IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 19, 2019, accepted November 12, 2019, date of publication November 21, 2019, date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2954859

Design of Momentum Fractional Stochastic Gradient Descent for Recommender Systems

ZESHAN ASLAM KHAN[1], SYED ZUBAIR[2,4], HANI ALQUHAYZ[3], MUHAMMAD AZEEM[4], AND ALLAH DITTA[5]

[1]Department of Electrical Engineering, International Islamic University, Islamabad 44000, Pakistan
[2]Analytics Camp, Islamabad 44000, Pakistan
[3]Department of Computer Science and Information, College of Science Al-Zulfi, Majmaah University, Al Majma'ah 11952, Saudi Arabia
[4]Faculty of Computing & IT, University of Sialkot, Sialkot 51310, Pakistan
[5]Division of Science & Technology, University of Education, Township Campus, Lahore 54000, Pakistan

Corresponding author: Syed Zubair (s.zubair@analyticscamp.org)

ABSTRACT The demand for recommender systems in E-commerce industry has increased tremendously. Efficient recommender systems are being proposed by different E-business companies with the intention to give users accurate and most relevant recommendation of products from huge amount of information. To improve the performance of recommender systems, various stochastic variants of gradient descent based algorithms have been reported. The scalability requirement of recommender systems needs algorithms with fast convergence to generate recommendations of specific items. Using the concepts of fractional calculus, an efficient variant of the stochastic gradient descent (SGD) was developed for fast convergence. Such fractional SGD (F-SGD) is further accelerated by adding a momentum term, thus termed as momentum fractional stochastic gradient descent (mF-SGD). The proposed mF-SGD method is shown to offer improved estimation accuracy and convergence rate, as compared to F-SGD and standard momentum SGD for different proportions of previous gradients, fractional orders, learning rates and number of features.

INDEX TERMS Recommender systems, e-commerce, momentum, fractional calculus, stochastic gradient descent.

I. INTRODUCTION

Nowadays, the need of e-commerce has increased rapidly and captured the businesses' interests in a short time span. At present, people are becoming habitual of using e-commerce applications and e-systems. Variety of available products pose challenges for businesses to full fill users' diverse demands. E-systems provide ease in users' taste management and allow users to explore a variety of options before taking a decision for a specific product. However, it is somehow difficult to get the useful data (information) about products for millions of users from enormous amount of data. To solve this problem, automated recommender systems are used by e-businesses with the intension to give users a precise and relevant recommendation of products.

Recommender systems are programs and procedures giving useful suggestions to users according to their liking for different products [1], [2]. An important feature of recommender systems is to predict user's interest and liking by analyzing the buying behavior of particular users, in order to give useful recommendations [3]. Recommender system plays a significant role for the customers as well as for service providers. For customers, it is used to find interesting items, to locate appropriate news content [4], discover new products and to explore new options matched with their interests. On the other hand, for service providers, recommender systems are used to promote their products, develop customer trust, obtain more knowledge about customers and enhance sales. Recommender systems have been commonly used in applications such as entertainment (e.g. music and movie recommendations), content (e.g. recommendations for documents, news and e-applications), e-commerce (e.g. recommendation for items to buy such as camera and

The associate editor coordinating the review of this manuscript and approving it for publication was Xiao-Jun Yang.

books) and services (e.g. travel and houses for rent service recommendations) [5]–[7].

There are different types of recommender systems based on different methods [8]–[13] such as collaborative filtering (CF), content based filtering (CB), demographic, knowledge-based, community-based and hybrid recommender systems. Widely applied techniques among those are CF [14]–[18] and CB [19]–[21]. In CB filtering approach, a system learns to recommend to users the same products that the user preferred in the past [22]. CF has scalability, sparseness and cold start issues whereas CB may provide overspecialized recommendations. In case of CF, recommendation of items for the specific users are based on those items which are mutually liked by other users. CF is also referred to as ''people-to-people correlation'' in [23]. There are mainly two types of methods applied under CF, memory based (neighborhood) method and model based (latent factor based) method. User to user and item to item relations are identified using neighborhood methods [22].

One of the important methods to determine latent factors is matrix factorization [24], [25]. A rating matrix consisting of user ratings for items is factorized to get latent user and item factors. Since not all users rate all the items, hence the rating matrix is sparse. For a large data set with hundreds of items and millions of users, a large rating-matrix factorization for extraction of latent factors becomes a computationally challenging task. Stochastic gradient descent (SGD) [24] and its variants [26]–[28] are commonly used for matrix factorization of such a large and sparse matrix. SGD has effectively been used for many recommender system problems and challenges [29].

With constantly increasing data sets, new techniques for matrix factorization are required that could increase the convergence speed and estimation accuracy of the recommender systems. For this purpose, fractional calculus based SGD (F-SGD) has been proposed [30]. The performance of the F-SGD can be enhanced through efficient utilization of gradient information. Therefore, in this paper, we propose a momentum version of the F-SGD for matrix factorization in which proportion of previous gradient is used to calculate the current gradient of objective function and thus yield better performance.

A. RELATED WORK

Matrix factorization based methods are one of the most popular methods for dealing with recommender system problems [31]–[33]. These techniques include singular value decomposition [34], probabilistic latent semantic analysis [35], maximum margin matrix factorization [36], [37], alternating least squares [38] and probabilistic matrix factorization [39]. For recommender systems, matrix factorization based algorithms such as stochastic gradient descent (SGD) and alternating least squares (ALS) [29], [40] became much popular due to their performance. For fast matrix factorization coordinate descent techniques [41] based on ALS and momentum based SGD method [26] were suggested.

To enhance convergence speed and estimation accuracy, several variations of SGD [24] and simple gradient descent [42] were proposed. To deal with large datasets, various SGD based scalable methods for matrix factorization were suggested in [26], where performance of SGD based approaches was demonstrated in terms of accuracy of predictions and training time. Asymmetric factor models and biased MF based efficient methods were proposed in [27]. Another SGD based strategy for matrix factorization was proposed in [43], where a learning rate schedule was presented to improve the convergence. Importance of SGD based techniques for large scale matrix factorization for recommender systems was also emphasized in [44]. The SGD based algorithms were also used for matrix completion in [45], based on non-convex matrix factorization. A non-convex SGD method was proposed in [46] for online matrix completion. An algorithm for the parallelization of SGD by MF was proposed in [47] such as distributed SGD (DSGD). However, as discussed in [48], SGD has not been parallelized for huge datasets yet. Although it is more efficient than ALS with respect to time complexity for a single iteration, it also requires more iterations for achieving a desired model. Another method was also proposed in [28] to parallelize SGD.

The weight update relations for different variants of SGD as mentioned above are based on integer order gradient. To improve the convergence speed further, fractional calculus [49]–[53] based fractional order gradient for the weight update relations of SGD have been developed in [30]. Fractional calculus based adaptive algorithms have been widely applied in diverse domains [54]–[57]. Additionally, momentum fractional least mean square algorithm has recently been developed and applied to solve parameter estimation problems [58, [59] and it has been shown that momentum based adaptive method provide better results than its standard counter-parts.

The superior performance of fractional gradient based adaptive algorithms in terms of accuracy and convergence is well proven in above referenced literature, which motivates the use of momentum fractional adaptive algorithms for recommender systems. With this motivation, in our previous work, we extended SGD to its fractional version based upon fractional calculus and named it fractional stochastic gradient descent (F-SGD) [30]. In standard SGD, integer order gradients are used while in fractional SGD (F-SGD) [30], fractional order gradient of the objective function is calculated in addition to integer order gradient. In this paper, based upon our previous work in [30], we further extend F-SGD for matrix factorization of recommender system rating matrix by incorporating momentum term and compare it with standard F-SGD and momentum SGD (mSGD).

B. OUR CONTRIBUTION

Based on the recent development [59], [30], we propose a new efficient fractional SGD algorithm by adding a momentum term to the standard F-SGD update equation termed as momentum F-SGD (mF-SGD). This variation in the F-SGD offers improvement in the estimation accuracy and convergence behavior of the recommender systems. The weights update procedure for mF-SGD includes percentage of previously calculated gradients in the weight update relation, which improves the convergence speed of mF-SGD relative to standard F-SGD for same learning rate parameters. We show that the proposed algorithm has higher convergence speed as compared to its counter-parts and achieves required estimation accuracy for lesser number of iterations as compared to the mSGD and F-SGD. The main contribution in terms of salient features of the proposed study are:

- The momentum term based F-SGD (mF-SGD) algorithm is presented for future generation recommender systems problem through efficient matrix factorization.
- The mF-SGD effectively exploits the gradient information by utilizing the proportion of previous gradients, α, in current update.
- The proposed scheme provides faster convergence for large value of α, while for lower values of α, mF-SGD provides better steady state performance.
- The correctness of the proposed approach is verified for small as well as, large datasets by testing it on ML-100K and ML-1M, respectively.
- The effectiveness of the method is validated through comparison with the standard counterparts for different learning rates, momentum weights (α) and fractional order (f_r) values.

C. PAPER OUTLINE

Rest of the paper is organized as follows: brief explanation of recommender systems problem is given in Section II. Section III defines three adaptive strategies along with their derivations and explains specifically the proposed method mF-SGD for dealing with recommender systems. Results and simulations are discussed in Section IV. Section V concludes the paper.

II. PROBLEM STATEMENT

Let $\boldsymbol{C} \in R^{p\times q}$ be a partially filled input rating matrix for recommender systems holding p users and q items. The objective function for resolving recommender system problem through matrix factorization is

$$G(\boldsymbol{a},\boldsymbol{b}) = \min_{\substack{\boldsymbol{A}\in R^{k\times p}\\ \boldsymbol{B}\in R^{k\times p}}} \sum_{(u,i)\in\Omega} (\boldsymbol{C}_{ui} - \boldsymbol{a}_u^T\boldsymbol{b}_i)^2 = \min_{\substack{\boldsymbol{A}\in R^{k\times p}\\ \boldsymbol{B}\in R^{k\times p}}} \sum_{(u,i)\in\Omega} E_{ui}^2 \quad (1)$$

where the error between observed and estimated rating is $E_{ui} = (\boldsymbol{C}_{ui} - \boldsymbol{a}_u^T\boldsymbol{b}_i)$, $\boldsymbol{a}$ and $\boldsymbol{b}$ are the u^{th} user and i^{th} item column vectors of user features matrix $\boldsymbol{A}(\boldsymbol{A} \in R^{k\times p})$ and the item feature matrix $\boldsymbol{B}(\boldsymbol{B} \in R^{k\times q})$ respectively, k denotes number of features for both users and items and Ω represents the specific indices for given ratings.

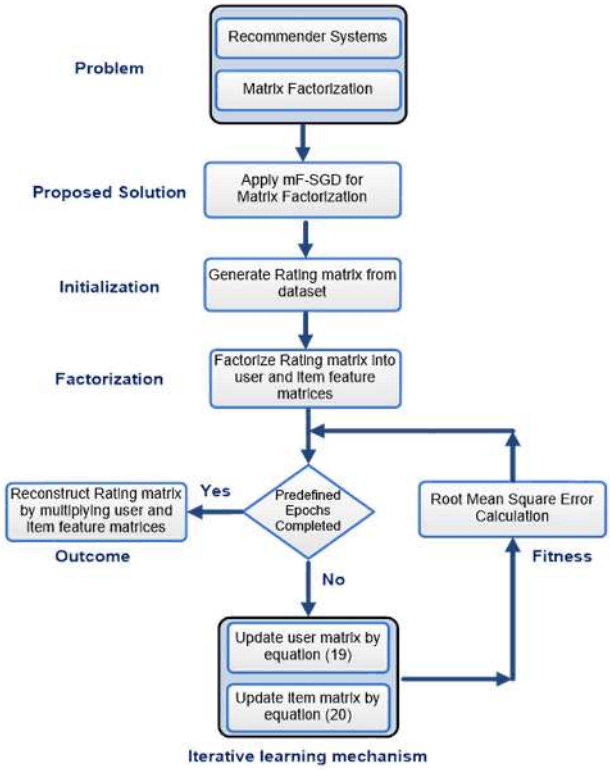

FIGURE 1. Graphical abstract of the proposed study.

