



SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Hybrid Novel Bat Twin Bounded Support Vector Machine Based Personalized Movie Recommendation System

Nisha Bhalse¹, Ramesh Thakur², Archana Thakur³

¹Institute of Engineering & Technology, DAVV, Indore, India, IPS Academy, Institute of Engineering & Science, Indore, India.

E-mail: nishabhalse@ipsacademy.org

²International Institute of Professional Studies, DAVV, Indore, India. E-mail: rameshthakur@iips.edu.in

³School of Computer Science & IT, DAVV, Indore, India. E-mail: archanathakur.scs@dauniv.ac.in

ABSTRACT

The growing demand for personalized content delivery has made recommendation systems a fundamental component in online streaming platforms. The user experience has been significantly enhanced by personalized movie recommendation systems that suggest relevant movies based on individual preferences. In this paper a collaborative filtering based personalized movie recommendation system is proposed that incorporates the novel bat algorithm (NBA) and twin bounded support vector machine (TBSVM), termed hybrid novel bat twin bounded support vector machine (HNBTBSVM). Traditional recommendation approaches are collaborative filtering (CF) and content-based filtering (CB), which often face limitations such as scalability, sparsity, and cold start problems, which limit the system's ability to provide recommendations for new users or items due to insufficient data. A support vector machine (SVM) is a classifier that discriminates between user preferences. The proposed HNBTBSVM model addresses the cold start issue by combining NBA optimization with TBSVM classification. The hybrid model improves classification accuracy and scalability in sparse data environments, a characteristic of cold-start situations. The TBSVM dual-boundary approach effectively handles data sparsity, enhancing boundary flexibility and classification accuracy with limited data. Additionally, similarity measures align new users or items with existing profiles, while augmented data enriches user interactions. Experimental results of the Movielens dataset achieving the accuracy of TBSVM is 85.3%, standard bat algorithm with twin bounded support vector machine (BATBSVM) is 88.3%, contraction factor and dynamic adaptive inertia weight particle swarm optimization support vector machine (CF-IWA PSO-SVM) model is 79.6%, linear kernel and Gaussian kernel HNBTBSVM models are 90.70% and 93.45%, with marked improvements in accuracy, personalization, and cold start mitigation, and reduced the root mean square error (RMSE) and mean absolute error (MAE).

Keywords: Recommendation system; Collaborative filtering; Classification; Support vector machine; optimization.

1. Introduction

The growing demand for personalized content delivery has made recommendation systems a fundamental component of online streaming platforms, due to information overload [1]. The sheer amount of information available can be overwhelming, and filtering out useful information online can be difficult. The information overload problem is overwhelming users in finding relevant and useful data on the internet [2]. A personalized movie recommendation system is essential for enhancing the user experience because it provides relevant movies based on to users' individual preferences [3-4]. These systems are widely used across a wide range of e-commerce platforms such as amazon, Facebook, movie lens, Netflix, and Spotify [5]. Figure 1 depicts the types of recommendation systems diagrammatically [6].

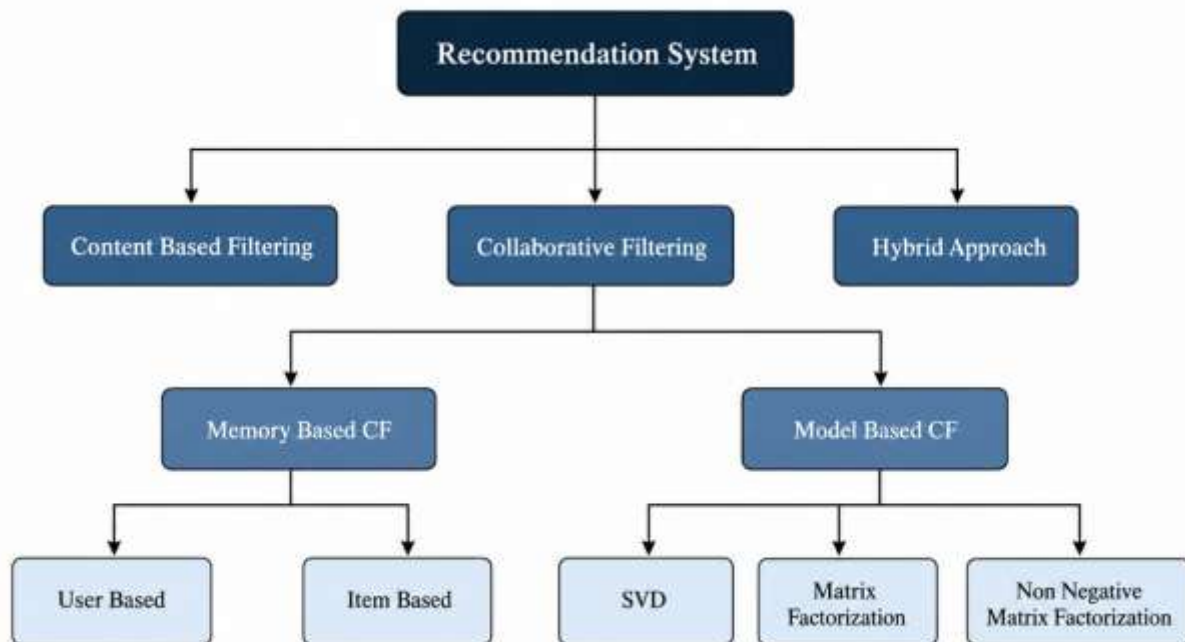


Figure 1. Various types of recommendation system

Content based filtering (CB) generates recommendations based on the user's profile that has previously shown interest in the content of items [7]. Despite several methods, collaborative filtering (CF) is widely used for user item interaction patterns without requiring item features. The machine learning based hybrid approaches, support vector machine (SVM), and its variants, twin SVM, and twin bounded support vector machine (TBSVM), are improving forecast reliability by extracting complex nonlinear interactions in user preference data [8-9]. Twin SVM increases computing efficiency by tackling two smaller optimization issues rather than a single large one [10]. The bounded SVM framework improves robustness by confining decision outputs to a predefined range, hence minimizing sensitivity to noise and outliers [11]. Twin SVM generates the two smaller optimization problems rather than a single large problem, increasing the efficiency of the recommendation task [12]. Bounded SVM increases the formulated robustness of the model by restricting function outputs within a predefined margin. It is reducing the sensitivity of unbalanced and noisy data [13]. In high-dimensional learning problems, the novel bat algorithm (NBA) has demonstrated a strong capacity to tune model hyper parameters while avoiding local minima and premature convergence [14-15]. Further research emphasizes the need for advanced and hybrid recommendation architectures through evidence that incorporating CF with optimized SVM-based frameworks dramatically enhances recommendation accuracy and minimizes root mean square error (RMSE) and mean absolute error (MAE).

The CF faces several challenges, such as data sparsity, which diminishes similarity accuracy and decreases recommendation prediction performance [16-17]. Due to the cold-start issue caused by a lack of interaction history, new users are not recommended accurate movies, and new movies are unable to effectively recommend to users [18]. Additionally, CF struggles to model nonlinear user-item relationships and remains sensitive to noise and imbalanced rating patterns, which may bias recommendations [19-20].

The main objective of the paper is to develop a hybrid novel bat twin bounded support vector machine (HNBTTBSVM) movie recommendation system incorporating NBA and TBSVM with CF that improves the accuracy, efficiency, and user satisfaction of movie recommendations. The previous researches worked in a traditional CF based system address the issues, including data sparsity, nonlinear rating patterns, and noise sensitivity. Additionally, the proposed HNBTTBSVM model improves the prediction recommendation of movie performance by efficiently tuning hyper parameters through the meta-heuristic approach and prediction error.

This paper's main contribution is the development of an HNBTTBSVM model that incorporates CF to increase the accuracy and reliability of movie recommendations. The proposed HNBTTBSVM model addresses the sparsity issue, nonlinear rating patterns, and noise sensitivity by introducing an NBA meta heuristic optimization technique for automatically adjusting SVM parameters, compared with TBSVM, standard bat algorithm with twin bounded support vector machine (BATBSVM), contraction factor and dynamic adaptive inertia weight particle swarm optimization support vector machine (CF-IWA PSO-SVM). While the NBA enhances global search capabilities and avoids premature convergence, incorporating the bounded twin formulation provides stable decision boundaries. The proposed

HNBTBSVM model's experimental results significantly enhance accuracy, reduce RMSE and MAE, and improve overall recommendation accuracy compared to existing CF and SVM-based models.

