focuses on the most important data and fires fewer neurons. The model learns robust features with the aid of the ReLU component. Positive activations can pass through unchanged, allowing the network to record and transmit important information without the dampening effect of other AFs.

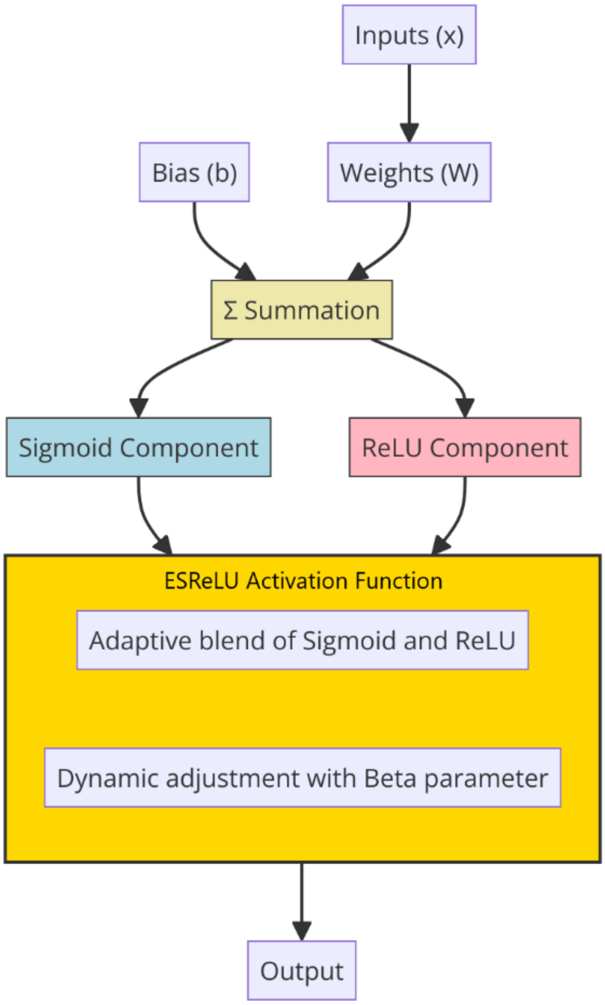
Inputs (x)
Bias (b)
Weights (W)
Σ Summation
Sigmoid Component
ReLU Component
ESReLU Activation Function
Adaptive blend of Sigmoid and ReLU
Dynamic adjustment with Beta parameter
Output

Figure. 1 Architecture of proposed ESReLU AF

DOI: 10.22266/ijies2025.0430.12

This could lead to more successful feature learning, especially in situations where it is advantageous to concentrate on positive activations.

Moreover, we can efficiently control the balance between sigmoid and ReLU AFs by dynamically adjusting the contribution of each component using the β parameter. It allows the deep learning models to be more flexible and adaptive by adjusting the β parameter dynamically, that is, parameterization in network layers enables AFs to acquire their activation through independent learning during training, along with the network's weights and biases. The ESReLU AF behaves like a normal ReLU function at $\beta = 1$, and like a traditional sigmoid function at $\beta = 0$.

This adaptive parameter improves the ESReLU's adaptability and versatility by dynamically modifying the parameter's value in response to the observed criteria during training. To handle complicated issues, however, a deep network is needed, which is challenging to train. Selecting an appropriate NN architecture is a challenging task that is based on trial and error. There are several methods proposed to maximize network performance, and combining reinforcement learning techniques with adaptive parameterization is one of them. Adapting the β parameter according to the input, allows the AF to learn and modify its response in response to the network output and is represented by the formula given in Eq. (4).

$$\beta(1+t) = \alpha.\nabla_{\beta}Obj\big(\theta,\beta(t)\big) + \beta(t) \qquad (4)$$

Where, β denotes the level of exploitation and exploration, α is the learning rate used to update the β parameter and θ represents model parameters. Obj function is the objective function which is used to calculate the gradient of the adaptive parameter at a time instance t. Incorporating the gradient information and adjusting the β values may contribute to maximizing the AF's performance.

To allow AF to explore various aspects of the parameter space, random perturbations to the β parameters are applied during the training process to prevent premature convergence of the neurons. Eq. (5) provides the modified form of the adaptive parameterization.

$$\beta(1+\ t) = \alpha.\nabla_{\beta}Obj\big(\theta,\beta(t)\big) + \beta(t)\ +\ \varepsilon(t) \quad (5)$$

where, $\varepsilon(t)$ is random agitation sampled from given distribution. This addition permits the ESReLU function to explore various behaviours and better adjustments in accordance with the changing data properties.