The goal of objective function (1) is to discover factorized matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ from sparse matrix $\boldsymbol{C}$. The missing rating entries of $\boldsymbol{C}$ are generated through the dot product of $\boldsymbol{A}$ and $\boldsymbol{B}$, i.e. $\boldsymbol{AB}^T$. Factors of $\boldsymbol{a}_u$ for a specific user u, define the amount of interest of a user for a variety of features of an item whereas, factors of $\boldsymbol{b}_i$ for a particular item i hold the features for that specific item. Liking of user u for item i is represented by $\boldsymbol{C}_{ui}$. In subsection $\boldsymbol{A}$ of section III, matrix factorization methods are used to find the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ using SGD and F-SGD. We propose a new mF-SGD method, which performs matrix factorization for recommender systems based upon momentum F-SGD.

III. ADAPTIVE METHODS FOR RECOMMENDER SYSTEMS

This section describes three adaptive strategies for dealing with recommender systems. The following subsections present the derivations for the standard SGD, fractional SGD and the proposed mF-SGD algorithms.

A. FRACTIONAL SGD

The alternative and recursive weight update expressions using SGD for both user and item feature vectors for the n-th iteration are written as

$$\boldsymbol{a}_u(n+1) = \boldsymbol{a}_u(n) - \frac{\mu}{2}\frac{\partial G(\boldsymbol{a},\boldsymbol{b})}{\partial \boldsymbol{a}_u} \quad (2)$$

$$\boldsymbol{b}_i(n+1) = \boldsymbol{b}_i(n) - \frac{\mu}{2}\frac{\partial G(\boldsymbol{a},\boldsymbol{b})}{\partial \boldsymbol{b}_i} \quad (3)$$

where μ represents the learning rate parameter.

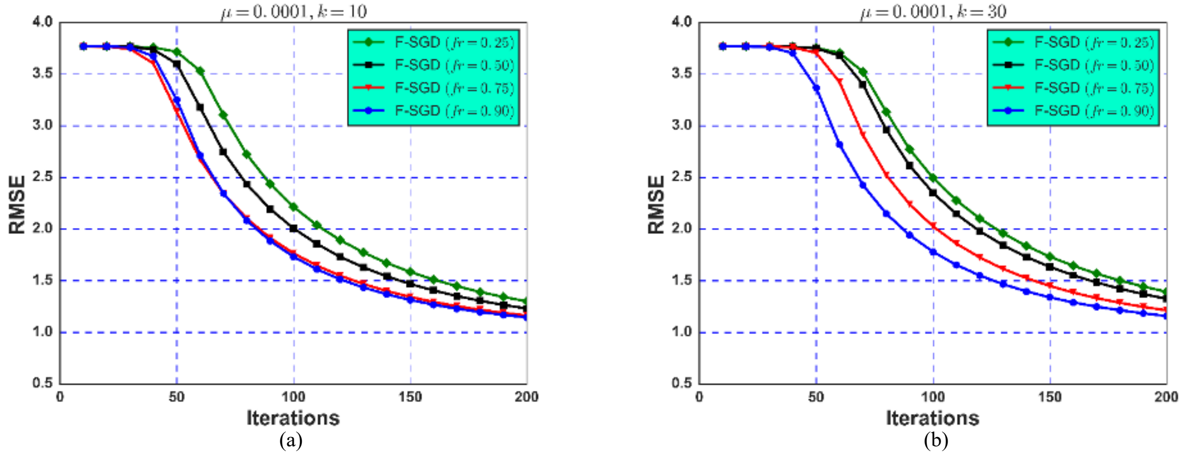

FIGURE 2. Convergence curves of F-SGD for different parametric values.

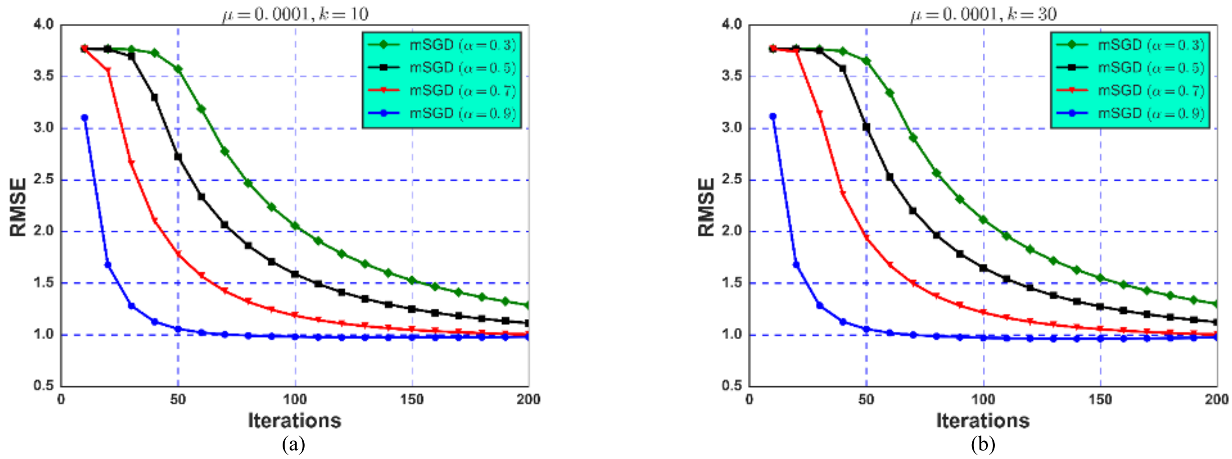

FIGURE 3. Convergence curves of mSGD for various parametric values.

By taking the gradient of objective function (1) *w.r.t* user feature vector $\boldsymbol{a}_u$, we get

$$\frac{\partial G(\boldsymbol{a},\boldsymbol{b})}{\partial \boldsymbol{a}_u} = -2E_{ui}\boldsymbol{b}_i \tag{4}$$

Likewise, by calculating the gradient of (1) with respect to item feature vector $\boldsymbol{b}_i$, we achieve

$$\frac{\partial G(\boldsymbol{a},\boldsymbol{b})}{\partial \boldsymbol{b}_i} = -2E_{ui}\boldsymbol{a}_u \tag{5}$$

At iteration n, the user and item feature weight update equations are evaluated by putting equations (4) and (5) in (2) and (3) respectively:

$$\boldsymbol{a}_u(n+1) = \boldsymbol{a}_u(n) + \mu E_{ui}\boldsymbol{b}_i(n) \tag{6}$$

$$\boldsymbol{b}_i(n+1) = \boldsymbol{b}_i(n) + \mu E_{ui}\boldsymbol{a}_u(n) \tag{7}$$

The above update equations are derived on the basis of integer order gradient and are standard SGD updates. The SGD updates can be extended to fractional calculus based SGD by incorporating fractional order gradient, in addition to the integer order gradients. This achieves better convergence rate and estimation accuracy as compared to the standard SGD [30].

In the FSGD method, user feature and item feature vectors are updated as:

$$\boldsymbol{a}_u(n+1) = \boldsymbol{a}_u(n) + \mu E_{ui}\boldsymbol{b}_i(n) - \frac{\mu_{f_r}}{2}\frac{\partial^{f_r} G(\boldsymbol{a},\boldsymbol{b})}{\partial \boldsymbol{a}_u^{f_r}} \tag{8}$$

$$\boldsymbol{b}_i(n+1) = \boldsymbol{b}_i(n) + \mu E_{ui}\boldsymbol{a}_u(n) - \frac{\mu_{f_r}}{2}\frac{\partial^{f_r} G(\boldsymbol{a},\boldsymbol{b})}{\partial \boldsymbol{b}_i^{f_r}} \tag{9}$$

where μ and μ_{f_r} denote the integer and fractional order learning rate parameters of the F-SGD algorithm respectively and f_r is the fractional order such that $0 < f_r < 1$.

For a function $y(t) = t^m$ the fractional derivative with order f_r is declared generally as [60]:

$$\mathcal{D}^{f_r} y(t) = \frac{\Gamma(\mathrm{m}+1)}{\Gamma(m - f_r + 1)} t^{m-f_r} \tag{10}$$

where, the fractional (f_r) order gradient is represented by $\mathcal{D}^{f_r}$ operator and Γ denotes a gamma function, represented as:

$$\Gamma(t) = (t-1)! \tag{11}$$

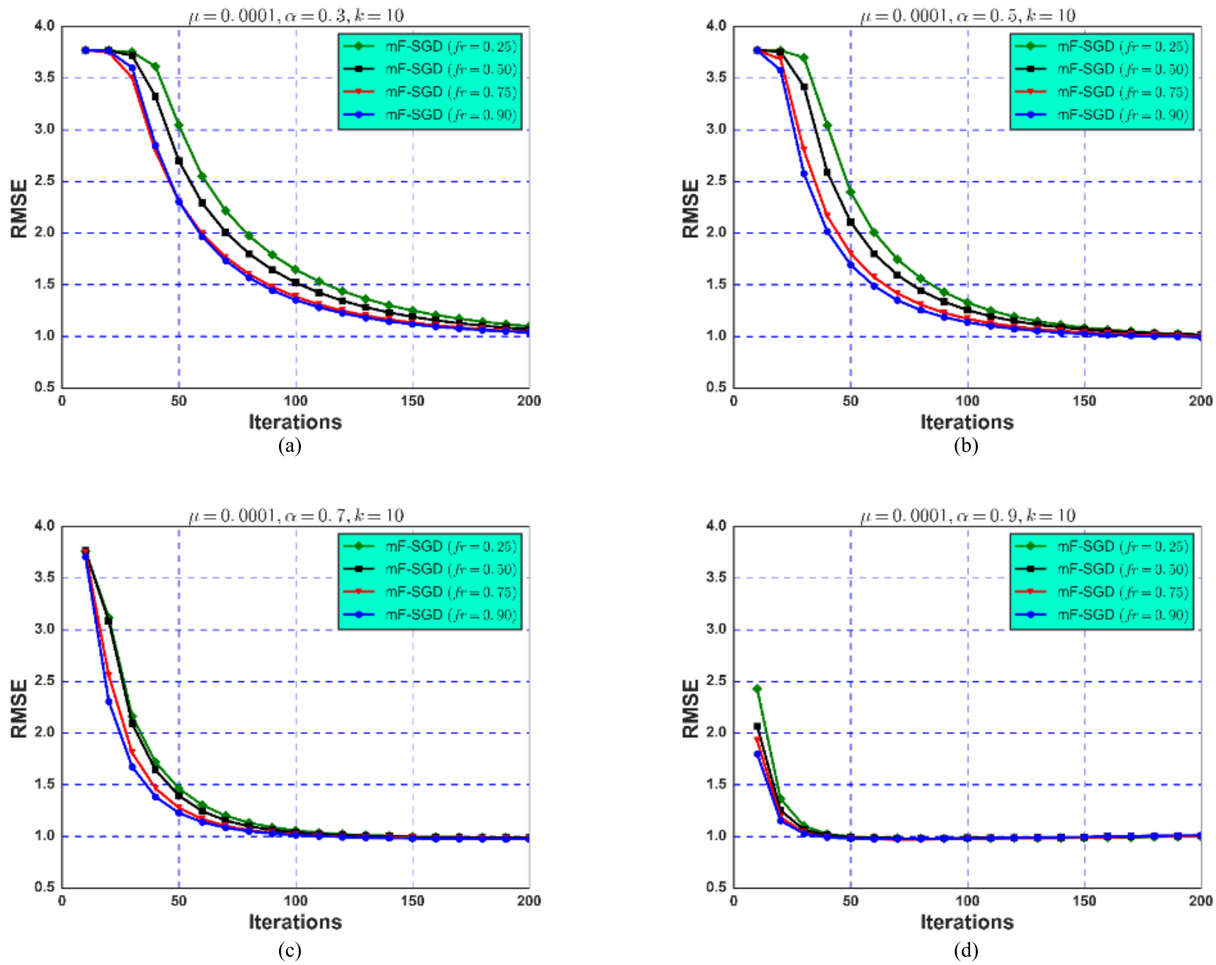

FIGURE 4. Convergence curves of mF-SGD for different f_r and α with $k = 10$, $\mu = 0.0001$.