The structure of the paper is as follows: The literature review is presented in section 2. The methods and proposed methodology of the HNBTBSVM model are covered in section 3. The experimental results will be provided and elaborated in section 4. Section 5 includes the conclusion and further research.

2. Literature Review

In the literature review, CF keeps going to be the leading technique for movie recommendations as it uses user-item behaviour instead of item information, which has several benchmark studies and techniques employing CF as the standard for evaluation. Miao et al. [21] designed and developed a movie recommendation system that integrates CF with a big data platform, improving the accuracy and scalability of the system, but data scalability and cold start are challenges. Gebreselassie et al. [22] reviewed particle swarm optimization, evolutionary approaches, genetic algorithms, and ant colony optimization that are used with CF algorithm, improving the performance of sparsity and RMSE, while hampering the ability to provide accurate recommendations.

Kouchi et al. [23] proposed a model that uses the bee colony optimization algorithm with user-based CF, enhancing the precision and recall performance, and computational cost is low, the number of recommendations and feature parameters must be identified empirically. Singh et al. [24] designed and demonstrated the CF recommendation system incorporating the grasshopper optimization algorithm, enhanced precision and recall, but did not explicate the issues of cold-start and scalability. Khedairia et al. [25] designed an iterative combining clustering method that combined multiple clustering algorithms. It is improving the robustness, novelty performance, but it is time consuming due to the iterative voting and re-clustering.

Li et al. [26] proposed a hybrid movie recommendations system that combined a convolutional multimodal auto multilayer graph with alternating least squares CF. It provides enhanced accuracy, but also presents the challenge of the cold-start issue for new users or movies. Jayalakshmi et al. [27] presented a comparative study of deep learning methods and hybrid systems. Hybrid systems performances in terms of accuracy and adaptability are good, but scalability, cold-start, and data sparsity are challenges. Meghana et al. [28] showed the comparative analysis on matrix factorization, ensemble methods, and hybrid methods using the Netflix dataset, reducing RMSE on rating prediction, but faced challenges of cold start, sparsity, and scalability.

Singh et al. [29] designed a matrix factorization recommendation system, which, despite its excellent results of MAE and RMSE, has prevalent challenges, such as data sparsity and cold start issues. Chen et al. [30] proposed a recommendation system that integrated CF with SVM, which improved prediction accuracy and recommendation relevancy. Scalability is an issue in real-time environments. Mu et al. [31] demonstrated a deep learning architecture that utilizes multimodal data to generate a personalized movie recommendation system. The model extracts hidden properties from various data types. Reduced sparsity, improved recommendations, but did not completely resolve cold start issues. Leander et al. [32] introduced a movie recommendation system that combines a CB approach with SVM, good accuracy, and a lack of diversity, novelty, and serendipity.

Walek et al. [33] proposed a hybrid recommendation system that combines three subsystems: fuzzy expert system, content-based filtering, and collaborative filtering. Precision, recall, and F1-score are satisfactory, but diversity and novelty are inadequate. Cai et al. [34] developed an inductive heterogeneous graph neural network recommendation system that uses a neural network to solve the user cold-start issue, which does not work for serendipity. Harte et al. [35] developed a sequential recommendation system by using the large language models (LLMs), which uses the augmented ID embeddings for recommending the items. Enhance normalized discounted cumulative gain, but increasing storage costs is also a challenge.

The reviewed literature highlights that the incorporation of hybrid models has led to a significant advancement in recommendation systems and enhanced interpretability and more accurate recommendations. But the main challenges of accuracy, sparsity, and scalability remain the same. The proposed HNBTBSVM movie recommendation system demonstrates significant potential as an effective solution for recommending personalized movie recommendations. By leveraging the non-parallel twin support vector machine approach, this offers an advanced alternative to traditional recommendation algorithms. The proposed HNBTBSVM model solved the key challenges such as data sparsity, scalability, and accurate recommendations. The proposed model improved predictive performance and robustness over other existing methods.

3. Materials and Methods

Various methodologies are presented to implement the HNBTBSVM. In this part, the fundamentals of the SVM and twin SVM algorithms are discussed. Therefore, the reader can understand how the concepts of SVM and twin SVM have evolved [36].

3.1. Proposed Methodology

3.1.1 Support vector machine

The input space is $x_i \in R^n$ ($i = 1, 2, 3, \dots, m$) and the corresponding output class is y_i . Here R^n is the real vector space of n dimensions. For a particular dataset specified by Equation (1), R represents the set of real numbers. The input vector of class +1 with m_1 patterns is $X^+ \in R^{m_1 \times n}$, and the class -1 with m_2 patterns is $X^- \in R^{m_2 \times n}$. Matrix $Y^+ \in R^{m_1 \times n}$ symbolizes the output of the m_1 patterns of class +1, and matrix $Y^- \in R^{m_2 \times n}$ symbolizes the output of the m_2 patterns of class -1.

$$(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m) \in X, Y \quad (1)$$

The linear SVM solves the classification problem by finding the parallel optimal hyperplane. In binary SVM classification, the decision hyperplane is located midway between two supporting hyperplanes. These hyperplanes represent the two class patterns respectively; they are placed as far from the opposite class as possible while remaining in close proximity to their own class. A new pattern is assigned to the class whose separating hyperplane it is nearest to. Equation (2) represents the hyperplane separation formula:

$$f(x) = (w^T \cdot x) + b = 0 \quad (2)$$

where w is the inertia weight vector, $w \in R^n$, $b \in R$, and the pre-specified threshold is denoted by b . To measure the empirical risk and determine the decision hyperplane together with its values of w and b , the following optimization problem is used. Equation (3) represents the linear optimization equation of structural risk minimization for computing w and b :

$$\min w, b, \xi \quad 1/2 \|w\|^2 + C \sum \xi_i \quad \text{s.t.} \quad y_i((w \cdot x_i) + b) \geq 1 - \xi_i \quad (3)$$

for $i = 1 \dots m$, $\xi_i \geq 0$

where C is the punishment coefficient and ξ_i is the slack variable that ensures the constraint condition is satisfied. C is a user-defined parameter that balances the machine's complexity against the non-separable data points.

Equation (4) is a decision function that assigns the class of a new pattern. It returns a value of 0 when the class is -1; otherwise it returns 1 when the class is +1:

$$f(x) = \text{sgn}((w^T \cdot x) + b) \quad (4)$$

In nonlinear classification, kernel-generated surfaces are considered instead of hyperplanes, while the remainder is the same. Equation (5) represents the Gaussian kernel function. Mathematically:

$$f(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / 2\sigma^2), \sigma > 0 \quad (5)$$

3.1.2 Twin support vector machine

TWSVM is a machine learning classification technique [37]. In the n -dimensional real space R^n , it has two non-parallel hyperplanes, which are as far from each other as possible while being closer to one of the two classes. Mathematically, Equations (6)–(7) represent the two non-parallel hyperplane equations:

$$f_+(x) = w^+ \cdot x + b^+ = 0 \quad (6)$$

$$f_-(x) = (w^- \cdot x + b^- = 0) \quad (7)$$

Equations (8)–(9) are used for finding the hyperplanes; two quadratic programming problems are formulated to obtain these values.

$$\begin{aligned} &\text{minimize } w^+, b^+, \xi^- \quad 1/2 \|X^+ w^+ + e^+ + b^+\|^2 + C_1 e^+ T \xi^-, \\ &\text{s.t. } (X^- w^+ + e^+ - b^+) + \xi^- \geq e^-, \xi^- \geq 0 \end{aligned} \quad (8)$$

And

$$\begin{aligned} &\text{minimize } w^-, b^-, \xi^+ \quad 1/2 \|X^- w^- + e^- - b^-\|^2 + C_2 e^- T \xi^+, \\ &\text{s.t. } (X^+ w^- + e^- + b^-) + \xi^+ \geq e^+, \xi^+ \geq 0 \end{aligned} \quad (9)$$

$$\text{class}_i = \arg \min_k = +, - |w^+ \cdot x_i + b^+| / \|w^+\| \quad (10)$$

Here, ξ is the slack variable, and $\xi^+ = (\xi_1, \dots, \xi_{m_1})$ and $\xi^- = (\xi_1, \dots, \xi_{m_2})$ for the classes. The user-defined penalty parameters are C_1 and C_2 . The consistent dimension vectors are e^- and e^+ , and the two norms are denoted by $\|\cdot\|$. b^- & $b^+ \in R$, w^- & $w^+ \in R^n$. Equations (8)–(9) are two hyperplanes, and the new pattern's class is determined by which of the two hyperplanes an item is closest to. The decision is made and a class is assigned using Equation (10), where the absolute value is denoted by $|\cdot|$.