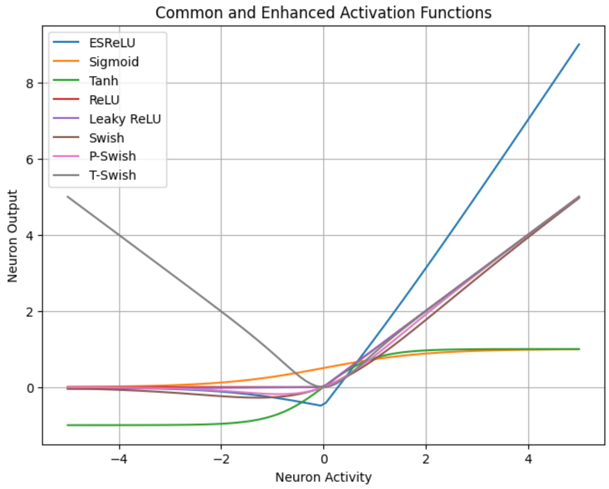

Figure. 2 Comparison of ESReLU with common AFs

This results in improved robustness and generalization of the deep learning models.

The comparison of various AFs, as shown in Fig. 2, yielded enlightening results.

It can be observed that when given huge positive and negative inputs, the sigmoid function, which returns values between 0 and 1, has a definite saturation effect. This may result in the vanishing gradient problem when the gradients are extremely small and either completely halt or drastically slow down learning. Like the Sigmoid function but with outputs spanning -1 to 1, the Tanh function is zero-centred, which might be useful during optimization. However, it also suffers from saturation.

Faster training is made possible by the ReLU function, which outputs zero otherwise and the input directly if it is positive, so avoiding the vanishing gradient issue for positive inputs. On the other hand, it can result in dying neurons with adverse inputs, meaning that neurons go dormant and don't pick up any new information during training. The Leaky ReLU and ReLU functions are not activated for negative input values, allowing them to avoid saturation. This may end up in dying neurons that never fire and hence fail to participate in the learning process. The P-Swish and T-Swish functions behave similarly to the Swish function but have a slightly different shape due to an additional parameter, which affects the activation slope.

When it comes to performance, the ESReLU AF outperforms all other functions by a considerable margin. The outcome of the ESReLU function increases linearly as the input 'x' rises from -5 to 5, suggesting a significant positive association. Since it enables the model to identify intricate patterns in the data, this is a desired feature in AFs. This implies that it may improve the neural network learning and generalization effectively.

4. Experimental results and discussion

This work aims to provide a thorough evaluation of the adaptability and efficiency of the proposed ESReLU AF in the context of recommendation process. The study was conducted using Python programming, a Jupiter notebook, and Anaconda software. We developed, trained, and tested a collaborative filtering deep model using Karas framework, to confirm the effectiveness of the proposed AF on the various datasets.

The main goal of the study is to provide a thorough comparative analysis that pits ESReLU against popular standard AFs such as Sigmoid, Tanh, ELU, ReLU, LeakyReLU, and Swish. Through a methodical examination of ESReLU in comparison with these benchmarks, our study seeks to clarify the unique benefits and potential advantages that this innovative AF might provide, offering important information for its practical implementation in DL applications and RSs.

4.1 Datasets and evaluation metrics

Using well-known datasets like MovieLens25, Netflix Prize, and the Online Retail dataset, the experiment assesses the efficacy of the recommended AF. The MovieLens25 dataset is a widely used resource in RSs domain, particularly for collaborative filtering [20]. This dataset is comprised of a wide range of user preferences and movie items with rankings on a scale of 1 to 5. The dataset has a bias toward highly rated well-known items. This bias might favour ESReLU's capacity to generalize across dense user-item interactions, but it might also limit its applicability in sparse situations. Data preprocessing involved generating a training set and mapping user and movie IDs to sequential indices.

The Netflix Prize competition made use of a substantial and wide-ranging Netflix Prize dataset [21], which contains millions of user reviews. Compared to MovieLens, the Netflix dataset is sparser, and is biased toward popular, highly rated films. Although ESReLU is well-suited for this dataset due to its adaptability in managing sparse gradients and including configurable negative input slopes, its assessment may be limited due to its bias toward frequent interactions. An overview of these datasets is shown in Table 1.

Online Retails dataset [22] is from a different domain and focuses on consumer purchasing behaviour instead of explicit ratings or reviews. Since the dataset does not contain explicit ratings, it is necessary to carefully consider how to represent the ratings' absence for the purposes of training collaborative filtering models. Models like ESReLU, which are made to hold negative inputs and dynamically adjust to changing sparsity, may benefit from such an imbalance.