by assuming the fractional order gradient of a constant value to be zero. Calculating the fractional order gradient of (1) with respect to the user feature vector and item feature vector respectively and using (10) and (11), we obtain:

$$\frac{\partial^{f_r} G(\boldsymbol{a}, \boldsymbol{b})}{\partial \boldsymbol{a}_u^{f_r}} \cong -2E_{ui}\boldsymbol{b}_i \frac{1}{\Gamma(2-f_r)}\boldsymbol{a}_u^{1-f_r} \tag{12}$$

$$\frac{\partial^{f_r} G(\boldsymbol{a}, \boldsymbol{b})}{\partial \boldsymbol{b}_i^{f_r}} \cong -2E_{ui}\boldsymbol{a}_u \frac{1}{\Gamma(2-f_r)}\boldsymbol{b}_i^{1-f_r} \tag{13}$$

After applying expressions (12) and (13) in (8) and (9) we get the F-SGD weight update rules for user features and items features vectors as:

$$\begin{aligned} \boldsymbol{a}_u(n+1) = \boldsymbol{a}_u(n) &+ \mu E_{ui}\boldsymbol{b}_i(n) \\ &+ \frac{\mu_{f_r}}{\Gamma\,(2-f_r)} E_{ui}\boldsymbol{b}_i(n) \\ &\circledcirc |\boldsymbol{a}_u(n)|^{1-f_r} \end{aligned} \tag{14}$$

$$\begin{aligned} \boldsymbol{b}_i(n+1) = \boldsymbol{b}_i(n) &+ \mu E_{ui}\boldsymbol{a}_u(n) \\ &+ \frac{\mu_{f_r}}{\Gamma\,(2-f_r)} E_{ui}\boldsymbol{a}_u(n) \\ &\circledcirc |\boldsymbol{b}_i(n)|^{1-f_r} \end{aligned} \tag{15}$$

where the sign $\circledcirc$ denotes element-wise multiplication of two vectors, and to ignore complex entries, absolute value of the vectors is considered. Equations (14) and (15) represent the F-SGD update relations for user and item feature vectors, respectively.

1) MOMENTUM FRACTIONAL SGD

Above mentioned F-SGD can be extended to a faster converging algorithm by introducing a momentum term in the update equation of the F-SGD. We call this proposed algorithm as momentum F-SGD. The momentum term exhibits the percentage of prior gradients instead of merely the current gradients, which is added to the existing weights. Accumulated proportions of preceding gradients help in making convergence and optimal search process faster and avoid trapping in local minima. The weight update expression for the proposed mF-SGD is given as:

$$\hat{\boldsymbol{w}}\,(n+1) = \hat{\boldsymbol{w}}\,(n) - \boldsymbol{v}\,(n+1) \tag{16}$$

where $\hat{\boldsymbol{w}}$ are the weights that need to be updated and the term $\boldsymbol{v}(n+1)$ in known as velocity term which holds the earlier gradients. The velocity vector is equal in dimension

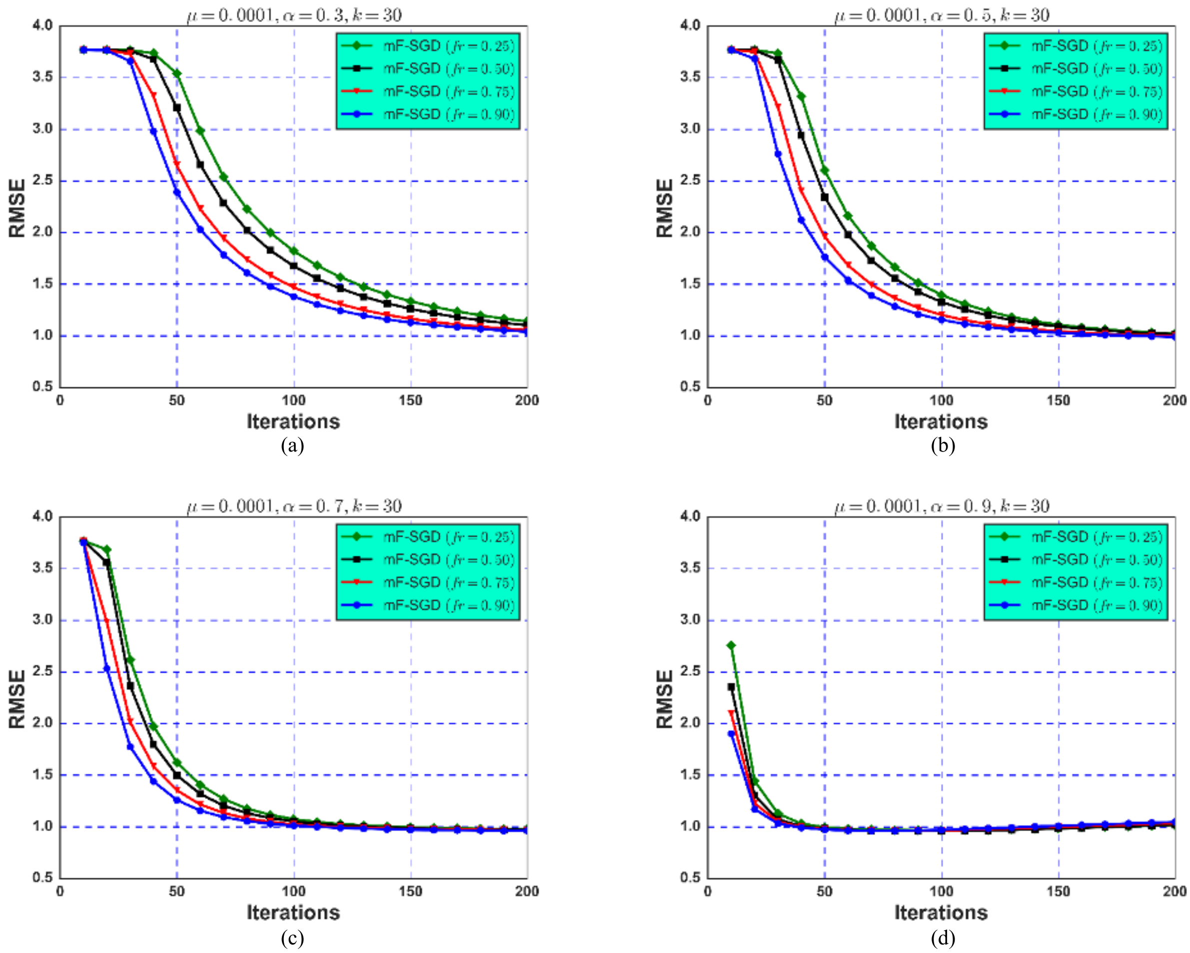

FIGURE 5. Convergence curves of mF-SGD for different f_r and α with $k = 30$, $\mu = 0.0001$.

as the weight vector and can be calculated as:

$$\boldsymbol{v}\,(n+1) = \alpha \boldsymbol{v}\,(n) + \mu \boldsymbol{g}(n) \tag{17}$$

Here in (17), the range of α is between 0 and 1 and it determines the percentage of previous gradients that is used for the current update of expression, μ is learning rate such that $\mu = \mu_{fr}$. $\boldsymbol{g(n)}$ shows the gradient (integer order and fractional order gradient) part of the expression at current iteration and is given as:

$$\boldsymbol{g}\,(n) = \frac{\partial G\left(\hat{\boldsymbol{w}}\right)}{\partial \hat{\boldsymbol{w}}} + \frac{\partial^{f_r} G\left(\hat{\boldsymbol{w}}\right)}{\partial \hat{\boldsymbol{w}}^{f_r}} \tag{18}$$

Using Equations. (16), (17) and (18), the updated weight update equations for the proposed mF-SGD for user and item vectors are:

$$\boldsymbol{a}_u(n+1) = \boldsymbol{a}_u(n) - \boldsymbol{v_1}(n+1) \tag{19}$$

$$\boldsymbol{b}_i(n+1) = \boldsymbol{b}_i(n) - \boldsymbol{v_2}(n+1) \tag{20}$$

where $\boldsymbol{v_1}(n+1)$ and $\boldsymbol{v_2}(n+1)$ are velocity terms holding previous gradient proportions of $\boldsymbol{a}_u$ and $\boldsymbol{b}_i$ respectively:

$$\boldsymbol{v_1}\,(n+1) = \alpha \boldsymbol{v_1}\,(n) + \mu \boldsymbol{g_1}(n) \tag{21}$$

$$\boldsymbol{v_2}\,(n+1) = \alpha \boldsymbol{v_2}\,(n) + \mu \boldsymbol{g_2}(n) \tag{22}$$

While initially $\boldsymbol{v_1}\,(0) = \boldsymbol{v_2}\,(0) = 0$ and $\boldsymbol{g_1}(n)$ denote the gradient (integer order and fractional order) of G *w.r.t* $\boldsymbol{a}_u$ and $\boldsymbol{g_2}(n)$ represent gradient of G *w.r.t* $\boldsymbol{b}_i$ given in (4), (12) and (5), (13) respectively.

$$\boldsymbol{g_1}\,(n) = \frac{\partial G(\boldsymbol{a},\boldsymbol{b})}{\partial \boldsymbol{a}_u} + \frac{\partial^{f_r} G(\boldsymbol{a},\boldsymbol{b})}{\partial \boldsymbol{a}_u^{f_r}} \tag{23}$$

$$\boldsymbol{g_2}\,(n) = \frac{\partial G(\boldsymbol{a},\boldsymbol{b})}{\partial \boldsymbol{b}_i} + \frac{\partial^{f_r} G(\boldsymbol{a},\boldsymbol{b})}{\partial \boldsymbol{b}_i^{f_r}} \tag{24}$$

The procedural steps for implementation of the proposed algorithm to solve recommender systems problem are provided in terms of the pseudo code in Algorithm 1. In case of standard momentum SGD algorithm, fractional gradient is not used. Thus, only integer gradient terms in (23) and (24) are considered. The overall graphical flow of the proposed study is given in Fig. 1.

IV. SIMULATIONS AND RESULTS

This section comprises of simulation parameters and results and discussion sections. Simulation parameters subsection holds dataset particulars and details of parameters involved in the simulation of F-SGD, standard mSGD and proposed

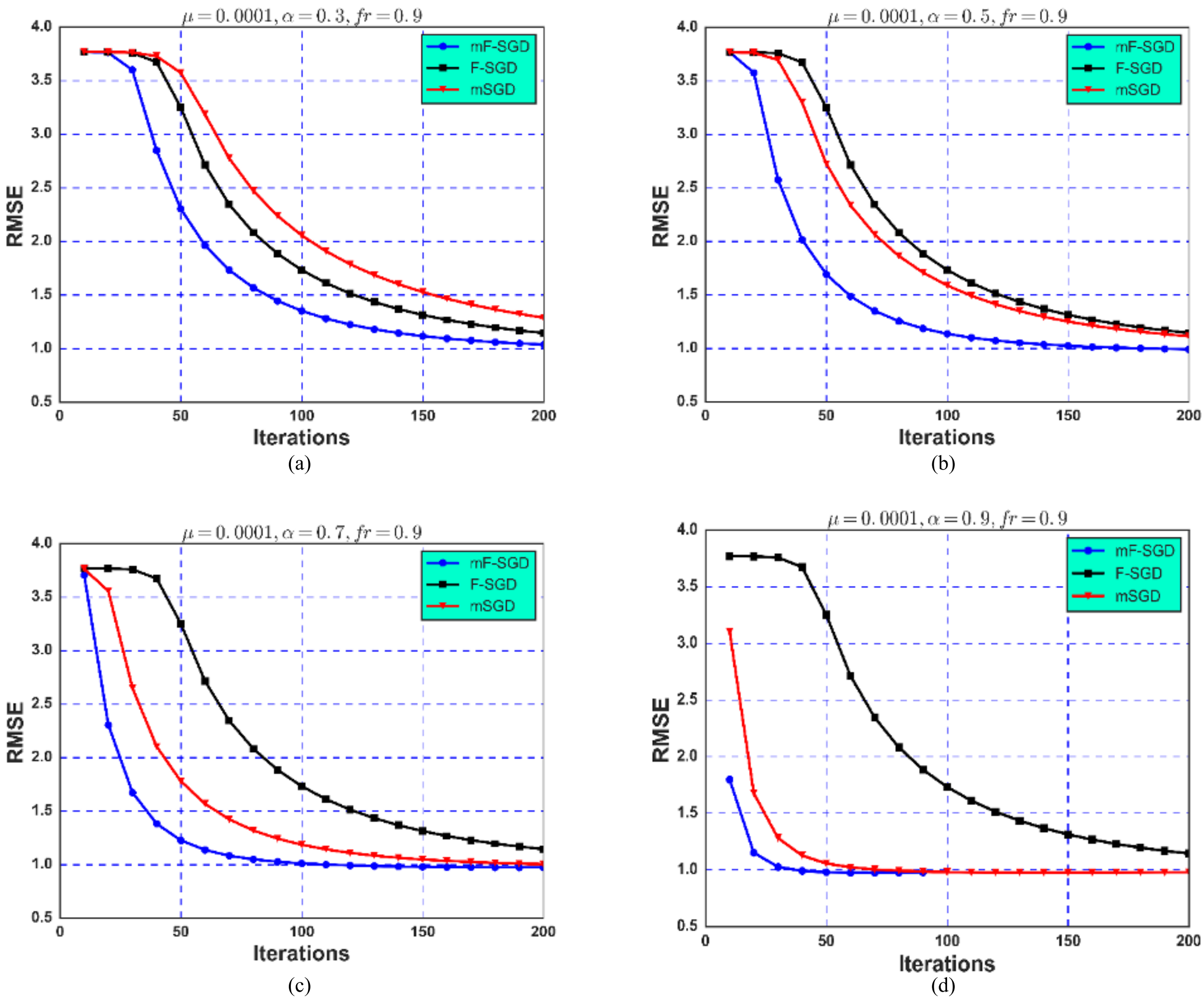

FIGURE 6. Convergence curves of mF-SGD vs F-SGD and mSGD for various α and $k = 10$.