3.1.3 Bat Algorithm

Xin-She Yang invented the bio-inspired bat algorithm in 2010. The bats use echolocation, which involves loudness and changing pulse rates of emission, serving as inspiration. The following fundamental steps are illustrated, based on some approximations and idealizations [38–39].

Step 1: The virtual bats N are initialized, and Equation (11) represents their velocities v_i and positions x_i in the search space. The N bats' velocity is zero or another low value.

$$x_i, j = x_j \text{Min} + (x_j \text{Max} - x_j \text{Min}) * \text{rand}(0,1) \quad (11)$$

Here, $i \in [1, 2, 3, \dots, N]$, $j \in [1, 2, 3, \dots, D]$ and $\text{rand}(0,1)$ are random vectors. The domain problem size depends on the values of $x_j \text{Min}$ and $x_j \text{Max}$.

Step 2: New solutions are generated. Equations (12)–(14) simulate bats flying toward promising areas in the search space.

$$f_i = f_{\min} + (f_{\max} - f_{\min}) \times \beta \quad (12)$$

$$v_{it+1} = v_{it} + (x_{it} - x^*) \times f_i \quad (13)$$

$$x_{it+1} = x_{it} + v_{it+1} \quad (14)$$

Where $\beta \in [0,1]$ is a random vector, x^* is the current global best solution, and the domain size of the problem depends on the frequency $f_{\min}$ and $f_{\max}$.

Step 3: The local search is invoked, which is activated if a large number of random numbers are compared to the pulse emission rate r . Equation (15) shows the best solution around a random walk.

If $\text{rand}(0,1) > r_i$

$$x_{\text{new}} = x_{\text{old}} + \varepsilon * A_{\text{meant}} \quad (15)$$

Where $\varepsilon \in [-1,1]$ is a random vector from a uniform distribution. All bats' average loudness at time-step t is A_{meant} , and the rate of pulse $r_i \in [0,1]$.

Step 4: The solutions are updated randomly. Equations (16)–(18) represent the updated accepted solution of position, loudness, and pulse rate.

If $\text{rand}(0,1) > A_i$ && $f(x_i) < f(x)$

$$f(x) = f(x_i) \quad (16)$$

$$A_{it+1} = \alpha A_{it} \quad (17)$$

$$r_{it+1} = r_{i0}(1 - e^{-\gamma t}) \quad (18)$$

The constants are α ($\alpha \in [0,1]$) and γ ($\gamma > 0$). The initial loudness A_{i0} and pulse rate r_{i0} are uniformly distributed in the intervals $[1,2]$ and $[0,1]$, with $\alpha = \gamma = 0.9$.

Step 5: The global best solution is updated.

3.1.4. The NBA

The NBA is an enhanced version of the original bat algorithm [40], which aims to be a better optimization algorithm. The virtual bats are represented in the search space by their position, velocity, and prey in D-dimensional space. The trajectory of virtual bats is determined by classical mechanics. In the NBA, the doppler effect is incorporated. Due to quantum behaviour, the virtual bats appear in the search space in any position. Two additional idealised rules must be formulated.

- A stochastic selection allows all bats to forage.
- The doppler effect in echoes can be compensated for by all bats. They have the ability to adaptively change the compensation rate based on how close they are to their targets.

The position of all N virtual bats is $X_{i,j,t}$ ($i \in 1,2,\dots,N, j \in 1,2,\dots,N$, velocity $v_{i,j,t}$ at time t step, food search in a D-dimensional space. Equation 19 finds the best parameter θ that minimizes the objective function(θ).

$$\theta_{\min} F\theta = \text{Accuracy} \text{val} \theta + \lambda \cdot \|\theta\|^2 \quad (19)$$

Bats have quantum behaviour. If one bat finds a food location, then others will start feeding there as well. g_{jt} shows the entirety of the best position in the bat swarm, they could serve as an attractor. The quantum behaviour of virtual bats positions are expresses in Equation 20.

$$x_{i,j,t+1} = g_{jt} + \theta * \text{mean}_{jt} - x_{i,j,t} \ln 1/u_{i,j}, \text{ if } \text{rand}_{j0,1} < 0.5, g_{jt} - \theta * \text{mean}_{jt} - x_{i,j,t} \ln 1/u_{i,j}, \text{ otherwise,} \quad (20)$$

Here, $\text{mean}_{jt} = 1/N * \sum x_{i,j,t}$, u is uniformly distributed in $(0,1)$.

For the doppler effect, updating the offspring at time step t $x_{i,j,t+1}$ and $v_{i,j,t+1}$. The bats' compensation rates are dependent on frequency, and they have three parts. The value of frequency is between $f_{\min}$ and $f_{\max}$. The global best solution g_{jt} . $\lambda_i f_i$ is product of velocity and appropriate frequency f_i . For the movements of the bats, inertia weight w additional parameters are added for velocity updates. Equations 21-24 are expressed the frequency update, doppler compensated frequency, velocity update and position update.

$$f_{i,j} = f_{\min} + f_{\max} - f_{\min} * \text{rand}_{0,1} \quad (21)$$

$$f'_{i,j} = (c + v_{i,j,t})c + v_{g,jt} * f_{i,j} * 1 + C_i * g_{jt} - x_{i,j,t} g_{jt} + x_{i,j,t} + \varepsilon \quad (22)$$

$$v_{i,j,t+1} = w * v_{i,j,t} + g_{jt} - x_{i,j,t} * f_{i,j} \quad (23)$$

$$x_{i,j,t+1} = x_{i,j,t} + v_{i,j,t} \quad (24)$$

Here, a uniform random vector is $w \in [0,1]$. is the smallest constant that is used to avoid zero-division error. The range of C is $C \in [0,1]$. One indicates that the bat has fully compensated for doppler effect in echoes, whereas zero indicates that it is unable to do so. c ($c = 340$ m/s) is the speed in air, $v_{g,jt}$ is the speed corresponding to the global best position. Each solution $x_{it} = [C1, C2, C3, C4, C5]$ are evaluated via the fitness function $f(\cdot)$ and the best parameter are updated accordingly. Equation 25-26 represent each bat new position using the gaussian perturbation around the global best.

If $\text{rand}_{0,1} > r_i$

$$x_{i,j,t+1} = g_{jt} * 1 + \text{rand}_{n0,\sigma2} \quad (25)$$

$$\sigma2 = A_{it} - A_{\text{meant}} + \varepsilon \quad (26)$$

Where, $\text{rand}_{n0,\sigma2}$ is a gaussian distribution with mean 0 and standard deviation $\sigma2$. ε is used to ensure $\sigma2 > 0$. The

average loudness of bats is A_{mean} at time step t .

Equation 27 represents the loudness updates.

$\omega_{it+1} = \alpha \omega_{it}$, α is a constant

(27)

Equation 28 represents the pulse rate update.

$r_{it+1} = r_{i0} \cdot \exp(-\gamma t)$, γ is a constant

(28)

When, over a number of time steps (G), all bats are unable to locate a better prey than the one they found earlier, this occurs repeatedly.

3.2 The Proposed HNBTSVM Movie Recommendation System

The HNBTSVM based personalized movie recommendation system integrates the NBA with TSVM. The NBA optimizes the TSVM parameters to enhance recommendation accuracy.

Our proposed model consists of two parts:

- The processed dataset from Movie lens.
- HNBTSVM algorithm

3.2.1 HNBTSVM movie recommendation system

The NBA is combined with TSVM, which is a powerful classifier. Then consider TSVM again, it has four user-defined parameters that need to be tuned for its particular application. By adopting the methodology from [20] the classification accuracy of TSVM is given by:

$$A = \frac{Tp + Tn}{Tp + Fp + Tn + Fn}$$

(29)

Where, Tp = number of positive patterns decided to be positive

Tn = number of negative patterns decided to be negative,

Fp = number of negative patterns decided to be positive,

Fn = number of positive patterns decided to be negative.