These datasets, each with different characteristics, enable a thorough evaluation of the proposed AF across several recommendation scenarios.

Selecting the appropriate metrics is critical to assessing the effectiveness and performance of machine learning models. These measures provide useful insight for the model's performance and prediction accuracy. We have selected mean absolute error (MAE), mean squared error (MSE), and the square root of mean squared error (RMSE), which is derived from the MSE and facilitates understanding of precision.

Grid Search optimization technique was applied to determine the ideal values of the adaptive parameters, which covered a wide search space (β from 0 to 1 in increments of 0.05, and α from 0.1 to 1 in stages of 0.1). Each combination was assessed based on its impact on model performance parameters such as accuracy and loss. The definition of convergence criteria, as tracked by validation data, was reaching consistent performance gains over time without overfitting. This guarantees that the chosen parameters are optimal for generalizability rather than just fitting the training data.

Table 1. Datasets used to verify the AF's efficacy

Dataset	Items	Users	Ratings/ Instances	Train Set	Test Set
Movie-lens25	59047	73050	25000095	20,000,076	5,000,019
Netflix Prize	17,770	480,189	100,000,000	80,000,000	20,000,000
Online Retail	541909	406829	---	11,776	2,944

Received: December 7, 2024. Revised: February 1, 2025.

4.2 Results and analysis

This section provides a detailed discussion on the tests conducted and their results on the selected datasets. We thoroughly evaluated the performance of the proposed AF in comparison to the baseline AFs against the selected measures. The following sections provide detailed analysis of the proposed model for selected datasets.

4.2.1. Performance analysis on movielens25 dataset

At first, we trained and tested the model on MovieLens25 dataset for various epochs to verify the performance of the proposed AF. Fig. 3 presents a comparison of the performance of baseline AFs and ESReLU for the MSE measure.

It can be observed that during the first five epochs, ESReLU shows a significant advantage over other AFs, with an MSE of 0.12. This implies that the process of gathering user preferences was initially successful. But as the training goes on, Swish and its variants show steady advances in lowering MSE, making them formidable competitors.

All parameterized AFs including ESReLU, TSwish and PSwish converge to exceptionally low MSE values, especially at 50 epochs, highlighting their effectiveness in identifying intricate patterns in the MovieLens25 dataset. On the other hand, ELU and ReLU show a significantly larger MSE throughout all epochs, suggesting that these AFs face some challenges. Even with its recent improvement, Swish still cannot match the reduced MSE values achieved by ESReLU, PSwish, and TSwish.

Moreover, the analysis of MAE, as displayed in Fig. 4, emphasizes even more how well the suggested ESReLU function performs on the MovieLens25 dataset throughout different epochs. ESReLU regularly shows lower MAE values than other AFs, suggesting that it can anticipate user ratings more accurately.