mF-SGD algorithms. The graphical representation of results and convergence tables are given in Results and Discussion subsection to highlight the performance of the standard and proposed algorithms. This section also includes simulation results of all three methods.

A. SIMULATION PARAMETERS

In this section, simulation parameters used for standard methods like mSGD, F-SGD and the proposed mF-SGD algorithm are presented. The popular dataset, MovieLens 100K [61], is used for testing the performance of recommender systems. The MovieLens 100K dataset includes 100k movie ratings given by 943 users for choosing 1682 movies. Each user has given feedback (ratings) for at least 20 movies. The dataset also contains users' demographic information.

According to the rating matrix of the users and items from the dataset, the column density corresponds to the average rating given for individual item by a set of users and row density corresponds to the average rating given by individual user for a set of items which are given as 59.45 and 106.04, respectively. The rated entries by users for movies are from 1 to 5. The density of the dataset is 6.30%, which is calculated as:

$$Density = Ratings\ Given/(No\ of\ \ Users \times No\ of\ \ Items).$$

In addition to the ML-100K (943 × 1682) dataset, a larger dataset ML-1M (6040 × 3952) is also used to verify the viability of the proposed algorithm. Different hyper-parameters are tested for F-SGD, mSGD and mF-SGD algorithms, and their performance is compared in terms of root mean square error (RMSE).

The input rating matrix, which is sparse in nature and factorized into two matrices, is created using the above dataset. The factorized user and item matrices contain latent factors possessed by users and items respectively and are calculated by each method individually. Performance of the methods is analyzed by reconstructing the rating matrix (estimated rating matrix) by means of the dot product of factor matrices and then calculating the RMSE between the original rating matrix and the reconstructed one. The estimated rating matrix

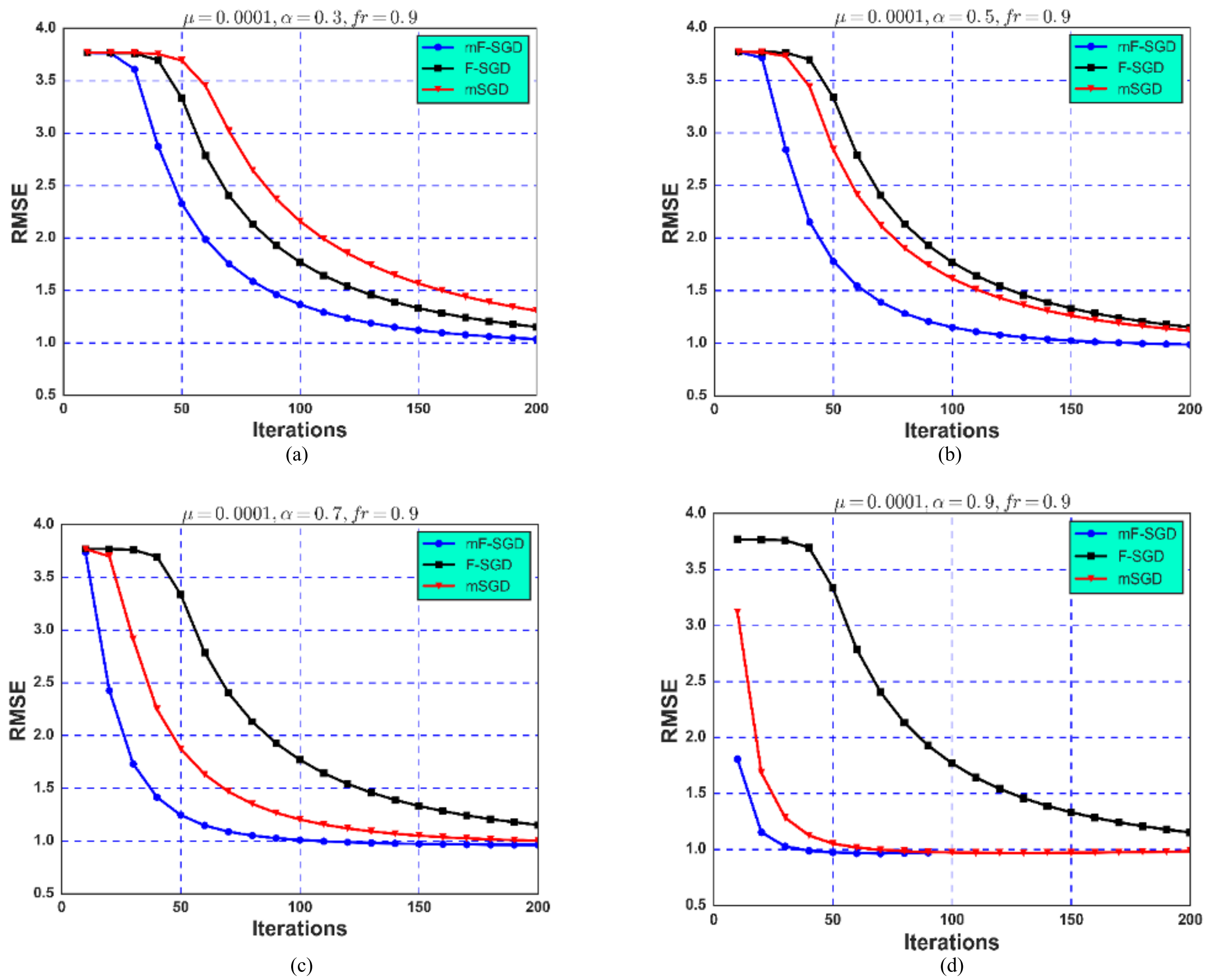

FIGURE 7. Convergence curves of mF-SGD vs F-SGD and mSGD for various α and $k = 20$.

is dense, which also gives recommendations for the movies not rated in the original matrix. RMSE is represented as:

$$RMSE_{test} = \sqrt{mean\left(\sum_{(u,i)\in\Omega_{\text{test}}} E_{ui}^2\right)}$$

For fair comparison of the proposed mF-SGD with F-SGD and mSGD, the parameters for each method are selected empirically after plotting error curves for 200 iterations and choosing the best parameter values in each case. The error curves (Figs 2-9) are obtained by randomly splitting the rating matrix in training and testing sub-matrices and calculating RMSE for each iteration. Different hyper-parameters used by each method are given in Table 1.

1) TUNING OF OPTIMAL LEARNING RATE (μ,μ_{f_r})

The learning rate μ value i.e., 10^{-4} for the given algorithms is empirically chosen after performing a number of experiments

TABLE 1. Tuning parameters for different algorithms.

Name	Symbol	Usage	Algorithms
Learning rate	μ	Step size for integer order gradient	F-SGD, mF-SGD, mSGD
Fractional learning rate	μ_{f_r}	Step size for fractional order gradient	F-SGD, mF-SGD
Fractional order	f_r	Fractional order of the gradient	F-SGD, mF-SGD
Features	k	Size of user/item feature vector	F-SGD, mF-SGD, mSGD

for different learning rate (μ) values i.e. [0.0001, 0.0005, 0.001, 0.005 and 0.01], using four different values of α i.e. [0.3, 0.5, 0.7 and 0.9] and fractional orders (f_r) i.e. [0.25, 0.5, 0.75 and 0.9] against different (k) features, to accomplish appropriate RMSE value. The values of two learning rate parameters i.e. [fractional order and integer order] are same

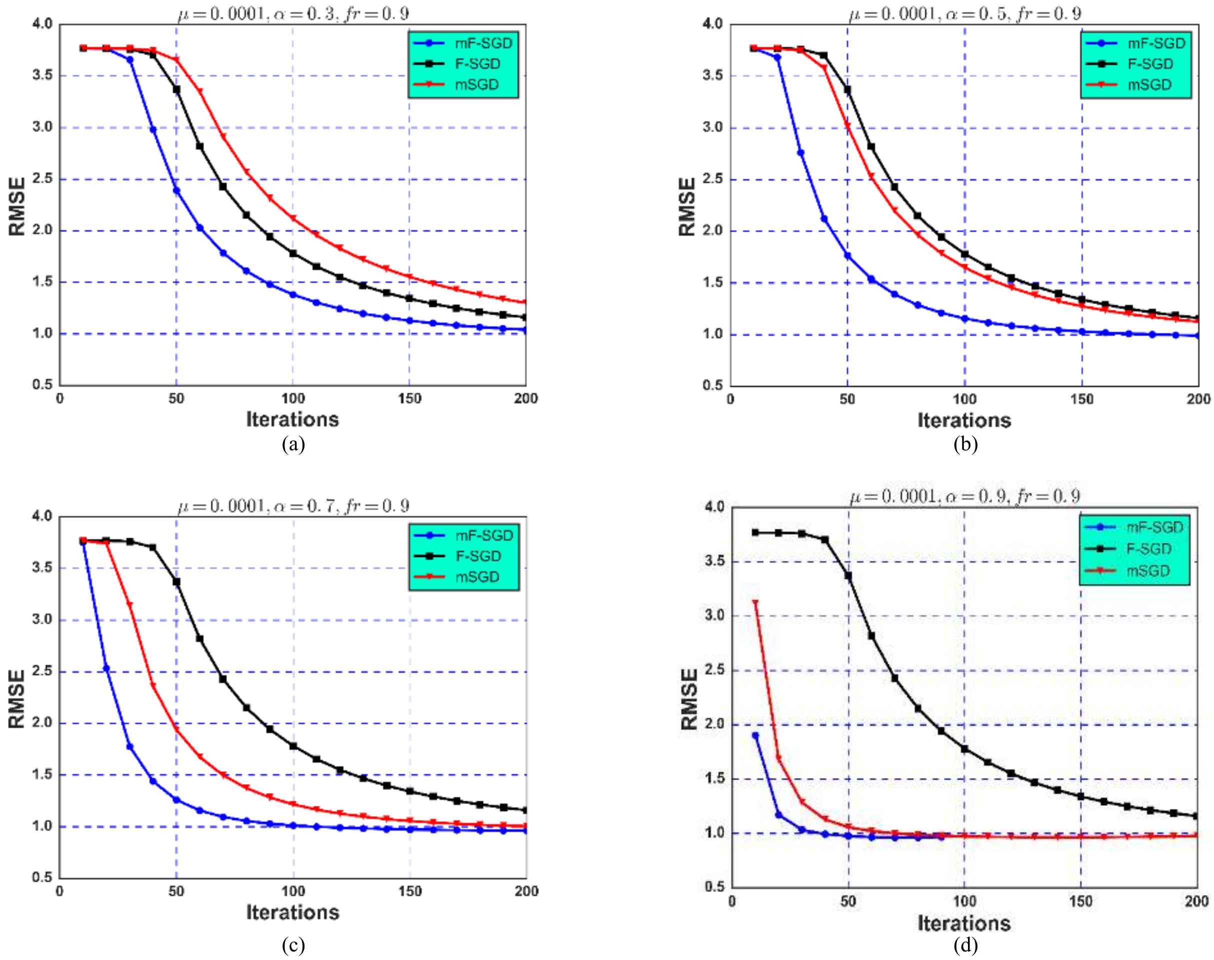

FIGURE 8. **Convergence curves of mF-SGD vs F-SGD and mSGD for various α and $k = 30$.**