The accuracy of TSVM by tuning C_1, C_2, C_3 , and C_4 for employed Movielens dataset.

$x = (C_1, C_2, C_3, C_4)$ in the linear kernel HNBTSVM Model.

$x = (C_1, C_2, C_3, C_4, \mu)$ in the gaussian kernel HNBTSVM model.

Figure 2 depicts the flowchart of the HNBTSVM model.

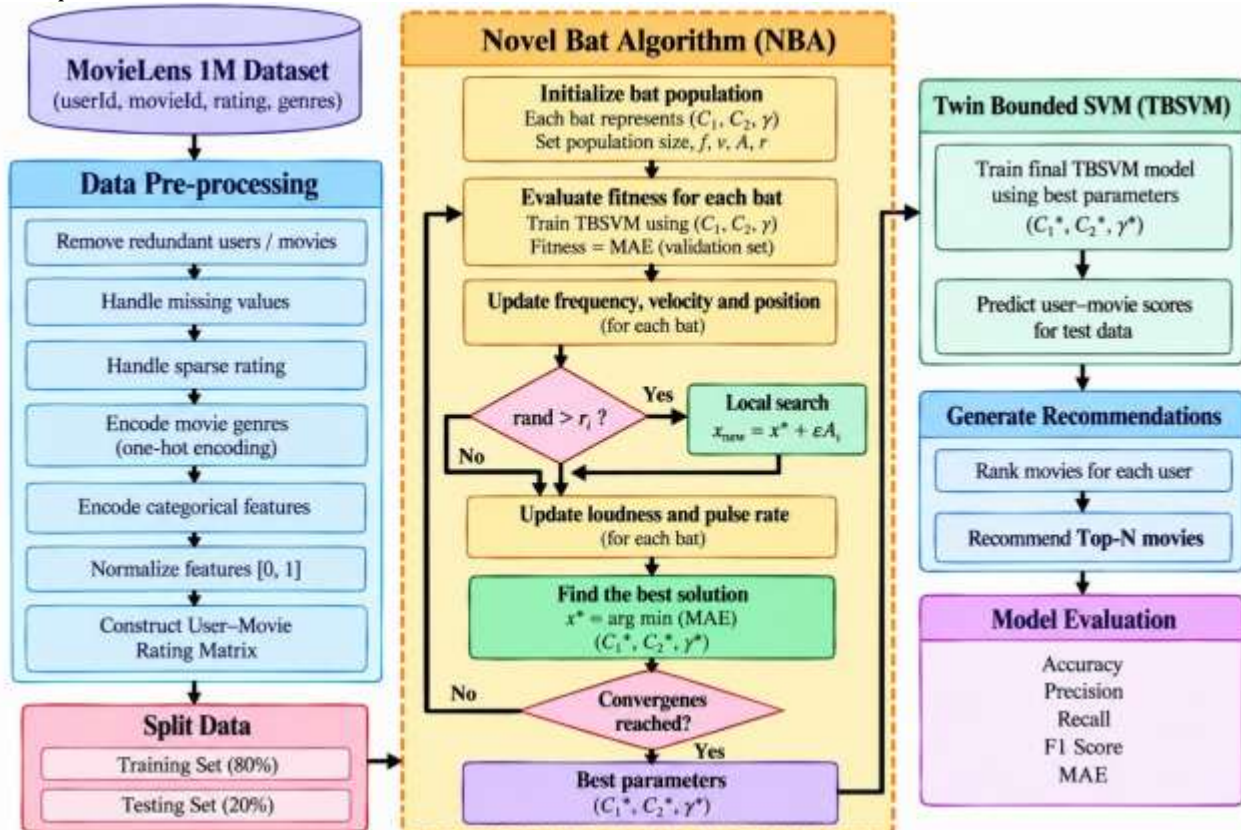


Figure 2. HNBTSVM model flowchart

Moreover, $x \in \mathbb{R}$ in both of the cases i.e., linear and nonlinear. The proposed model HNBTSVM algorithm is shown as algorithm 1.

```

INPUT-User-Movie rating matrix
OUTPUT-Top-K movie recommend
Step 1: Load the user-item rating matrix  $R=[r_{u,i}]^{m \times n}$ 
Step 2: Normalize the rating matrix
Step 3: Apply dimensionality reduction using singular value decomposition
 $R'=U_k \Sigma_k V_k^T$ 
Step 4: Construct user feature vectors
 $x_u=U_k(u,:)$ 
Step 5: Construct movie feature vectors
 $x_i=V_k(i,:)$ 
Step 6: Form user-movie combined feature vectors
 $x_{u,i}=[x_u, x_i]$ 
Step 7: Initialize Novel Bat Algorithm population with random solutions (positions, velocities, parameters)
 $x_{i,j}=x_j \text{Min}+x_j \text{Max}-x_j \text{Min} * \text{rand}(0,1)$ 
Step 8: Define fitness function based on prediction error
 $\theta \text{minF}\theta = \text{Accuracyval}\theta + \lambda \cdot \|\theta\|^2$ 
Step 9: Update bat positions, velocities, frequency, loudness, and pulse rate
 $f_{i,j}=f_{\text{min}}+f_{\text{max}}-f_{\text{min}} * \text{rand}(0,1)$ 
 $f_{i,j}=(c+v_{i,j})c+v_{g,jt} * f_{i,j} * 1+C_i * g_{jt}-x_{i,jt}g_{jt}+x_{i,jt} \varepsilon$ 
 $v_{i,jt+1}=w * v_{i,jt}+g_{jt}-x_{i,jt} * f_{i,j}$ 
Step 10: Generate new candidate solutions and perform local search.
If  $\text{rand}(0,1) > r_i$ 
 $x_{i,jt+1}=g_{jt} * 1 + \text{rand}(0, \sigma_2)$ 
 $\sigma_2 = \text{Ait} - \text{Ameant} + \varepsilon$ 
Step 11: Evaluate fitness of each bat and update global best solution.
 $\omega_{it+1} = \alpha \omega_{it}$ ,  $\alpha$  is a constant
 $r_{it+1} = r_i(1 - \exp(-\gamma t))$ ,  $\gamma$  is a constant
Step 12: Repeat Steps 8–11 until convergence condition is satisfied
Step 13: TWSVM Optimization
 $f+x=w+Tx+b=0$ 
 $f-x=w-Tx+b=0$  minimize  $w, b, \xi, 12\|X+w++e+b\|^2 + C1e-T\xi$ ,
s.t.  $X-w++e-b++\xi \geq e$ ,  $\xi \geq 0$ 
minimize  $w, b, \xi, 12\|X-w--e-b\|^2 + C2e+T\xi$ , s.t.  $X+w--e-b--\xi \geq e$ ,  $\xi \geq 0$ 
Kernel function used
 $K_{xi}, x_j = \exp(-\|x_i - x_j\|^2 / 2\sigma^2)$ 
TWSVM Decision Rule
 $\text{class}_i = \arg \min_k = +, - |w+Tx+b_k| / \|w_k\|$ 
Step 14: Predict ratings for user-movie pairs using optimized TWSVM model
 $\text{Rank}_u = f+x+f-x_2$ 
Step 15: Rank items based on predicted ratings
 $\text{Rank}_u = \text{argsort}(r^{\wedge}_{u,i})$ 
Step 16: Select Top-N movies with highest predicted ratings for each user
 $\text{Top-K}(u) = \text{argtopK}_{i \in \text{Rank}_u}$ 
Step 17: Evaluate performance using metrics such as RMSE, Accuracy, Precision@K

```

3.2.2 Experimental dataset

The proposed HNBTSVM is implemented in MATLAB R2014a running on Intel(R) Core (TM) i3-4005U CPU @ 1.70GHz with 4 GB of RAM on Windows (10 Pro) OS. Table I shows the MovieLens dataset information.