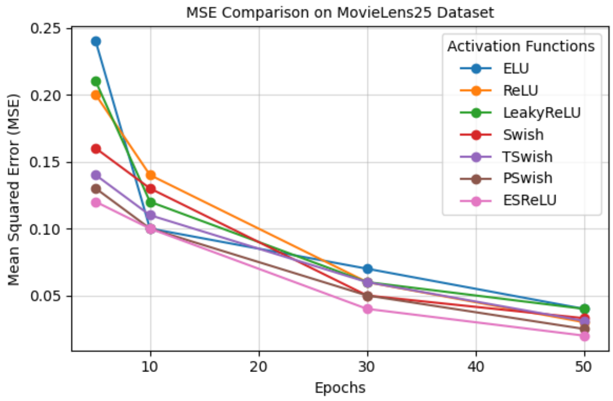

Figure. 3 MSE measure against Movielens25

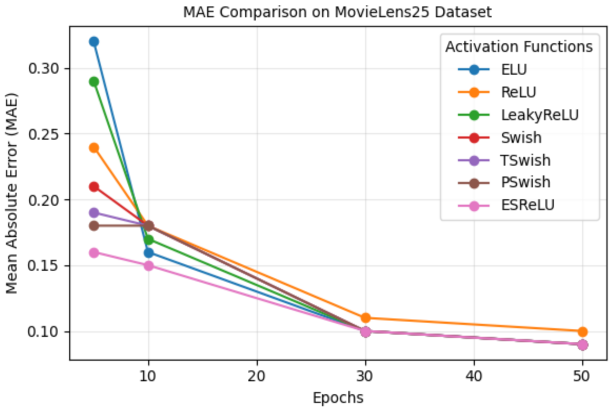

Figure. 4 MAE measure against Movielens25

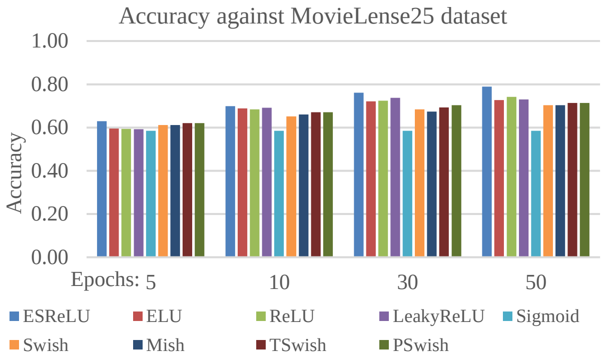

Figure. 5 Accuracy measure against Movielens25

Similarly, over 50 epochs, ESReLU outperforms all AFs with an exceptionally low MAE score of 0.09. This means that ESReLU does exceptionally well in reducing the average absolute differences between expected and actual values, which results in recommendations that are more accurate and trustworthy.

The comparison also demonstrates the limitations of conventional AFs, such as ELU, which provide higher MAE values that signify a larger departure from the actual scores. As demonstrated by its persistent superiority in minimizing MAE, ESReLU is a promising AF for recommender systems, with the potential to improve accuracy and decrease prediction errors in collaborative filtering and recommendation process.

Likewise, a comparison of the accuracy values for various AFs on the MovieLens25 dataset is shown in the accuracy graph in Fig. 5. During the first epoch, all AFs have an identical accuracy of near around 60% with ESReLU on top with an accuracy of 62.97%. This consistency points to an accuracy level that existed prior to the model being exposed to the training set. ESReLU, Swish, PSwish and TSwish consistently outperform the other AFs as the training goes on, with ESReLU winning at every epoch.

In comparison, analysis shows that ESReLU is effective in capturing complex user-item interactions, as evidenced by its greatest accuracy of 78.85% at 50th epoch. ELU, ReLU, LeakyReLU and Swish perform competitively as well, with respective accuracies of 72.66%, 74.22%, 73.02% and 74.37%. On the other hand, Sigmoid remains consistently accurate at 58.42% during the training process, indicating difficulties in simulating the dynamics of collaborative filtering with this AF.

4.2.2. Performance analysis on netflix prize dataset

Performance analysis of AFs on Netflix Prize dataset, is illustrated in Fig. 6.

It can be observed from the figure that ESReLU demonstrates constistent superiority over baseline AFs. At every examined epoch, the ESReLU exhibits much lower MSE values. This improved performance demonstrates how well ESReLU works to reduce the squared disparities between expected and actual values, leading to forecasts that are more accurate. Swish and its variants also perform similarly to ESReLU, albeit with somewhat higher MSE values. Conversely, ELU and LeakyReLU consistently show higher MSE values throughout epochs, indicating that they might have trouble accurately capturing the underlying patterns in the Netflix Prize dataset.

In comparison to ELU, ReLU, LeakyReLU, Swish, TSwish and PSwish, ESReLU consistently exhibits competitive performance for MAE scores, as seen in Fig. 7. This steady improvement points to ESReLU's effectiveness in recording absolute differences between expected and actual values, which improves prediction accuracy.

Similar patterns are seen in the performance of ELU, ReLU, LeakyReLU, Sigmoid, Tanh, and Swish, which have somewhat higher MAE values than ESReLU. Sigmoid and Swish exhibit relatively larger MAE values, suggesting difficulties in precisely forecasting ratings in the Netflix Prize sample.