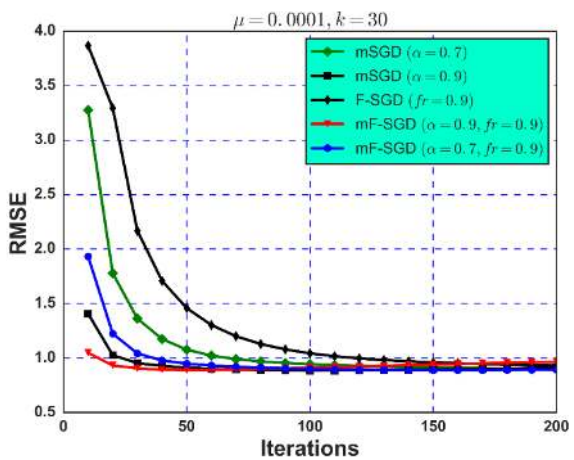

FIGURE 9. **RMSE Convergence of mF-SGD, F-SGD and mSGD for ML-1M Dataset.**

for fractional order methods, i.e. $\mu_{f_r} = \mu = \mu$. It is observed that a large value of RMSE is obtained when higher values of μ are used, i.e. [0.0005, 0.001, 0.005, and 0.01] or they did not show smooth convergence behavior for various (k) values.

2) FEATURE DIMENSION (k) SELECTION

Computational efficiency of the proposed method (mF-SGD) is examined for variations in latent features (k), i.e., [10, 20, 30]. It is observed that the computational complexity increases with the increase in number of features because time consumed by an algorithm primarily depends on the selection of the optimal number of runs and features (k). Algorithms are examined for 200 iterations to estimate the data matrix and to obtain the RMSE. It is also found that all the algorithms perform better when more features e.g., 30 features are used. In Figs 2-5, error curves for $k = 10$ and 30 are given to show the behavior of the algorithms due to variations in k.

3) SELECTING THE MOMENTUM TERM (α)

Two algorithms i.e. mSGD and mF-SGD involve the momentum term α. Therefore, for both methods, four different values of α are tested, i.e. [0.3, 0.5, 0.7 and 0.9].The proposed mF-SGD algorithm is evaluated by considering four different values of α i.e. [0.3, 0.5, 0.7 and 0.9] against four fractional orders (f_r) i.e., [0.25, 0.5, 0.75, and 0.9] and learning rates (μ) i.e., [0.0001, 0.0005, 0.001, 0.005 and 0.01]. It is seen

TABLE 2. Convergence comparison of F-SGD w.r.t RMSE attained for particular iterations.

Epochs	$\mu = 0.0001$											
	$k = 10$				$k = 20$				$k = 30$			
	$f_r =$ 0.25	$f_r =$ 0.50	$f_r =$ 0.75	$f_r =$ 0.90	$f_r =$ 0.25	$f_r =$ 0.50	$f_r =$ 0.75	$f_r =$ 0.90	$f_r =$ 0.25	$f_r =$ 0.50	$f_r =$ 0.75	$f_r =$ 0.90
10	3.769	3.769	3.769	3.770	3.770	3.769	3.769	3.769	3.769	3.769	3.769	3.769
20	3.769	3.769	3.766	3.769	3.769	3.769	3.769	3.768	3.769	3.769	3.769	3.768
30	3.767	3.764	3.744	3.759	3.769	3.767	3.764	3.761	3.768	3.768	3.768	3.761
40	3.758	3.738	3.603	3.673	3.767	3.758	3.737	3.696	3.765	3.765	3.760	3.703
50	3.716	3.600	3.136	3.251	3.758	3.714	3.571	3.335	3.753	3.750	3.706	3.369
60	3.532	3.177	2.673	2.714	3.721	3.516	3.093	2.788	3.705	3.678	3.423	2.818
70	3.105	2.747	2.345	2.346	3.562	3.072	2.656	2.405	3.523	3.401	2.910	2.427
80	2.722	2.433	2.102	2.083	3.159	2.684	2.343	2.132	3.135	2.961	2.522	2.149
90	2.436	2.195	1.915	1.885	2.766	2.396	2.109	1.927	2.772	2.614	2.240	1.942
100	2.215	2.008	1.767	1.733	2.471	2.174	1.926	1.769	2.495	2.352	2.027	1.781
110	2.038	1.857	1.647	1.612	2.242	1.997	1.780	1.643	2.278	2.146	1.859	1.654
120	1.893	1.732	1.549	1.514	2.060	1.854	1.662	1.541	2.103	1.980	1.725	1.551
130	1.773	1.629	1.468	1.434	1.911	1.735	1.564	1.458	1.958	1.844	1.616	1.467
140	1.671	1.542	1.400	1.368	1.787	1.635	1.483	1.389	1.837	1.731	1.525	1.397
150	1.585	1.468	1.343	1.314	1.683	1.550	1.415	1.331	1.734	1.635	1.450	1.340
160	1.511	1.405	1.295	1.268	1.594	1.478	1.357	1.283	1.645	1.553	1.386	1.291
170	1.447	1.352	1.254	1.229	1.518	1.416	1.308	1.242	1.569	1.484	1.332	1.250
180	1.391	1.305	1.219	1.196	1.452	1.362	1.266	1.207	1.502	1.423	1.286	1.215
190	1.342	1.265	1.188	1.169	1.396	1.315	1.230	1.178	1.444	1.371	1.247	1.185
200	1.300	1.231	1.163	1.145	1.346	1.275	1.198	1.152	1.393	1.326	1.214	1.159

that faster convergence of mF-SGD is accomplished with the increase in the value of α i.e., 0.7 but at the cost of steady state performance i.e. better steady state is achieved for lower value of α and fast convergence is achieved for higher value of α.

4) SELECTION OF THE FRACTIONAL ORDER (f_r)

The proposed method mF-SGD and FSGD are evaluated for four different values of fractional orders (f_r) i.e., [0.25, 0.5, 0.75, and 0.9], chosen from the range (0, 1). It is noticed that the rate of convergence and the steady state error increases for higher fractional orders e.g., 0.9. The impact of fractional order on the convergence speed and accuracy was explored in [62] using the fractional order in the range of (1, 1.5).

5) SIMULATION ENVIRONMENT

Experiments are executed on a laptop with (Core-i3-4005U @ 1.70 GHz) Processor and 4.00 GB DDR2 RAM. Simulations are carried out in Spyder 2.3.8, release 2015 using Python 2.7.13 (64 bit) on Windows 10 Pro Education (64 bit) operating system.

B. RESULTS AND DISCUSSION

The update rules in the standard SGD based algorithms are based on integer order gradient of the cost function. It is seen that by incorporating the fractional order gradient, the performance of mF-SGD is improved with respect to estimation accuracy and convergence at the cost of a small increase in computational complexity.

For the proposed method (mF-SGD), the convergence behaviour is estimated to demonstrate its performance. The fitness achieved in terms of RMSE for three methods (F-SGD, mSGD and mF-SGD) is presented in Tables 2-6 and graphically shown in Figs 2-5 for 200 iterations with different number of features (k), fractional order (f_r) values and previous gradient values (α) using selected learning rate i.e., $\mu = 0.0001$. It is observed from the Tables 2-6 and the learning curves in Figs 2-5 that (mSGD) and (mF-SGD) exhibit faster convergence than F-SGD. Moreover, the convergence rate of momentum based methods increase with the increase in the percentage of previous gradients (α). It is also noticed that in terms of convergence, the proposed mF-SGD method outperforms other algorithms (mSGD and F-SGD) against different parameter values.

It is also demonstrated in Tables 4-6 and depicted in Figs 4-5 that RMSE for mF-SGD decreases significantly for large values of f_r against different values of α and the number of latent features (k). The best RMSE of mF-SGD (0.962) is achieved with $\alpha = 0.7$, $f_r = 0.9$ and for $k = 30$, which clearly shows its advantage over other counterparts. It is observed that for all the algorithms (mSGD, F-SGD and mF-SGD), RMSE decreases with an increase in the number of features for different parameter values of f_r and α. It is seen from the Figs 2-5 that mF-SGD and mSGD outperform F-SGD in terms of RMSE for various parameter setting.

For comparison purpose, the error curves for the three algorithms for different values of f_r, k and α with optimal

TABLE 3. Convergence comparison of mSGD w.r.t RMSE attained for particular iterations.

Epochs	$\mu = 0.0001$											
	$k = 10$				$k = 20$				$k = 30$			
	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.9$
10	3.769	3.769	3.764	3.104	3.769	3.769	3.768	3.118	3.769	3.769	3.769	3.116
20	3.768	3.763	3.555	1.677	3.769	3.765	3.699	1.684	3.769	3.768	3.741	1.682
30	3.762	3.696	2.649	1.282	3.767	3.727	2.913	1.284	3.765	3.749	3.139	1.284
40	3.728	3.298	2.103	1.127	3.755	3.438	2.249	1.125	3.747	3.577	2.360	1.128
50	3.573	2.725	1.779	1.057	3.693	2.841	1.872	1.053	3.653	3.015	1.938	1.056
60	3.187	2.337	1.568	1.023	3.454	2.412	1.631	1.016	3.346	2.527	1.674	1.020
70	2.776	2.065	1.423	1.004	3.020	2.118	1.468	0.996	2.910	2.200	1.497	1.000
80	2.471	1.864	1.319	0.993	2.647	1.904	1.352	0.985	2.568	1.964	1.373	0.987
90	2.239	1.709	1.243	0.986	2.372	1.741	1.267	0.977	2.314	1.787	1.283	0.979
100	2.056	1.588	1.186	0.981	2.160	1.614	1.203	0.972	2.116	1.649	1.216	0.972
110	1.908	1.492	1.143	0.978	1.992	1.512	1.155	0.969	1.957	1.541	1.165	0.968
120	1.787	1.413	1.110	0.976	1.855	1.430	1.119	0.968	1.827	1.453	1.127	0.965
130	1.686	1.349	1.084	0.975	1.742	1.363	1.090	0.967	1.719	1.382	1.097	0.964
140	1.600	1.295	1.064	0.975	1.647	1.308	1.068	0.968	1.629	1.324	1.074	0.963
150	1.527	1.251	1.048	0.975	1.567	1.262	1.050	0.969	1.552	1.275	1.055	0.963
160	1.465	1.214	1.035	0.976	1.499	1.223	1.036	0.972	1.486	1.234	1.040	0.965
170	1.411	1.182	1.025	0.977	1.440	1.190	1.024	0.974	1.430	1.200	1.028	0.967
180	1.364	1.156	1.016	0.978	1.389	1.163	1.015	0.977	1.381	1.170	1.018	0.969
190	1.323	1.133	1.009	0.979	1.345	1.139	1.007	0.980	1.338	1.146	1.010	0.973
200	1.287	1.113	1.004	0.980	1.306	1.119	1.001	0.984	1.300	1.124	1.003	0.977

learning rate $\mu = 0.0001$ are given in Figs 6-8. The improved performance of proposed method i.e., mF-SGD is visibly shown in Figs 6-8 for different α and f_r values. A RMSE convergence of mF-SGD against F-SGD and mSGD with ($\mu = 0.0001, k = 10$ and $\alpha = 0.3, 0.5, 0.7$ *and* 0.9) and $f_r = 0.9$ is presented in Figures 6a, 6b, 6c and 6d respectively, the finest convergence can be seen in Figure 6c in which mF-SGD for $\alpha = 0.7$ and $f_r = 0.9$ converge significantly fast and achieved minimum RMSE (0.973) after 200 iterations. Similarly, it is shown in Figures 7a – 7d that mF-SGD also accomplished fast convergence for $k = 20$ with ($\mu = 0.0001$ and $\alpha = 0.3, 0.5, 0.7$ *and* 0.9) with fractional order ($f_r = 0.9$) but mF-SGD optimal convergence in terms of RMSE (0.963) for $k = 20$ is achieved at $\alpha = 0.7$ and $f_r = 0.9$. Likewise, Figures 8a – 8d illustrate the better convergence offered by mF-SGD over F-SGD and mSGD for $k = 30$ with ($\mu = 0.0001$ and $\alpha = 0.3, 0.5, 0.7$ *and* 0.9) using fractional order ($f_r = 0.9$), while the minimum convergence in terms of RMSE (0.962) for 30 features is achieved for $\alpha = 0.7$ and $f_r = 0.9$. In Figures 6d, 7d and 8d RMSE curves for the proposed mF-SGD are early stopped at about 100 iterations to avoid over-fitting.