Table 1. MovieLens dataset

Dataset	No. of samples			No. of features	No. of classes
	Movies	Users	Ratings		
MovieLens 1M	3900	6040	1000209	12	2

The evaluation of the proposed HNBTSVM model with traditional methods of recommendation systems, the Movielens 1 M dataset [41] is used. For the HNBTSVM model, hyperparameter settings are described in Table 2. The population size is exploring the solution space effectively in high dimensionality for the recommendation task. The $\gamma = 0.9$ is selected to balance between exploration and exploitation for reducing the loudness. In the prediction model, experiments were performed on a linear kernel function and a non-linear gaussian kernel.

Table 2. Parameter Used

Parameter	HNBTSVM model value	CF-IWA PSO-SVM model value
Population size	500	50
Maximum number of iteration(M)	100	100
alpha (α)	0.99	-
gamma(γ)	0.9	-
Maximum Frequency	1.5	-
Minimum Frequency	0	-
Dim	5	2
Γ_{0Max}	1	-
Γ_{0Min}	0	-
A_{Max}	2	-
G(Growth factor)	10	-
prob _{Max}	.9	-
prob _{Min}	0.6	-
θ_{Max}	1	-
θ_{Min}	0.5	-
W_{Max}	0.9	0.9
W_{Min}	0.5	0.5
C_{Max}	0.9	100
C_{Min}	0.1	0.01

The Movielens dataset is pre-processed and user-movies matrix is generated. It contains the ratings.dat, users.dat, and movies.dat files. Users.dat file in user demographic information, movies.dat in movie information, and ratings.dat contain the users' movie ratings information. All three files combine with the UserID and MovieID into a single table. Table 3 shows the description of the combined dataset. The single table is called the user-rating matrix. Then, the categorical attribute of features is converted into a numerical format. Genres converted into multi-label binarization and label encoding. Ratings are converted into a binary form of likes and dislikes, and then the user-rating matrix is normalised. The new user and new movies have no rating or a very low rating, and sparse matrix uses the singular value decomposition for dimension reduction. Table 4 shows the dataset after pre-processing.

Table 3. Description of the dataset before pre-processing

User ID	Movie ID	Gender	Genres (Multi Class)	Rating (binary)
1	1432	Male	[Thriller, Action, Horror]	4

Table 4. Description of the dataset after pre-processing

User ID	Movie ID	Gender	Genres (Multi Class)	Rating (binary)
1	1432	Male	[1,2,8]	4

The performance analysis of the proposed HNBTSVM model with Movielens 1M dataset. The dataset contains the 3900 movie ratings from 6040 users. Only 2,000 sample data are used. The proposed model first converts the Movielens dataset into numerical data. Then randomly select 10 folds of the dataset as a testing set, and then add to the test dataset. The remaining is used for the training dataset. Figure 3 depicts the flowchart of pre-processing of Movielens dataset.

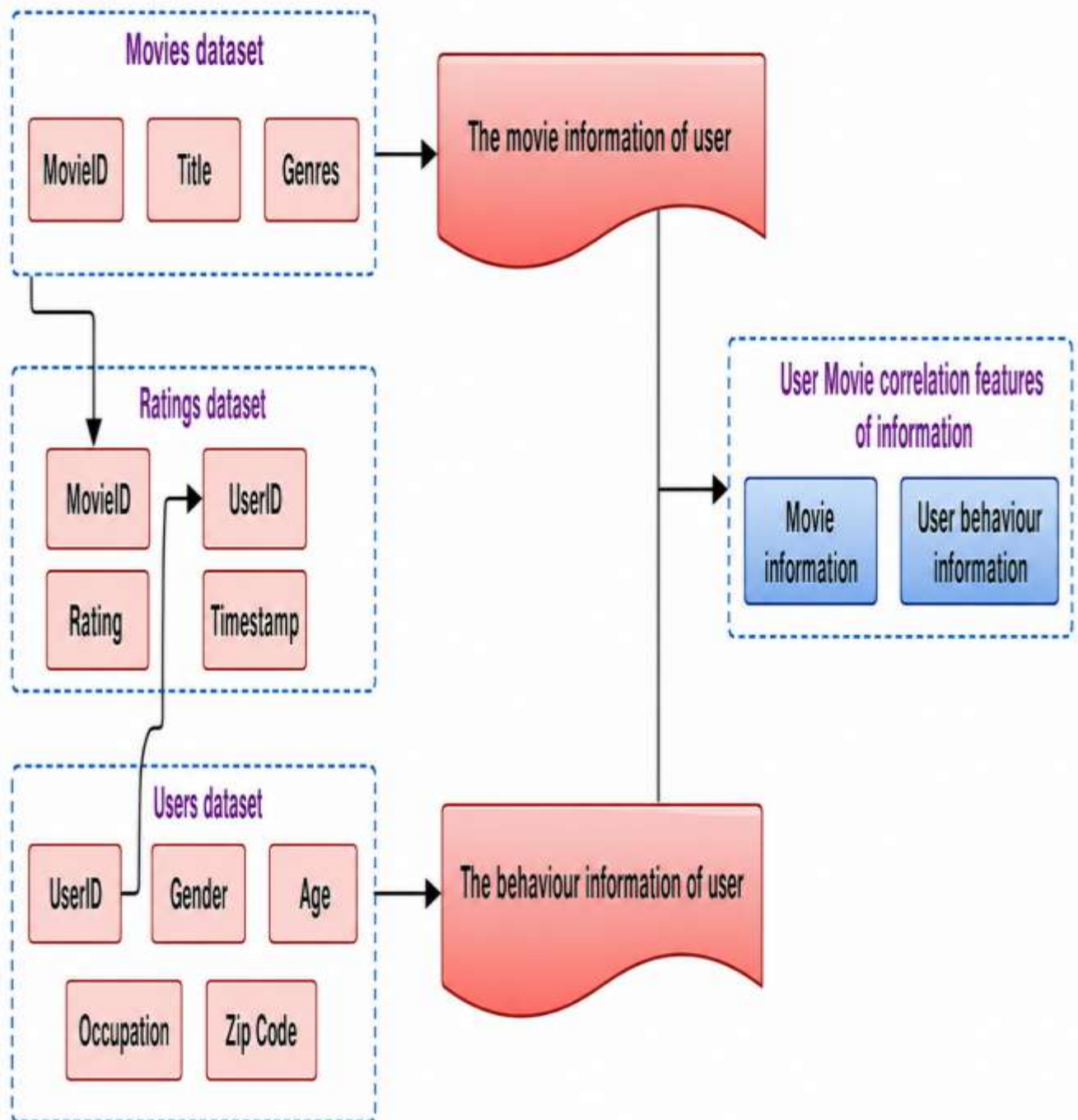


Figure 3. User-Movie feature extraction information

The user demographic information of users alleviates the problem of ‘cold start’. In the proposed HNBTSVM model, user information, movie information, and rating information are used for building the “user-movie” matrix. The classification model concept is based on user ratings: likes and dislikes. The 4- or 5-star rating shows the “like” category the 1, 2- or 3-star rating denotes the “dislike” category. The proposed HNBTSVM model was trained on the user-movie correlation matrix, and optimized TSVM with NBA. Figure 4 depicts the proposed HNBTSVM model for a movie recommendation system. The HNBTSVM model recommends the movie list to the users.