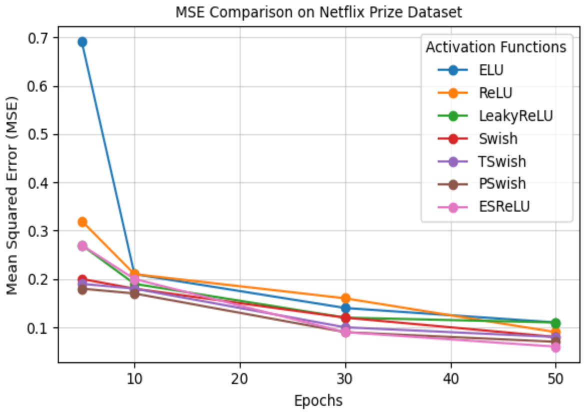

Figure. 6 MSE measure against Netflix Prize

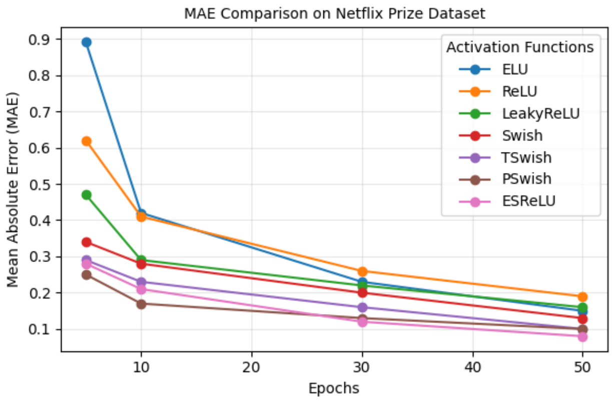

Figure. 7 MAE measure against Netflix Prize

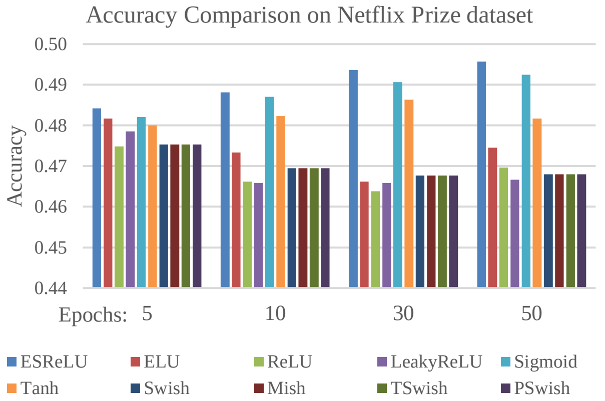

Figure. 8 Accuracy measure against Netflix Prize dataset

Similarly, accuracy comparison on the Netflix Prize dataset as illustrated by Fig. 8, shows unique patterns among different AFs over various epochs. Comparing ESReLU to ELU, ReLU, LeakyReLU, Sigmoid, Tanh, Swish, Mish, TSwish, and PSwish, it exhibits greater accuracy values and continuously retains a competitive advantage.

This pattern suggests that ESReLU can more accurately identify occurrences, which enhances model precision. With accuracy values that are close to one another, Sigmoid, Tanh, ReLU, and ELU all functions perform well. On the other hand, Swish and its variants produce lower accuracy values, which suggests their difficulty in accurately identifying preferences in the Netflix Prize dataset. RSs using the Netflix Prize dataset can benefit from ESReLU's consistent superior performance over other models, which increases the model's accuracy in classifications and predictions.

4.2.3. Performance analysis on online retail dataset

Similarly, when measured for performance accuracy on Online retails dataset, as shown in Fig. 9, it can be observed that ESReLU AF continuously performed with higher accuracy and accurate predictions as compared to other AFs.

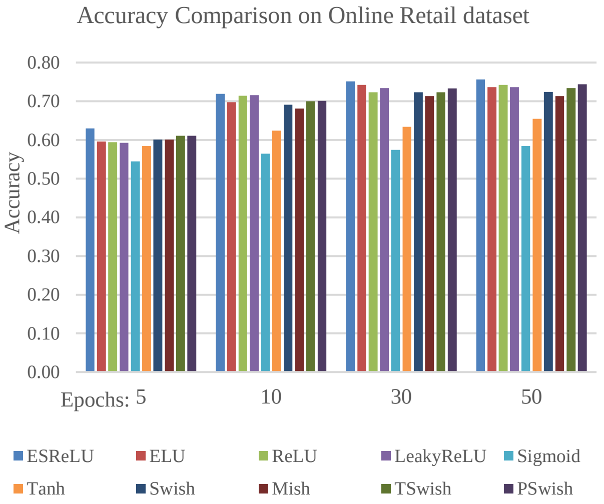

Figure. 9 Accuracy measure on Online Retail dataset

Table. 2 Performance analysis of AFs across datasets

Activation Function	MovieLens 25M	Netflix Prize	Online Retails
ELU	0.73	0.48	0.74
ReLU	0.74	0.47	0.74
LeakyReLU	0.74	0.47	0.73
Sigmoid	0.6	0.49	0.58
Tanh	0.74	0.48	0.59
Swish	0.73	0.47	0.71
Mish	0.74	0.46	0.72
TSwish (β = 1.0)	0.75	0.47	0.72
TSwish (β = 0.5)	0.74	0.43	0.73
PSwish (β = 1.0)	0.77	0.46	0.74
PSwish (β = 0.5)	0.75	0.45	0.73
ESReLU (β = 0.95, α = 0.8)	0.65	0.45	0.72
ESReLU (β = 0.95, α = 0.1)	**0.79**	**0.5**	**0.76**
ESReLU (β = 0, α = 0.8)	0.55	0.45	0.73
ESReLU (β = 0, α = 0.1)	0.70	0.46	0.74