Furthermore, it is observed that for different features ($k = 10, 20, 30$), F-SGD and mSGD achieved the minimum RMSE after about 200 iterations, while mF-SGD takes a smaller number of iterations. It is also seen that the momentum based algorithms have shown improved performance in terms of RMSE for a large k. It is observed that after 200 iterations, bigger value of α and a large f_r leads to a smaller RMSE except for $\alpha = 0.9$. It is also observed that for the proposed method (mF-SGD), a large value of f_r with higher value of α speeds up the convergence. In addition, increasing the f_r value leads to a small increase in the accuracy.

A Comprehensive set of investigations of the algorithms (mF-SGD, mSGD and F-SGD) are carried out for bigger dataset i.e., ML-1M (1-million), for different values of α i.e. [0.3, 0.5, 0.7 and 0.9], fractional order (f_r) i.e., [0.3, 0.5, 0.75, and 0.9] and learning rates (μ) i.e., [0.0001, 0.0005, 0.001, 0.005 and 0.01] against 30 features. The results achieved by mF-SGD, mSGD and F-SGD based on optimal parameters for the ML-1M dataset are given in the Fig 9.

Similar behavior (as that for the ML-100k dataset) in terms of convergence is observed for the ML-1M dataset. The optimal learning rate (0.0001) chosen for the bigger dataset is the same as that for the smaller dataset. It was noticed that all the compared methods show divergence or increasing trend in terms of RMSE for bigger values of learning rate μ i.e. [0.0005, 0.001, 0.005, and 0.01] against different features. The proposed algorithm performs better for higher values of both the fractioanal order f_r and weight α. It is shown that fast convergence of mFSGD is achieved for a large value of α but with the compromise of steady state performance. The best RMSE value achieved by the proposed algorithm for the ML-1M dataset is 0.887 for $\alpha = 0.7, f_r = 0.9$ after 150 iterations, whereas, the optimal RMSE values accomplished by mSGD and F-SGD after 200 iterations are 0.897 for $\alpha = 0.7$ and 0.930 for $f_r = 0.9$ respectively. However, for $\alpha = 0.9$, mF-SGD has fastest initial convergence than mSGD

TABLE 4. Convergence comparison of mF -SGD w.r.t RMSE attained for particular iterations with $k = 10$, $\mu = 0.0001$.

Epochs	$\alpha = 0.3$				$\alpha = 0.5$				$\alpha = 0.7$				$\alpha = 0.9$			
	$f_r =$ 0.25	$f_r =$ 0.50	$f_r =$ 0.75	$f_r =$ 0.90	$f_r =$ 0.25	$f_r =$ 0.50	$f_r =$ 0.75	$f_r =$ 0.90	$f_r =$ 0.25	$f_r =$ 0.50	$f_r =$ 0.75	**$f_r =$ 0.90**	$f_r =$ 0.25	$f_r =$ 0.50	$f_r =$ 0.75	$f_r =$ 0.90
10	3.769	3.769	3.768	3.770	3.769	3.769	3.768	3.767	3.761	3.764	3.747	**3.707**	2.427	2.068	1.928	1.797
20	3.767	3.765	3.750	3.762	3.766	3.755	3.685	3.576	3.116	3.083	2.556	**2.305**	1.364	1.252	1.186	1.152
30	3.752	3.718	3.503	3.600	3.700	3.419	2.801	2.576	2.163	2.090	1.808	**1.673**	1.101	1.064	1.035	1.027
40	3.612	3.320	2.790	2.850	3.042	2.589	2.163	2.015	1.718	1.642	1.463	**1.382**	1.024	1.012	0.993	0.991
50	3.044	2.697	2.307	2.304	2.396	2.107	1.803	1.692	1.463	1.393	1.276	**1.227**	0.998	0.993	0.979	0.980
60	2.551	2.290	1.990	1.964	2.006	1.803	1.573	1.488	1.303	1.244	1.167	**1.137**	0.987	0.986	0.974	0.976
70	2.216	2.008	1.767	1.733	1.746	1.594	1.417	1.350	1.200	1.153	1.100	**1.083**	0.982	0.982	0.972	0.975
80	1.973	1.801	1.603	1.568	1.562	1.445	1.307	1.255	1.132	1.097	1.059	**1.049**	0.980	0.981	0.972	0.976
90	1.789	1.643	1.479	1.445	1.428	1.336	1.228	1.186	1.087	1.061	1.033	**1.027**	0.980	0.981	0.973	0.978
100	1.646	1.520	1.383	1.352	1.327	1.255	1.170	1.137	1.056	1.038	1.016	**1.012**	0.981	0.983	0.976	0.981
110	1.531	1.423	1.308	1.280	1.250	1.194	1.128	1.100	1.036	1.023	1.004	**1.001**	0.981	0.985	0.978	0.983
120	1.438	1.345	1.248	1.224	1.191	1.149	1.096	1.073	1.021	1.012	0.996	**0.993**	0.982	0.987	0.980	0.986
130	1.363	1.282	1.201	1.180	1.147	1.114	1.072	1.053	1.010	1.005	0.990	**0.987**	0.984	0.990	0.983	0.989
140	1.300	1.231	1.163	1.145	1.112	1.088	1.053	1.037	1.003	0.999	0.986	**0.983**	0.986	0.992	0.986	0.992
150	1.249	1.189	1.132	1.117	1.085	1.068	1.039	1.025	0.997	0.995	0.983	**0.979**	0.988	0.995	0.989	0.996
160	1.206	1.155	1.107	1.094	1.064	1.052	1.027	1.015	0.993	0.992	0.980	**0.977**	0.990	0.998	0.992	0.999
170	1.171	1.127	1.086	1.075	1.048	1.040	1.019	1.007	0.990	0.990	0.979	**0.975**	0.992	1.000	0.995	1.002
180	1.142	1.103	1.070	1.060	1.035	1.030	1.012	1.001	0.987	0.988	0.978	**0.974**	0.994	1.003	0.997	1.006
190	1.117	1.084	1.056	1.048	1.025	1.022	1.006	0.996	0.985	0.987	0.977	**0.974**	0.996	1.006	1.000	1.009
200	1.097	1.068	1.044	1.038	1.017	1.016	1.001	0.992	0.984	0.985	0.977	**0.973**	0.998	1.009	1.003	1.012

TABLE 5. Convergence comparison of mF-SGD w.r.t RMSE attained for particular iterations with $k = 20$, $\mu = 0.0001$.

Epochs	$\alpha = 0.3$				$\alpha = 0.5$				$\alpha = 0.7$				$\alpha = 0.9$			
	$f_r =$ 0.25	$f_r =$ 0.50	$f_r =$ 0.75	$f_r =$ 0.90	$f_r =$ 0.25	$f_r =$ 0.50	$f_r =$ 0.75	$f_r =$ 0.90	$f_r =$ 0.25	$f_r =$ 0.50	$f_r =$ 0.75	**$f_r =$ 0.90**	$f_r =$ 0.25	$f_r =$ 0.50	$f_r =$ 0.75	$f_r =$ 0.90
10	3.769	3.769	3.769	3.769	3.769	3.768	3.769	3.769	3.767	3.764	3.760	**3.737**	2.442	2.301	2.024	1.805
20	3.768	3.768	3.767	3.761	3.764	3.751	3.727	3.712	3.514	3.193	2.783	**2.427**	1.333	1.308	1.206	1.153
30	3.758	3.754	3.734	3.609	3.674	3.408	3.005	2.840	2.408	2.177	1.908	**1.731**	1.079	1.079	1.044	1.027
40	3.680	3.618	3.357	2.874	2.938	2.601	2.285	2.153	1.870	1.711	1.518	**1.414**	1.011	1.014	0.998	0.989
50	3.255	3.039	2.674	2.328	2.329	2.131	1.887	1.778	1.567	1.449	1.309	**1.245**	0.988	0.990	0.980	0.974
60	2.721	2.530	2.242	1.987	1.960	1.829	1.634	1.545	1.378	1.289	1.187	**1.147**	0.977	0.978	0.972	0.967
70	2.353	2.190	1.951	1.754	1.712	1.621	1.463	1.390	1.254	1.187	1.114	**1.088**	0.970	0.973	0.969	0.965
80	2.087	1.945	1.743	1.586	1.535	1.470	1.343	1.283	1.171	1.120	1.068	**1.051**	0.966	0.972	0.970	0.968
90	1.885	1.761	1.588	1.461	1.405	1.359	1.255	1.207	1.114	1.075	1.038	**1.026**	0.964	0.973	0.973	0.973
100	1.728	1.619	1.469	1.366	1.308	1.275	1.191	1.151	1.075	1.046	1.018	**1.009**	0.963	0.976	0.978	0.979
110	1.602	1.506	1.377	1.292	1.234	1.211	1.143	1.110	1.048	1.025	1.004	**0.997**	0.964	0.981	0.984	0.987
120	1.500	1.416	1.304	1.234	1.178	1.162	1.107	1.080	1.028	1.010	0.994	**0.988**	0.966	0.986	0.991	0.994
130	1.417	1.342	1.246	1.188	1.135	1.124	1.079	1.056	1.014	0.999	0.987	**0.981**	0.969	0.992	0.997	1.002
140	1.348	1.282	1.200	1.151	1.102	1.095	1.058	1.038	1.003	0.991	0.982	**0.976**	0.973	0.998	1.004	1.010
150	1.290	1.233	1.162	1.121	1.077	1.072	1.041	1.025	0.996	0.986	0.978	**0.972**	0.977	1.004	1.011	1.017
160	1.242	1.192	1.131	1.097	1.057	1.054	1.028	1.014	0.990	0.981	0.975	**0.969**	0.981	1.010	1.017	1.024
170	1.203	1.158	1.106	1.077	1.042	1.039	1.018	1.005	0.985	0.978	0.973	**0.967**	0.985	1.016	1.024	1.032
180	1.169	1.130	1.085	1.061	1.030	1.027	1.009	0.998	0.982	0.975	0.971	**0.965**	0.990	1.021	1.030	1.039
190	1.141	1.107	1.068	1.047	1.020	1.018	1.002	0.992	0.979	0.974	0.970	**0.964**	0.994	1.027	1.036	1.045
200	1.117	1.088	1.054	1.036	1.012	1.010	0.997	0.987	0.977	0.972	0.969	**0.963**	0.999	1.033	1.042	1.052

and F-SGD. For about 55 early iterations mF-SGD remained convergent but it starts diverging for subsequent iterations. Whereas, mSDG achieved minimum convergence for $\alpha = 0.9$, which is attained 30 iterations later than mF-SGD but starts diverging after around 100 iterations. On the other hand mF-SGD exhibits slow initial convergent for $\alpha = 0.7$

TABLE 6. Convergence comparison of mF-SGD w.r.t RMSE attained for particular iterations with $k = 30$, $\mu = 0.0001$.