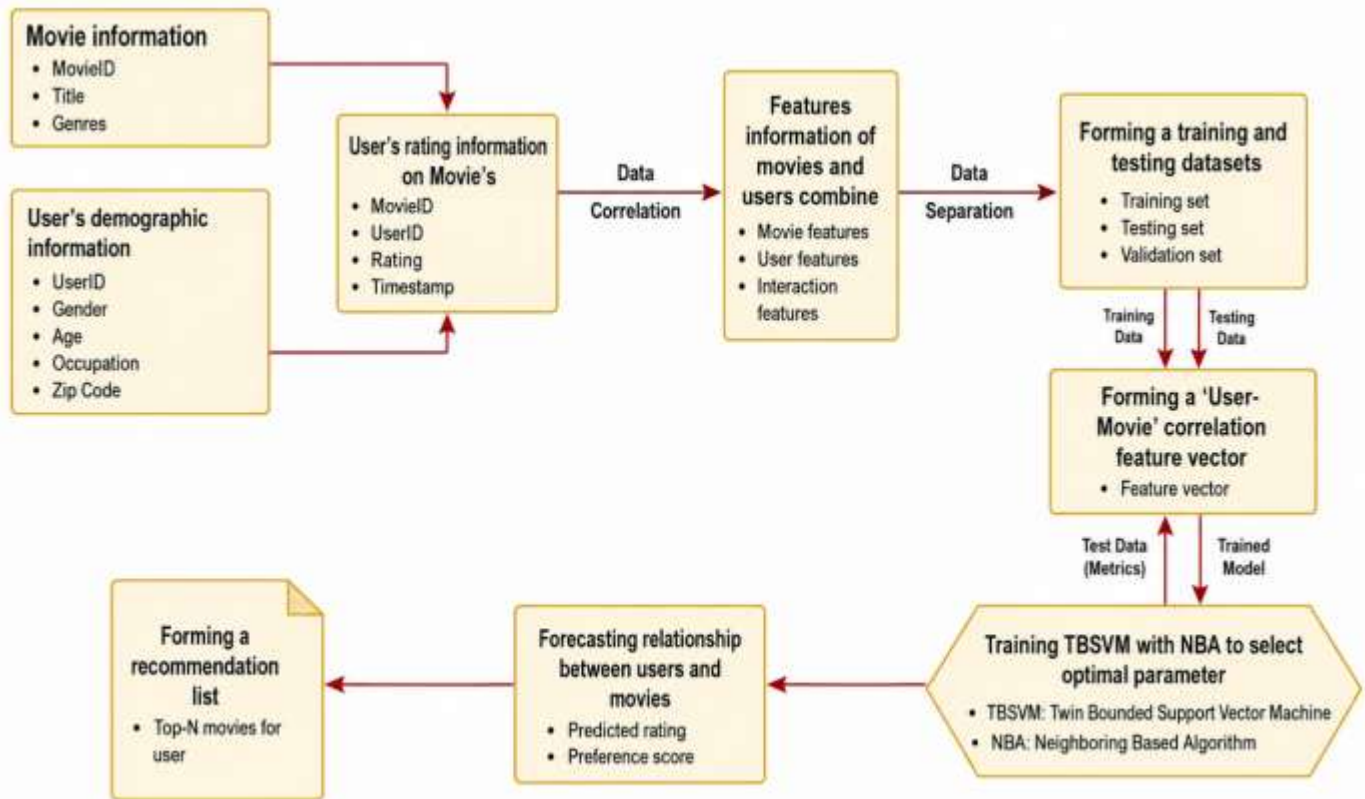


Figure 4. Personalized HNBTSVM model for movie recommendation system

4. Experimental Results & Discussion

The performance evaluation of TBSVM model and the standard bat algorithm with TBSVM is called the BATBSVM model, and TBSVM with a novel bat algorithm is called the HNBTSVM model. The experiments were performed on the Movielens dataset across an 80:20 training and test set. Table 5 shows the performance analysis of different models of TBSVM algorithm.

The proposed HNBTSVM model performance comparison with CF-IWA PSO-SVM movie recommendation system, TBSVM, and BATBSVM models on the Movielens dataset reveals notable differences across various train-test splits. It is implemented in MATLAB [42].

Table 5. shows the performance analysis of different models of TBSVM algorithm

Models\ Metrics	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC-ROC	MAE
TBSVM	85.3	84.1	83.2	83.4	0.85	0.74
BATBSVM	87.3	87.4	85.1	86.2	0.90	0.73

Table 6 shows the average accuracy of the linear kernel HNBTSVM model on the Movielens dataset splits across training & test sets (70:30, 80:20, 90:10, 95:05), along with the parameter values C1, C2, C3, and C4. As the training data percentage rises, model accuracy increases significantly. Starting from an accuracy of 70.43% at the 70:30 split, the model's performance rises to 75.3% at 80:20, 85% at 90:10, and reaches a high of 90.70% at the 95:05 split. The upward trend shows the model's capacity for leveraging more data effectively, and the results depict progressively better performance.

The values of parameters C1 through C4, which regulate different aspects of the model, vary across splits, indicating adaptive parameter adjustments with changes in training data. The model dynamically adjusts the parameter weights based on the available data, with higher training percentages yielding more balanced parameter distributions and enhanced accuracy. The linear kernel HNBTSVM model is particularly effective with larger training sets, where parameter tuning becomes more impactful for maximizing accuracy.

Table 6. Accuracy (%) of linear kernel HNBTSVM model

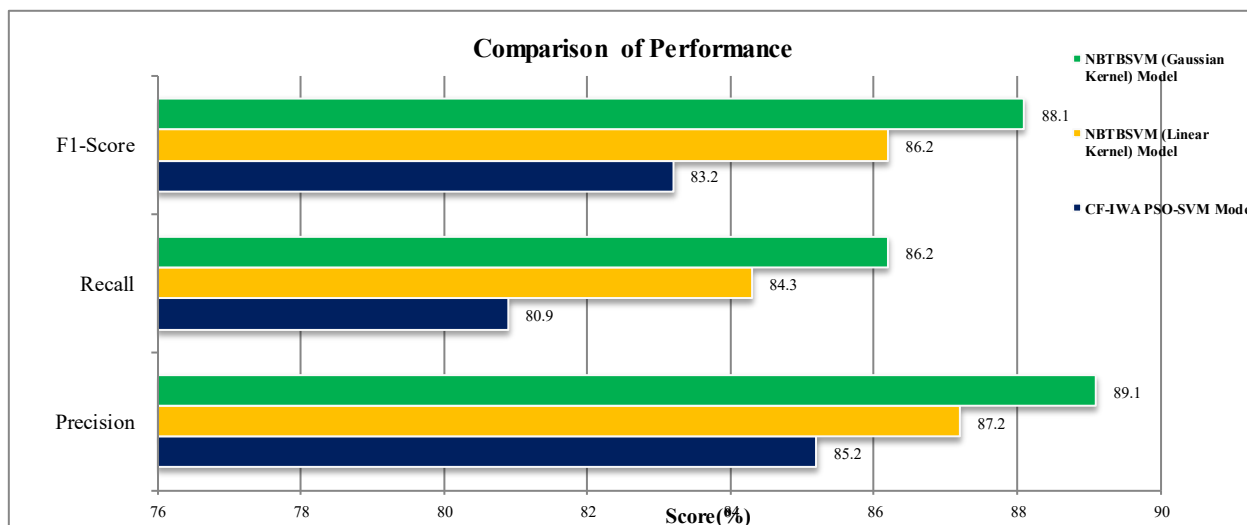
Movielens Dataset	Accuracy	C1	C2	C3	C4
70:30	70.43	0.0000	0.0000	37.4929	9.6563
80:20	75.3	0.5618	0.0043	0.0043	6.6636e ⁻⁰⁴
90:10	85	14.0255	26.0130	8.6815	42.9397
95:05	90.70	0.7842	6.6690e ⁻¹⁰	4.1118 e ⁻¹⁰	33.6884

Table 7 shows the comparison analysis of classification accuracy of CF-IWA PSO-SVM model and the proposed HNBTSVM model on Movielens dataset. It is observed that when training dataset is increased, the CF-IWA PSO-SVM model, linear kernel, and gaussian kernel HNBTSVM model's classification accuracy also increases. The CF-IWA PSO-SVM provides steady accuracy, HNBTSVM models—particularly with the gaussian kernel—excels as training data increases, suggesting they are more effective in scenarios where larger datasets are available, offering higher accuracy in recommendation classification.