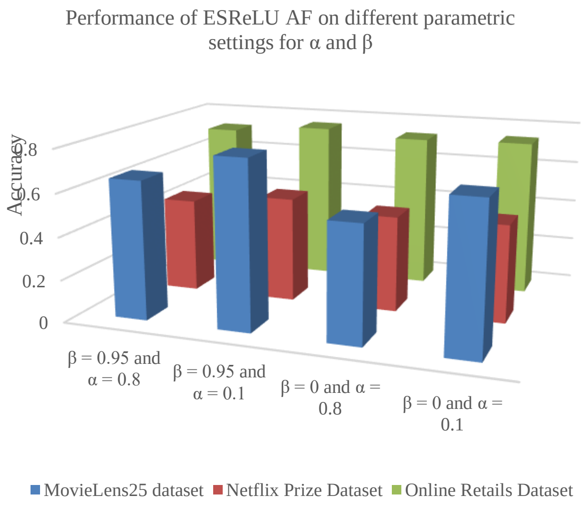

Figure. 10 Performance of ESReLU AF on different settings for α and β parameters

ESReLU outperformed other baselines with an accuracy of 74.65% at the 50 epochs, which demonstrates ESReLU's reliability and predictive accuracy.

While Sigmoid, Tanh, and Swish show lower accuracy scores, suggesting a higher number of misclassified occurrences, ELU, ReLU, and LeakyReLU demonstrate competitive accuracy. Sigmoid, Tanh, and Swish exhibit consistent accuracy across the epochs, indicating a limited capacity for prediction improvement.

Furthermore, variants of Swish AF also shown better performance in terms of accuracy. Particularly, TSwish and PSwish shown better results on denser datasets for β value approaching 1.

Table 2 shows overall performance comparison for the selected baselines AFs and the proposed ESReLU AF on various datasets used in experimental evaluation and analysis.

It can be seen that all the AFs suffered for the Netflix Prize dataset in particular, whereas most of these were able to achieve higher accuracy values on the other two datasets. Swish variants including PSwish and TSwish were also examined for different configurations for β parameter to observe their behaviour on the selected datasets.

Furthermore, Fig. 10 presents an elaboration of the individual performance of ESReLU AF in terms of sensitivity analysis under different parametric settings across datasets. The analysis confirm that slight variations in the β and α values had little effect on performance. However, performance was reduced by extreme values (e.g., $\beta = 0$ or $\alpha = 1$), highlighting the necessity of balanced parameter selection.

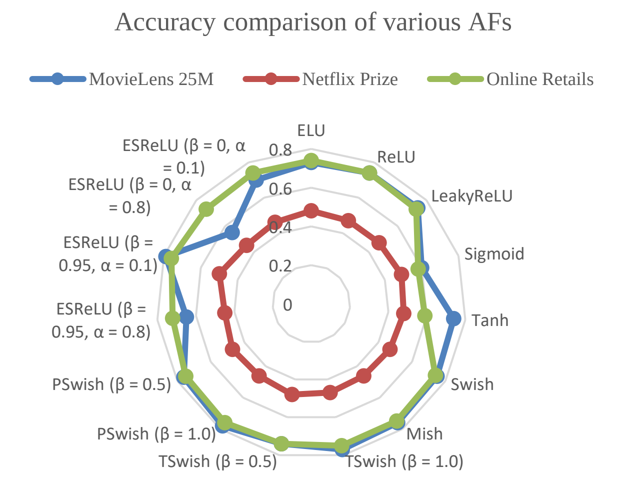

Figure. 11 Performance analysis of AFs across datasets

It can be observed that ESReLU was able to achieve a maximum accuracy of 79% for β = 0.95 and α = 0.1 on Movielens25 dataset but increasing the learning rate resulted in a decrease in its performance to a lower accuracy of 65.20%. Similarly, for β = 0 and learning rate α = 0.1, AF produced significantly better results as compared to larger values for α parameter.

Similarly, for the Netflix prize dataset, ESReLU outperformed other parametric settings producing better results for β = 0.95 and α = 0.1 values. Furthermore, ESReLU also achieved significant performance improvements for those parametric settings and achieved an improved accuracy of 75.6% for the Online Retails dataset. These results indicate that the proposed AF is likely to achieve maximum performance under these experimental settings where the adaptive parameter approaches to nearly 1, and a relatively smaller value for the learning rate parameter.