Epochs	$\alpha = 0.3$				$\alpha = 0.5$				$\alpha = 0.7$				$\alpha = 0.9$			
	$f_r =$ 0.25	$f_r =$ 0.50	$f_r =$ 0.75	$f_r =$ 0.90	$f_r =$ 0.25	$f_r =$ 0.50	$f_r =$ 0.75	$f_r =$ 0.90	$f_r =$ 0.25	$f_r =$ 0.50	$f_r =$ 0.75	$\boldsymbol{f_r =}$ **0.90**	$f_r =$ 0.25	$f_r =$ 0.50	$f_r =$ 0.75	$f_r =$ 0.90
10	3.769	3.769	3.769	3.769	3.769	3.769	3.769	3.768	3.768	3.768	3.765	**3.752**	2.755	2.360	2.094	1.903
20	3.769	3.768	3.766	3.764	3.767	3.765	3.749	3.683	3.685	3.562	2.983	**2.533**	1.446	1.303	1.227	1.172
30	3.765	3.759	3.728	3.658	3.736	3.672	3.216	2.761	2.619	2.363	2.014	**1.776**	1.129	1.073	1.051	1.033
40	3.736	3.677	3.321	2.979	3.319	2.946	2.402	2.123	1.973	1.801	1.585	**1.439**	1.032	1.007	0.999	0.993
50	3.539	3.206	2.648	2.392	2.602	2.345	1.960	1.766	1.622	1.499	1.353	**1.260**	0.997	0.983	0.978	0.975
60	2.984	2.655	2.227	2.030	2.164	1.978	1.683	1.541	1.407	1.318	1.217	**1.157**	0.982	0.971	0.967	0.966
70	2.539	2.287	1.942	1.785	1.872	1.731	1.497	1.390	1.269	1.205	1.134	**1.094**	0.975	0.964	0.962	0.962
80	2.228	2.024	1.738	1.609	1.666	1.555	1.366	1.286	1.177	1.132	1.081	**1.055**	0.970	0.960	0.961	0.962
90	1.998	1.827	1.586	1.479	1.513	1.426	1.272	1.211	1.116	1.084	1.047	**1.030**	0.968	0.959	0.963	0.966
100	1.821	1.675	1.469	1.381	1.397	1.329	1.203	1.156	1.075	1.052	1.025	**1.012**	0.968	0.959	0.967	0.972
110	1.681	1.555	1.378	1.304	1.308	1.255	1.152	1.116	1.047	1.030	1.009	**0.999**	0.969	0.962	0.972	0.979
120	1.568	1.458	1.306	1.244	1.238	1.197	1.114	1.085	1.027	1.014	0.998	**0.990**	0.973	0.966	0.979	0.987
130	1.475	1.379	1.249	1.197	1.184	1.153	1.085	1.062	1.014	1.002	0.990	**0.983**	0.977	0.971	0.986	0.995
140	1.399	1.315	1.203	1.159	1.142	1.118	1.062	1.044	1.004	0.993	0.983	**0.977**	0.982	0.977	0.994	1.003
150	1.335	1.262	1.165	1.128	1.109	1.091	1.045	1.030	0.996	0.986	0.979	**0.973**	0.987	0.984	1.000	1.011
160	1.281	1.218	1.135	1.103	1.083	1.069	1.031	1.019	0.990	0.981	0.975	**0.969**	0.993	0.991	1.008	1.019
170	1.236	1.181	1.110	1.083	1.063	1.051	1.020	1.010	0.986	0.977	0.972	**0.966**	0.999	0.998	1.015	1.026
180	1.199	1.150	1.089	1.066	1.046	1.037	1.012	1.002	0.982	0.974	0.969	**0.965**	1.005	1.005	1.022	1.034
190	1.167	1.125	1.072	1.052	1.034	1.025	1.004	0.996	0.979	0.971	0.968	**0.963**	1.011	1.012	1.030	1.041
200	1.140	1.103	1.058	1.040	1.023	1.016	0.998	0.991	0.976	0.969	0.966	**0.962**	1.017	1.020	1.037	1.048

TABLE 7. Performance comparison of mF-SGD with deep learning based Matrix Factorization methods.

DATASETS	METHODS	RMSE
ML-100K	ConvMF	1.000
	DBPMF	0.990
	DCBPMF	0.985
	mF-SGD	**0.962**
ML-1M	ConvMF	0.980
	DBPMF	0.945
	DCBPMF	0.943
	mF-SGD	**0.887**

and achieved minimum RMSE = 0.887 after 150 iterations and maintains stable steady state behavior for almost 180 iterations, which is not shown by mSGD for $\alpha = 0.9$.

C. PERFORMANCE COMPARISON WITH DEEP LEARNING BASED MATRIX FACTORIZATION MODELS

Apart from competing methods presented in this paper i.e., mSGD and F-SGD, the effectiveness of the proposed mF-SGD is also proved by comparing it with recent deep learning based matrix factorization methods using **ML-100K** and **ML-1M datasets.** The performance in terms of RMSE of deep learning based models is reported with optimum parameter values. Overview of the deep learning based MF models given in [63] is as follows.

- **ConvMF** [63] To improve the prediction accuracy of ratings, a context-aware recommendation model ConvMF) is proposed. For achieving high prediction accuracy, ConvMF integrates CNN with PMF to capture contextual information of documents as stated in [64].
- **DBPMF** [63]: Deep Bias Probabilistic Matrix Factorization model (DBPMF) uses CNN to extract hidden user/item characteristics. Moreover, DBPMF also adds bias into PMF to tract user ratings behavior and item reputation.
- **DCBPMF** [63]: Deep Constrain Bias PMF method is used to further improve the performance of standard DBPMF by adding constrain to the user specific and item specific vectors.

The performance comparison between proposed mF-SGD and deep learning based MF methods for recommender systems [63] is demonstrated in Table 7 for both ML-100K and ML-1M datasets. It is noticed from the results given in Table 7 that mF-SGD achieved improved results than **ConvMF, DBPMF** and **DCBPMF** for both datasets. The best performance (RMSE = 0.962) accomplished by mF-SGD against counterparts for ML-100K is with $\alpha = 0.7, f_r = 0.7$ and $k = 30$. Whereas, for ML-1M dataset mF-SGD also achieved finest RMSE = 0.887 against competing methods with similar parameters setting i.e., $\alpha = 0.7$, $f_r = 0.9$ and $k = 30$. The significant performance with regard to

Algorithm 1 Stepwise Pseudo-Code of Proposed mF-SGD for Recommender Systems

Input: $\boldsymbol{C} \in R^{p\times q}$: Sparse rating matrix, μ: Integer order learning rate, f_r: Fractional order and μ_{f_r}: Fractional order learning rate, α: Percentage of previous gradients, k: Number of features, Epochs

Output: $\hat{C} \in R^{p\times q}$: Rebuilt rating matrix, $\boldsymbol{A} \in R^{k\times p}$: Learned user factor matrix, $\boldsymbol{B} \in R^{k\times q}$: Learned item factor matrix

1) Divide $\boldsymbol{C}$ into Train ($Train \in R^{p\times q}$) and Test ($Test \in R^{p\times q}$)
2) Initialize factorized matrices $\boldsymbol{A} \in R^{k\times p}$ and $\boldsymbol{B} \in R^{k\times q}$ randomly
3) Find position indices $\wp = (u, i)$ of all non-zero entries in Train
4) **While** $epoch < epochs$ **do**
5) **for all** $(u, i) \in \wp$ **do**
6) Compute $E_{ui} = (\boldsymbol{C}_{ui} - \boldsymbol{a}_u^T \boldsymbol{b}_i)$
7) Compute Gradients $w.r.t$ $\boldsymbol{a}_u$ via Eq. (**23**)
8) Compute Gradients $w.r.t$ $\boldsymbol{b}_i$ via Eq. (**24**)
9) Compute weights vector holding previous gradients of $\boldsymbol{a}_u$ via Eq. (**21**)
10) Compute weights vector holding previous gradients of $\boldsymbol{b}_i$ via Eq. (**22**)
11) Learn $\boldsymbol{a}_u$, u^{th} factor vector of $\boldsymbol{A}$ through Eq. (**19**)
12) Learn $\boldsymbol{b}_i$, i^{th} factor vector of $\boldsymbol{B}$ through Eq. (**20**)
13) **end for**
14) Reconstruct $\hat{\boldsymbol{C}} = \boldsymbol{A}\boldsymbol{B}^T$ via learned factor matrices $\boldsymbol{A}$ and $\boldsymbol{B}$
15) Calculate RMSE using *Test*
16) $epoch = epoch + 1$
17) **end while**

deep learning based MF methods confirms the usefulness of the proposed mF-SGD for proposing accurate and fast recommendations.

V. CONCLUSION

The momentum fractional stochastic gradient descent paradigm has been presented for solving the recommender systems problem. The algorithm is designed for making matrix factorization more efficient to solve future generation complex recommender systems. The correctness of the proposed mF-SGD is established by testing it on ML-100K and ML-1M datasets. The usefulness of the presented scheme mF-SGD is confirmed by comparing with standard mSGD and F-SGD for different learning rates, momentum weights (α) and fractional order (f_r) values. It is observed that increase in α value (i.e. the percentage of previous gradients) has directly proportional to the convergence rate. It is also witnessed that faster convergence in the mF-SGD is achieved for large values of α and f_r. However, a smaller value of α with small value of f_r results in slower convergence. It is noticed that by incorporating large value of α, initial convergence for momentum-based algorithms, i.e., mF-SGD and mSGD, is much faster than F-SGD without momentum term, while for lower values of α, mF-SGD provides better steady state performance. It is also witnessed that with the increase in number of features (k), the proposed algorithm gives better results. Moreover, the proposed mF-SGD provides superior results to other methods for all k variants.

REFERENCES

[1] C. C. Aggarwal, "An introduction to recommender systems," in *Recommender Systems*. Cham, Switzerland: Springer, 2016, pp. 1–28.
[2] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl. Syst.*, vol. 46, pp. 109–132, Jul. 2013.
[3] I. Heimbach, J. Gottschlich, and O. Hinz, "The value of user's Facebook profile data for product recommendation generation," *Electron. Markets*, vol. 25, no. 2, pp. 125–138, Jun. 2015.
[4] M. Karimi, D. Jannach, and M. Jugovac, "News recommender systems—Survey and roads ahead," *Inf. Process. Manag.*, vol. 54, no. 6, pp. 1203–1227, Nov. 2018.
[5] A. O. Afolabi and P. Toivanen, "Integration of recommendation systems into connected health for effective management of chronic diseases," *IEEE Access*, vol. 7, pp. 49201–49211, 2019.
[6] F. Lin, Y. Zhou, I. You, J. Lin, X. An, and X. Lü, "Content recommendation algorithm for intelligent navigator in fog computing based IoT environment," *IEEE Access*, vol. 7, pp. 53677–53686, 2019.
[7] P. Chamoso, A. Rivas, S. Rodríguez, and J. Bajo, "Relationship recommender system in a business and employment-oriented social network," *Inf. Sci.*, vols. 433–434, pp. 204–220, Apr. 2018.
[8] S. Alonso, J. Bobadilla, F. Ortega, and R. Moya, "Robust model-based reliability approach to tackle shilling attacks in collaborative filtering recommender systems," *IEEE Access*, vol. 7, pp. 41782–41798, 2019.
[9] P. Valdiviezo-Diaz, F. Ortega, E. Cobos, and R. Lara-Cabrera, "A collaborative filtering approach based on Naïve Bayes classifier," *IEEE Access*, vol. 7, pp. 108581–108592, 2019.
[10] M. Hong and J. J. Jung, "Multi-Sided recommendation based on social tensor factorization," *Inf. Sci.*, vol. 447, pp. 140–156, Jun. 2018.
[11] H. Samin and T. Azim, "Knowledge based recommender system for academia using machine learning: A case study on higher education landscape of Pakistan," *IEEE Access*, vol. 7, pp. 67081–67093, 2019.
[12] S. Meng, L. Qi, Q. Li, W. Lin, X. Xu, and S. Wan, "Privacy-preserving and sparsity-aware location-based prediction method for collaborative recommender systems," *Futur. Gener. Comput. Syst.*, vol. 96, pp. 324–335, Jul. 2019.
[13] D. Peng, W. Yuan, and C. Liu, "HARSAM: A hybrid model for recommendation supported by self-attention mechanism," *IEEE Access*, vol. 7, pp. 12620–12629, 2019.
[14] S. Köhler, T. Wöhner, and R. Peters, "The impact of consumer preferences on the accuracy of collaborative filtering recommender systems," *Electron. Markets*, vol. 26, no. 4, pp. 369–379, Nov. 2016.
[15] C. He, D. Parra, and K. Verbert, "Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities," *Expert Syst. Appl.*, vol. 56, pp. 9–27, Sep. 2016.
[16] R. Chen, Q. Hua, Y.-S. Chang, B. Wei, L. Zhang, and X. Kong, "A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks," *IEEE Access*, vol. 6, pp. 64301–64320, 2018.
[17] T. Cunha, C. Soares, and A. C. P. L. F. de Carvalho, "Metalearning and recommender systems: A literature review and empirical study on the algorithm selection problem for Collaborative Filtering," *Inf. Sci.*, vol. 423, pp. 128–144, Jan. 2018.
[18] J. Li, "Category preferred canopy-K-means based collaborative filtering algorithm," *Futur. Gener. Comput. Syst.*, vol. 93, pp. 1046–1054, Apr. 2019.