Table 7. shows the comparison analysis of the classification accuracy of CF-IWA PSO-SVM & HNBTSVM models

Movielens Dataset	CF-IWA PSO-SVM Model	Linear kernel HNBTSVM Model	Gaussian kernel HNBTSVM Model
70:30	76	70.43	73.78
80:20	76.4	75.3	78.86
90:10	79.6	85	89.20
95:05	79.6	90.70	93.45

Figure 5 & 6 shows the graphical representation of different models with various parameters. The CF-IWA PSO-SVM Model & HNBTSVM Model with different parameters on the Movielens dataset across training and test set 80:20. The performance evaluation of the proposed HNBTSVM model is more accurate as compared to other models in personalized movie recommendation

**Figure 5. Comparative analysis of different models on different parameters.**

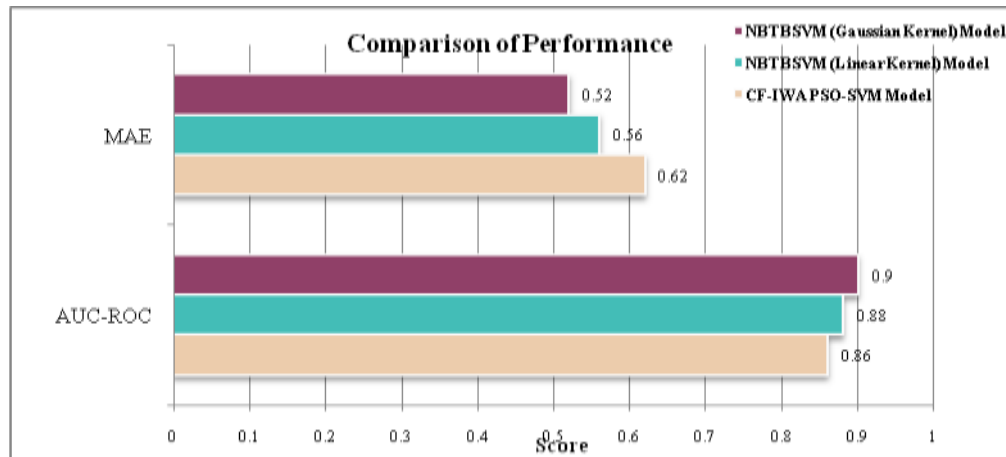


Figure 6. Comparative analysis of different models of MAE & AUC-ROC.

The confusion matrix depicts the HNBTBSVM model performance on a sample of 2000 users, demonstrating its great classification capacity. Table 8 shows the confusion matrix of the HNBTBSVM model. The large number of true positives and true negatives, together with relatively low false predictions, reveal that the HNBTBSVM model effectively distinguishes between relevant and irrelevant recommendations, demonstrating its accuracy and precision in personalised movie recommendation tasks

Table 8. Confusion matrix of the HNBTBSVM model for movies

	Actual relevant movies	Actual irrelevant movies
Predicted relevant movies	TP = 690	FN = 110
Predicted irrelevant movies	FP = 105	TN = 1095

Table 9 shows the computational performance of the proposed HNBTBSVM model and the CF-IWA PSO-SVM model. The time complexity and scalability of the proposed HNBTBSVM model is less, and CPU utilisation is more as compared to the CF-IWA PSO-SVM model for 100 iterations.

Table 9. Computational performance of different models

Metric	HNBTBSVM model	CF-IWA PSO-SVM model
Training Time (seconds)	900	1300
Iterations	100	100
CPU usage	85%	80%
Scalability (time growth)	$O(n)$	$O(n \log n)$

The performance of the proposed personalized movie recommendation HNBTSVM model was evaluated through a series of experiments. The classification capability of the proposed model was evaluated using different parameters and compared with the CF-IWA PSO-SVM model, as well as the TSVM and BATTSVM models.

The three models—CF-IWA PSO-SVM, linear kernel HNBTSVM, and gaussian kernel HNBTSVM—across varying training sample sizes on the Movielens dataset. The CF-IWA PSO-SVM model exhibits stable but limited accuracy, with minimal improvement as the training data increases, plateauing around 79.6%.

The HNBTSVM models—linear and gaussian kernel show significant improvements with larger training sets. The linear kernel HNBTSVM model increases steadily in accuracy, reaching around 90% at the highest training proportion, indicating its effective utilization of more data. The gaussian kernel HNBTSVM model consistently outperforms the other models, achieving a high value of accuracy, recall, precision, AUC-ROC, and F1-score and reducing MAE. Across all splits and reaching approximately 93.5% with 95% training data. The gaussian kernel HNBTSVM captures complex data patterns, enhancing its performance accuracy as training data grows.

The HNBTSVM model in experimental metrics, precision, and recall are the main parameters for addressing the cold start issue. The TSVM dimension size is 5, which specifies the number of selected features utilised for categorisation (e.g., genre, user demographics, movie information). The gaussian kernel was essential for representing complex non-linear relationships between sparse user-item rating matrices. The frequency range (0-1.5) and control parameters $\alpha = 0.99$ and $\gamma = 0.9$ improved the model's adaptability under a sparse matrix. These parameters allowed the HNBTSVM model to efficiently project cold-start users and movies into the learnt preference space and generate relevant recommendations.

5. Conclusion

This work proposed an evolutionary movie recommendation model, the HNBTSVM model that overcomes the limitations of the traditional CF technique. The model uses the users, items, and behaviour information for the construction of a “user-item” matrix. The matrix contains the user’s rating information according to their interest and preferences. The HNBTSVM model is hypothesized to overcome the limitations of existing recommendation methods.

According to experimental results, the proposed HNBTSVM model recommends more accurate movies to the users as compared to the TSVM, BATTSVM and CF-IWA PSO SVM models. The HNBTSVM models, particularly the gaussian kernel variant, outperform the CF-IWA PSO-SVM model in terms of overall performance, and the rate of error is less on the Movielens dataset, especially as the training data proportion increases. The gaussian kernel HNBTSVM model's ability to capture complex, non-linear patterns in the data enables it to achieve the highest accuracy. The linear kernel HNBTSVM also shows promising performance, adapting well to larger training sets and offering a simpler alternative to the gaussian kernel model. In contrast, the CF-IWA PSO-SVM model, while consistent, reaches a plateau in accuracy and does not fully benefit from increased data, suggesting its limitations in scalability.

For future work, there are several avenues to explore. First, optimizing the computational efficiency of the HNBTSVM models, especially the gaussian kernel variant, would make them more practical for large-scale applications. Additionally, investigating hybrid approaches that combine the strengths of CF-IWA PSO-SVM and HNBTSVM models might yield a model that is both robust with smaller datasets and adaptable with larger ones. Future research, further tuning of hyper parameters, and experimenting with other kernel functions may enhance the models' performance, potentially leading to even higher classification accuracy and efficiency.

Conflicts of Interest: The authors have no conflicts of interest to declare.

Funding Statement: No Funding

Acknowledgments: None

Appendix 1

S. No.	Abbreviation	Description
1	BATTSVM	standard bat algorithm with twin bounded support vector machine
2	CB	content based filtering
3	CF	collaborative filtering
4	CF-IWA PSO-SVM)	contraction factor and dynamic adaptively inertia weight particle swarm optimization support vector machine
5	MAE	mean absolute error
6	NBA	novel bat algorithm
7	HNBTSVM	hybrid novel bat twin bounded support vector machine