Fig. 11 presents an overall performance comparison of the proposed AF with the baseline AFs on selected datasets and different parametric settings.

It is depicted from the figure that keeping the exploitation higher with β approaching near to 1, and a lower learning rate, produced better results during the experimental analysis over 50 epochs. Proposed AF achieved an accuracy of 79% on MovieLens25 dataset which is higher in comparison to the baseline AFs. Furthermore, it is observed that ESReLU outperformed the rest of the AFs on other datasets as well when experimented with setting β = 0.95 and α = 0.1. Netflix Prize dataset was just able to secure an accuracy of 0.49 which shows the deep learning model faced difficulty in particular dataset as compared to the rest of the two.

Meanwhile, the model was able to achieve fair accuracy of 75.6% on Online Retails dataset which is closer to that of MovieLens25 dataset. On the other hand, PSwish (β = 1.0) was observed to be the second top performer that produced better results for the MovieLens25 and Online Retails datasets with a performance accuracy of 78% and 76% respectively, however its performance was lower than most of AFs in consideration when experimented on Netflix Prize datasets.

To conclude, the results depict that the proposed function performed very well and outperformed the rest of AFs for most of the time and shown continues improvement trends across selected datasets during the evaluation process.

4.2.4. Computational cost analysis

Evaluating ESReLU's computational cost in relation to baseline AFs requires taking into account both its effectiveness and trade-offs with regard to memory and training time.

In terms of training time, β parameterazation allows the AF to adjust the activation output dynamically on the basis of input range. While this flexibility improves accuracy and resilience, it adds a little amount of computing overhead when compared to baseline AFs. For instance, ReLU being the fastest one due to simple SoftMax function. Similarly, Swish and TSwish being little more complex as they involve certain operations. ESReLU is computationally more intensive due to additional parameterization and mathematical calculations, but requires often fewer epochs resulting in the faster convergence.

Similarly, ESReLU introduces additional memory cost in comparison to the non-parameterized AFs like Swish or ReLU but remains comparable to that of TSwish or PSwish. However, this cost in minimal and is unlikely to affect its applicability in large-scale datasets.

5. Conclusion

This extensive analysis of the proposed ESReLU function against other known AFs on several datasets sheds light on how well it works and how useful it is for recommendation systems. According to the experimental findings, ESReLU routinely beats or competes favourably with other AFs on a variety of metrics and datasets, including ELU, ReLU, LeakyReLU, Sigmoid, Tanh, and Swish and its variants. ESReLU performs better on the MovieLens25 and Netflix Prize datasets in terms of accuracy and loss functions. These results demonstrate ESReLU's potential to improve predictive accuracy by indicating that it is especially good at capturing intricate patterns and correlations in recommendation tasks. Likewise, ESReLU demonstrates its cross-domain versatility by maintaining competitive performance on the Online Retail dataset. The experimental results validate ESReLU as a strong AF for recommendation systems. Its constant performance across various datasets further demonstrates its applicatcability in real-wold applicatins. Although ESReLU has potential use in the recommendation systems, it is critical to recognize its limitations and more research is necessary to fully comprehend how it behaves in other domains and latest deep architectures like transformers. Furthermore, it could be interesting to investigate how well it performs in collaborative filtering models with various topologies and to incorporate hybrid models that combine content-based and collaborative filtering techniques.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Author Contributions

The authors confirm contribution to the paper as follows: Study conception and design: Syed Irteza Hussain Jafri, Rozaida Ghazali; analysis of results: Rozaida Ghazali, Yana Mazwin Mohmad Hassim; draft manuscript preparation: Syed Irteza Hussain Jafri, Rozaida Ghazali, Ahmed Khalaf Zager Al saedi, Irfan Javid; Funding acquisition: Abubakar Elsafi, Ghada Ahmed Abdelguiom. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials

The data that supports the findings of this study are openly available on the internet. The MovieLens25 dataset can be found on Kaggle at https://www.movielens-25m-Dataset. Additionally, the Netflix Prize dataset can be accessed via Kaggle at https://www.kaggle.com/datasets/netflix-inc. The UCI Machine Learning Repository has an Online Retail dataset available at www.uci.edu/dataset/352/online+retail.

References

[1] S. I. H. Jafri, R. Ghazali, I. Javid, Z. Mahmood, and A. A. Hassan, "Deep transfer learning with multimodal embedding to tackle cold-start and sparsity issues in recommendation system", *PLos ONE*, Vol. 17, No. 8, 2022.