[19] J. Salter and N. Antonopoulos, "CinemaScreen recommender agent: Combining collaborative and content-based filtering," *IEEE Intell. Syst.*, vol. 21, no. 1, pp. 35–41, Jan. 2006.
[20] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*. Berlin, Germany: Springer, 2007, pp. 325–341.
[21] E. Aslanian, M. Radmanesh, and M. Jalili, "Hybrid recommender systems based on content feature relationship," *IEEE Trans. Ind. Informat.*, to be published.
[22] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
[23] J. Ben Schafer, J. A. Konstan, and J. Riedl, "E-commerce recommendation applications," in *Applications of Data Mining to Electronic Commerce*, vol. 5, nos. 1–2. Boston, MA, USA: Springer, 2001, pp. 115–153.
[24] X. Luo, Y. Xia, and Q. Zhu, "Incremental collaborative filtering recommender based on regularized matrix factorization," *Knowl. Syst.*, vol. 27, pp. 271–280, Mar. 2012.
[25] X. Luo, Y. Xia, and Q. Zhu, "Applying the learning rate adaptation to the matrix factorization based collaborative filtering," *Knowl. Syst.*, vol. 37, pp. 154–164, Jan. 2013.
[26] G. Takács, I. Pilászy, B. Németh, and D. Tikk, "Scalable collaborative filtering approaches for large recommender systems," *J. Mach. Learn. Res.*, vol. 10, pp. 623–656, Mar. 2009.
[27] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering categories and subject descriptors," in *Proc. KDDCup*, 2007, pp. 39–42.
[28] W.-S. Chin, Y. Zhuang, Y.-C. Juan, and C.-J. Lin, "A fast parallel stochastic gradient method for matrix factorization in shared memory systems," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 1, pp. 1–24, Mar. 2015.
[29] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, "Large-scale parallel collaborative filtering for the Netflix prize," in *Algorithmic Aspects in Information and Management* (Lecture Notes in Computer Science), vol. 5034. Berlin, Germany: Springer, 2008, pp. 337–348.
[30] Z. A. Khan, N. I. Chaudhary, and S. Zubair, "Fractional stochastic gradient descent for recommender systems," *Electron. Markets*, vol. 29, no. 2, pp. 275–285, Jun. 2019.
[31] T. V. R. Himabindu, V. Padmanabhan, and A. K. Pujari, "Conformal matrix factorization based recommender system," *Inf. Sci.*, vol. 467, pp. 685–707, Oct. 2018.
[32] C. Xu, "A novel recommendation method based on social network using matrix factorization technique," *Inf. Process. Manage.*, vol. 54, no. 3, pp. 463–474, May 2018.
[33] B. Zhu, F. Ortega, J. Bobadilla, and A. Gutiérrez, "Assigning reliability values to recommendations using matrix factorization," *J. Comput. Sci.*, vol. 26, pp. 165–177, May 2018.
[34] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system—A case study," Dept. Comput. Sci. Eng., Univ. Minnesota, Minneapolis, MN, USA, Tech. Rep. TR-00-043, 2000.
[35] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 89–115, Jan. 2004.
[36] N. Srebro, J. D. M. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in *Proc. NIPS*, vol. 17, 2004, pp. 1329–1336.
[37] V. Kumar, A. K. Pujari, S. K. Sahu, V. R. Kagita, and V. Padmanabhan, "Collaborative filtering using multiple binary maximum margin matrix factorizations," *Inf. Sci.*, vol. 380, pp. 1–11, Feb. 2017.
[38] R. M. Bell and Y. Koren, "Scalable collaborative filtering with jointly derived neighborhood interpolation weights," in *Proc. 7th IEEE Int. Conf. Data Mining (ICDM)*, Oct. 2007, pp. 43–52.
[39] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. Neural Inf. Process. Syst.*, 2007, pp. 1257–1264.
[40] P.-L. Chen, "A linear ensemble of individual and blended models for music rating prediction," in *Proc. KDDCup*, Jun. 2011, pp. 21–60.
[41] H.-F. Yu, C.-J. Hsieh, S. Si, and I. Dhillon, "Scalable coordinate descent approaches to parallel matrix factorization for recommender systems," in *Proc. IEEE 12th Int. Conf. Data Mining*, Dec. 2012, pp. 765–774.
[42] S. Funk. (2006). *Netflix Update: Try This at Home*. [Online]. Available: https://sifter.org/simon/journal/20061211.html
[43] W.-S. Chin, Y. Zhuang, Y.-C. Juan, and C.-J. Lin, *A Learning-Rate Schedule for Stochastic Gradient Methods to Matrix Factorization* (Lecture Notes in Computer Science), vol. 9077. Cham, Switzerland: Springer, 2015, pp. 442–455.
[44] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2011, pp. 1–35.
[45] R. Sun and Z.-Q. Luo, "Guaranteed matrix completion via non-convex factorization," *IEEE Trans. Inf. Theory*, vol. 62, no. 11, pp. 6535–6579, Nov. 2016.
[46] C. Jin, S. M. Kakade, and P. Netrapalli, "Provable efficient online matrix completion via non-convex stochastic gradient descent," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, May 2016, pp. 4520–4528.
[47] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis, "Large-scale matrix factorization with distributed stochastic gradient descent," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2011, p. 69.
[48] H.-F. Yu, C.-J. Hsieh, S. Si, and I. S. Dhillon, "Parallel matrix factorization for recommender systems," *Knowl. Inf. Syst.*, vol. 41, no. 3, pp. 793–819, Dec. 2014.
[49] J. Sabatier, O. P. Agrawal, and J. A. T. MacHado, *Advances in Fractional Calculus*. Dordrecht, The Netherlands: Springer, 2007.
[50] S. Qureshi and A. Atangana, "Mathematical analysis of dengue fever outbreak by novel fractional operators with field data," *Phys. A Stat. Mech. Appl.*, vol. 526, Jul. 2019, Art. no. 121127.
[51] A. Atangana and M. A. Khan, "Validity of fractal derivative to capturing chaotic attractors," *Chaos, Solitons Fractals*, vol. 126, pp. 50–59, Sep. 2019.
[52] A. Atangana and A. Shafiq, "Differential and integral operators with constant fractional order and variable fractional dimension," *Chaos, Solitons Fractals*, vol. 127, pp. 226–243, Oct. 2019.
[53] J. A. T. Machado, "Fractional order description of DNA," *Appl. Math. Model.*, vol. 39, no. 14, pp. 4095–4102, Jul. 2015.
[54] N. I. Chaudhary, Z. A. Khan, S. Zubair, M. A. Z. Raja, and N. Dedovic, "Normalized fractional adaptive methods for nonlinear control autoregressive systems," *Appl. Math. Model.*, vol. 66, pp. 457–471, Feb. 2019.
[55] N. I. Chaudhary, M. Ahmed, Z. A. Khan, S. Zubair, M. A. Z. Raja, and N. Dedovic, "Design of normalized fractional adaptive algorithms for parameter estimation of control autoregressive autoregressive systems," *Appl. Math. Model.*, vol. 55, pp. 698–715, Mar. 2018.
[56] N. I. Chaudhary, S. Zubair, and M. A. Z. Raja, "A new computing approach for power signal modeling using fractional adaptive algorithms," *ISA Trans.*, vol. 68, pp. 189–202, May 2017.
[57] N. I. Chaudhary, M. S. Aslam, D. Baleanu, and M. A. Z. Raja, "Design of sign fractional optimization paradigms for parameter estimation of nonlinear Hammerstein systems," *Neural Comput. Appl.*, pp. 1–19, Jul. 2019, doi: 10.1007/s00521-019-04328-0.
[58] N. I. Chaudhary, S. Zubair, M. S. Aslam, M. A. Z. Raja, and J. A. T. Machado, "Design of momentum fractional LMS for Hammerstein nonlinear system identification with application to electrically stimulated muscle model," *Eur. Phys. J. Plus*, vol. 134, no. 8, p. 407, Aug. 2019.
[59] S. Zubair, N. I. Chaudhary, Z. A. Khan, and W. Wang, "Momentum fractional LMS for power signal parameter estimation," *Signal Process.*, vol. 142, pp. 441–449, Jan. 2018.
[60] M. D. Ortigueira, *Fractional Calculus for Scientists and Engineers*, vol. 84. Dordrecht, The Netherlands: Springer, 2011, p. 154.
[61] F. M. Harper and J. A. Konstan, "The MovieLens datasets," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 1–19, Dec. 2015.
[62] S. Cheng, Y. Wei, Y. Chen, Y. Li, and Y. Wang, "An innovative fractional order LMS based on variable initial value and gradient order," *Signal Process.*, vol. 133, pp. 260–269, Apr. 2017.
[63] K. Li, X. Zhou, F. Lin, W. Zeng, and G. Alterovitz, "Deep probabilistic matrix factorization framework for online collaborative filtering," *IEEE Access*, vol. 7, pp. 56117–56128, 2019.
[64] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proc. 10th ACM Conf. Recommender Syst. (RecSys)*, 2016, pp. 233–240.

ZESHAN ASLAM KHAN received the B.E. degree from the University of Engineering and Technology, Peshawar, Pakistan, in 2005, and the M.S. degree from Halmstad University, Sweden, in 2010, all in computer systems. He is currently pursuing the Ph.D. degree from International Islamic University, Islamabad, Pakistan. His areas of interests are embedded systems, recommender systems, and machine learning.

SYED ZUBAIR received the B.Sc. and M.Sc. degrees from the University of Engineering and Technology, Lahore, in 2002 and 2007, respectively, all in electrical engineering, and the Ph.D. degree in sparse signal processing and machine learning from the University of Surrey, Guildford, U.K., in 2014. He worked at different national defence Research and Development organizations. He has applied his expertise in the field of recommender systems, speech processing, control systems, language pronunciation detection, and EMG and MRI signal reconstruction. He is currently a Founder of Analytics Camp, a machine-learning technology start-up that provides hands-on training of machine learning tools and technologies. He is also a Professor (Adjunct) with the Faculty of Computing & IT, University of Sialkot, Pakistan. His areas of interests are machine learning, sparse coding, deep learning, and speech and audio processing.

HANI ALQUHAYZ received the bachelor's degree in computer science, the master's degree in information systems management, and the Ph.D. degree in computer science from De Montfort University, U.K. He is currently an Assistant Professor with Majmaah University, Saudi Arabia. He is also an Assistant Professor with the Computer Science Department, College of Sciences, Majmaah University. His research interests are in wireless security, scheduling, and image processing.

MUHAMMAD AZEEM received the master's degree in information technology from Quaid-i-Azam University, Islamabad, Pakistan, and the Ph.D. degree in computer science from the Beijing University of Technology, China. He is currently working as an Assistant Professor and the Head of Department of Information Technology, Faculty of Computing and IT, University of Sialkot. He received special contribution award, science and technology award during Ph.D. and Chancellor and Vice Chancellors' medal by President of Pakistan in the master studies. He has worked in leading technology companies of Pakistan as a Software Design Engineer. His research interests include information security, steganography, cryptography, wireless sensor networks, machine learning, mobile, desktop, and web applications development extended with software engineering.

ALLAH DITTA received the M.Sc. degree in computer science and technology from Quaid-i-Azam University (Q.A.U), Islamabad, Pakistan, in 2012, and the Ph.D. degree in computer science and technology from the College of Computer Science, Beijing University of Technology, China, in 2017. In 2017, he joined the Division of Science and Technology, University of Education, Lahore, Pakistan, where he is working as an Assistant Professor. His research interests include information security, steganography, cryptography, network security protocols, and wireless sensor networks.