8	RMSE	root mean square error
9	SVM	support vector machine
10	TBSVM	twin bounded support vector machine

References

- [1] S. Shafiee, "Unveiling the latest trends and advancements in machine learning algorithms for recommender systems: a literature review," *Procedia CIRP*, vol. 121, pp. 115–120, 2024, DOI:10.1016/j.procir.2023.08.062.
- [2] Y. Wang, W. Ma, M. Zhang, Y. Liu, and S. Ma, "A survey on the fairness of recommender systems," *ACM Trans. Inf. Syst.*, vol. 41, no. 3, article 52, pp. 1–43, 2023, DOI:10.1145/3547333.
- [3] M. Furini, "Recommendation systems: issues, challenges and regulations," in *Proc. 2024 IEEE Symp. Computers and Communications (ISCC)*, Paris, France, 2024, pp. 1–6, DOI:10.1109/ISCC61673.2024.10733684.
- [4] Y. Li, K. Liu, R. Satapathy, S. Wang, and E. Cambria, "Recent developments in recommender systems: a survey," *IEEE Comput. Intell. Mag.*, vol. 19, no. 2, pp. 78–95, 2024, DOI:10.1109/MCI.2024.3363984.
- [5] G. Behera and N. Nain, "Collaborative filtering with temporal features for movie recommendation system," *Procedia Comput. Sci.*, vol. 218, pp. 1366–1373, 2023, DOI:10.1016/j.procs.2023.01.115.
- [6] L. Valentine, S. D'Alfonso, and R. Lederman, "Recommender systems for mental health apps: advantages and ethical challenges," *AI & Soc.*, vol. 38, no. 4, pp. 1627–1638, 2023, DOI:10.1007/s00146-021-01322-w.
- [7] N. Alharbe, M. A. Rakrouki, and A. Aljohani, "A collaborative filtering recommendation algorithm based on embedding representation," *Expert Syst. Appl.*, vol. 215, art. no. 119380, 2023.
- [8] Y. Ge and S. Liu, "A survey on trustworthy recommender systems," *ACM Trans. Recommender Syst.*, art. no. 13, 2024, doi: 10.1145/3652891.
- [9] H. Veisi, "Introduction to SVM," in *Learning with Fractional Orthogonal Kernel Classifiers in Support Vector Machines: Theory, Algorithms and Applications*. Singapore: Springer Nature Singapore, 2023, pp. 3–18.
- [10] M. Tanveer, T. Rajani, R. Rastogi, Y.-H. Shao, and M. A. Ganaie, "Comprehensive review on twin support vector machines," *Ann. Oper. Res.*, vol. 339, no. 3, pp. 1223–1268, 2024.
- [11] Y. Li and W. Yan, "Twin support vector regression with privileged information," in *Proc. CCF Conf. Big Data*, Singapore: Springer Nature Singapore, Sep. 2023, pp. 125–135.
- [12] K.-L. Du, B. Jiang, J. Lu, J. Hua, and M. N. S. Swamy, "Exploring kernel machines and support vector machines: Principles, techniques, and future directions," *Mathematics*, vol. 12, no. 24, art. no. 3935, 2024.
- [13] W. Hu, S. Mao, M. Liu, M. Dong, Y. Zhang, and T. Liu, "Two twin extreme learning machines for regression and their applications in industry," in *Proc. IECON 2023 - 49th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2023, pp. 1–6.
- [14] B. Alsalihi, L. Abualigah, and A. T. Khader, "A novel bat algorithm with dynamic membrane structure for optimization problems," *Appl. Intell.*, vol. 51, no. 4, pp. 1992–2017, 2021.
- [15] X.-B. Meng, X.-Z. Gao, Y. Liu, and H. Zhang, "A novel bat algorithm with habitat selection and Doppler effect in echoes for optimization," *Expert Syst. Appl.*, vol. 42, no. 17–18, pp. 6350–6364, 2015.
- [16] P. Prem Kumar, N. U. N. M. M., and H. V., "Movie recommender system: Addressing scalability and cold start problems," in *Proc. Int. Conf. Intell. Comput. Optim.*, Cham, Switzerland: Springer Nature Switzerland, Apr. 2023, pp. 155–164.
- [17] G. Behera, N. Nain, and R. K. Soni, "Integrating user-side information into matrix factorization to address data sparsity of collaborative filtering," *Multimedia Syst.*, vol. 30, no. 2, art. no. 64, 2024.
- [18] R. J. Kuo and S. S. Li, "Applying particle swarm optimization algorithm-based collaborative filtering recommender system considering rating and review," *Appl. Soft Comput.*, vol. 135, art. no. 110038, 2023.
- [19] X. Wang, F. Luo, C. Sang, J. Zeng, and S. Hirokawa, "Personalized movie recommendation system based on support vector machine and improved particle swarm optimization," *IEICE Trans. Inf. Syst.*, vol. E100-D, no. 2, pp. 285–293, 2017, DOI:10.1587/transinf.2016EDP7054.
- [20] G. Miao, Y. Gao, and Z. Zhu, "Digital movie recommendation algorithm based on big data platform," *Math. Problems Eng.*, vol. 2022, no. 1, art. no. 4163426, 2022.
- [21] M. H. Gebreselassie and M. Olusanya, "The use of metaheuristics in the evolution of collaborative filtering recommender systems: A review," in *Proc. Metaheuristics Int. Conf. (MIC)*, Cham, Switzerland: Springer Nature Switzerland, Jun. 2024, pp. 234–248.
- [22] F. Rajabi Kouchi, S. Oftadeh Balani, A. Esmaeilpour, M. Shafieian, R. Sirwan, and A. Hussein Mohammed, "A movie recommender system based on user profile and artificial bee colony optimization," *Comput. Intell. Neurosci.*, vol. 2023, article ID 2311817, 2023, DOI:10.1155/2023/2311817.
- [23] V. K. Singh, S. Sabharwal, and G. Gabrani, "A novel collaborative filtering based recommendation system using exponential grasshopper algorithm," *Evol. Intell.*, vol. 16, no. 2, pp. 621–631, 2023.
- [24] S. Khedairia and M. T. Khadir, "A multiple clustering combination approach based on iterative voting

- process," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 1, pp. 1370–1380, 2022.
- [25] N. Li and Y. Xia, "Movie recommendation based on ALS collaborative filtering recommendation algorithm with deep learning model," *Entertainment Comput.*, vol. 51, article 100715, 2024, DOI:10.1016/j.entcom.2024.100715.
- [26] S. Jayalakshmi, N. Ganesh, R. Čep, and J. Senthil Murugan, "Movie recommender systems: Concepts, methods, challenges, and future directions," *Sensors*, vol. 22, no. 13, art. no. 4904, 2022.
- [27] C. H. Meghana, K. Divya, C. Kesava, D. Devi, and S. Sajida Sultana, "Machine learning algorithms: Netflix recommendation system," in *Proc. Int. Conf. Innov. Comput. Commun. (ICICC)*, Singapore: Springer Nature Singapore, Feb. 2025, pp. 513–526.
- [28] R. Singh, P. Dwivedi, and V. Kant, "Comparative analysis of collaborative filtering techniques for the multi-criteria recommender systems," *Multimedia Tools Appl.*, vol. 83, no. 24, pp. 64551–64571, 2024.
- [29] L. Chen, R. Xiong, and Y. Ji, "Application of SVM model based on collaborative filtering hybrid algorithm in e-commerce recommendation," *Int. J. Comput. Appl.*, vol. 46, no. 5, pp. 292–300, 2024, DOI:10.1080/1206212X.2024.2309809.
- [30] Y. Mu and Y. Wu, "Multimodal movie recommendation system using deep learning," *Mathematics*, vol. 11, no. 4, 895, 2023, DOI:10.3390/math11040895.
- [31] J. Leander and A. Wicaksana, "Optimizing a personalized movie recommendation system with support vector machine and content-based filtering," *J. Syst. Manag. Sci.*, vol. 14, no. 1, pp. 490–501, 2024.
- [32] B. Walek and P. Fajmon, "A hybrid recommender system for an online store using a fuzzy expert system," *Expert Syst. Appl.*, vol. 212, art. no. 118565, 2023.
- [33] D. Cai, S. Qian, Q. Fang, J. Hu, and C. Xu, "User cold-start recommendation via inductive heterogeneous graph neural network," *ACM Trans. Inf. Syst.*, vol. 41, no. 3, pp. 1–27, 2023.
- [34] J. Fu et al., "Exploring adapter-based transfer learning for recommender systems: Empirical studies and practical insights," in *Proc. 17th ACM Int. Conf. Web Search Data Mining (WSDM)*, Mar. 2024, pp. 208–217.
- [35] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995, DOI:10.1007/BF00994018.
- [36] M. Tanveer, C. Gautam, and P. N. Suganthan, "Comprehensive evaluation of twin SVM based classifiers on UCI datasets," *Appl. Soft Comput.*, vol. 83, art. no. 105617, 2019.
- [37] J. Ali, M. Aldhaifallah, K. S. Nisar, A. A. Aljabr, and M. Tanveer, "Regularized least squares twin SVM for multiclass classification," *Big Data Res.*, vol. 27, art. no. 100295, 2022.
- [38] X.-S. Yang and X.-S. He, "Nature-inspired algorithms," in *Mathematical Foundations of Nature-Inspired Algorithms*. Cham, Switzerland: Springer International Publishing, 2019, pp. 21–40.
- [39] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, *Studies in Computational Intelligence*, vol. 284. Springer, 2010, pp. 65–74, DOI:10.1007/978-3-642-12538-6_6.
- [40] MovieLens 1M Dataset. [Online]. Available: <https://www.kaggle.com/datasets/odedgolden/movielens-1m-dataset>.
- [41] MATLAB System Requirements, MathWorks. [Online]. Available: <https://nl.mathworks.com/support/requirements/matlab-system-requirements.html>.