[2] I. Javid, R. Ghazali, I. Syed, N. A. Husaini, and M. Zulqarnain, "Developing Novel T-Swish Activation Function in Deep Learning", In: *Proc. of IT and Industrial Technologies (ICIT)*, pp. 1-7, 2022.

[3] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark", *Neurocomputing*, 2022.

[4] H. Pratiwi et al., "Sigmoid activation function in selecting the best model of artificial neural networks", *Journal of Physics: Conference Series, IOP Publishing*, Vol. 1471, No. 1, p. 012010, 2020.

[5] F. Agarap, "Deep learning using rectified linear units (relu)", *arXiv:1803.08375*, 2018.

[6] Y. Zhang, J. Du, and H. Gao, "Exploiting the edge information in convolutional neural network for fine-grained image classification", In: *Proc. of the IEEE Conference on Computer Vision & Pattern Recognition*, 2017.

[7] K. Dubey and V. Jain, "Comparative study of convolution neural network's relu and leaky-relu activation functions", *Applications of Computing, Automation and Wireless Systems in Electrical Engineering: Proceedings of MARC*, pp.873-880, 2018.

[8] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks", *Advances in Neural Information Processing Systems*, Vol. 30, 2017.

[9] J. U. Rahman, R. Zulfiqar, and A. Khan, "SwishReLU: A unified approach to activation functions for enhanced deep neural networks performance", *arXiv:2407.08232*, 2024.

[10] P. P. Kumar, R. Satheesh, and H. H. Alhelou, "Impact of Activation Functions in Deep Learning Based State of Charge Estimation for Batteries", In: *Proc. of 2024 IEEE 4th International Conference on Sustainable Energy and Future Electric Transportation*, pp. 1-6, 2024.

[11] W. Hao, W. Yizhou, L. Yaqin and S. Zhili, "The role of activation function in CNN", In: *Proc. of Information Technology and Computer Application (ITCA)*, pp. 429-432, 2020.

[12] H. H. Chieng, N. Wahid, P. Ong, "Parametric flatten-TSwish: an adaptive non-linear activation function for deep learning", *arXiv:2011.03155*, 2020.

[13] M. A. Mercioni and S. Holban, "P-swish: Activation function with learnable parameters based on swish activation function in deep learning", In: *Proc. of International Symposium*

on Electronics and Telecommunications (ISETC), 2020.

[14] A. Kumar and S.S. Sodhi, "Some Modified Activation Functions of Hyperbolic Tangent (TanH) Activation Function for Artificial Neural Networks", In: *Proc. of Innovations in Data Analytics*, pp. 369-392, 2022.

[15] S.I. Jafri, R. Ghazali, I. Javid, Y.M.M. Hassim Y, M. H. Khan, "Hybrid Solution For The Cold Start Problem In Recommendation", *The Computer Journal*, 2023.

[16] F. Kolb, and A. Thor, "A MobileNet based recommendation system", In: *Proc. of the 1st International Conference on Conversational User Interfaces*, pp. 1-7, 2019.

[17] S. Zhang, M. Zhao, L. Xu, and L. Zhao, "Deep Interest Network for Click-Through Rate Prediction", In: *Proc. of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[18] A. Joglekar, D. H. Chau, and N. Sgircea, "A comparative study of activation functions for deep neural networks in recommender systems", In: *Proc. of the 26th International Conference on World Wide Web Companion*, pp. 1165-1173, 2017.

[19] MovieLens 25M Dataset, Kaggle, 2021. https://www.kaggle.com/datasets/garymk/movielens-25m-dataset

[20] Netflix Prize Dataset, Kaggle, 2019. https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data

[21] Online Retail, UCI Machine Learning Repository, 2015. https://doi.org/10.24432/C5BW33.

[22] S. Zhang, J. Lu, and H. Zhao, "Deep network approximation: Beyond relu to diverse activation functions", *Journal of Machine Learning Research*, Vol. 25, No. 35, pp. 1-39, 2024.

[23] A. D. Jagtap and G. E. Karniadakis, "How important are activation functions in regression and classification? A survey, performance comparison, and future directions", *Journal of Machine Learning for Modeling and Computing*, 2023.

[24] S. Rajput, N. Mehta, A. Singh, R. H. Keshavan, T. Vu, L. Heldt, ... and M. Sathiamoorthy, "Recommender systems with generative retrieval", *Advances in Neural Information Processing Systems*, Vol. 36, pp. 10299-10315, 2023.