**Title:**

**A Deep Neural Network-Based Approach for Sentiment Analysis of Movie Reviews**

**Authors:**

Kifayat Ullah, Anwar Rashad, Muzammil Khan, Yazeed Ghadi, Hanan Aljuaid, and Zubair Nawaz

**Summary:**

This paper presents a **seven-layer deep learning model** designed to **automatically detect sentiment (positive or negative)** in **movie reviews**. With the vast number of reviews posted online, it’s not realistic to read them all manually — so this model helps automate the task using artificial intelligence.

**How the Model Works:**

The model reads reviews and breaks them down into a format a computer can understand (numbers), then identifies **important patterns in the text** (like “really enjoyed” or “not worth watching”) to decide whether the overall tone is positive or negative.

The architecture includes:

1. An **embedding layer** that turns words into vectors (numerical form).
2. **Two 1D convolutional layers** that extract patterns from the text.
3. A **global max-pooling layer** that summarizes the most important features.
4. A **dropout layer** to prevent overfitting (so it works well on new data).
5. **Dense layers** for the final classification.

**What Data Was Used:**

They tested the model on two large IMDb datasets:

* **Dataset I**: 25,000 reviews
* **Dataset II**: 50,000 reviews  
  Each dataset had an equal number of positive and negative reviews.

**Key Results:**

* The model achieved **about 92% accuracy** on both datasets.
* It **outperformed older models**, including a similar six-layer version, which only managed around 53–55% accuracy.
* The best results came from using **average-length reviews** (about 1200 words), and a **dropout rate of 0.3** helped prevent overfitting.

**Why It Matters:**

This model helps **quickly and accurately analyze public opinion** from movie reviews, which can be useful for film studios, streaming platforms, and researchers. The approach can also be adapted for other types of sentiment analysis beyond just movies.

**Title:**

**Enhancing Movie Recommendation Systems Through CNN-Based Feature Extraction and Optimized Collaborative Filtering**

**Author:**

Rui Guo

**Summary:**

This paper introduces a **hybrid movie recommendation system** that blends **deep learning (CNN)** with **collaborative filtering** to give users more accurate and personalized movie suggestions.

**What’s the problem?**

With thousands of movies available online, people often feel overwhelmed. Traditional recommendation systems struggle with two big issues:

* **Cold-start**: When there's not enough data about a user or movie.
* **Data sparsity**: Not every user has rated enough movies, making it hard to find patterns.

**The Solution:**

This system combines two ideas:

1. **CNN (Convolutional Neural Network)**: It looks at **movie posters** and extracts meaningful visual features that reflect movie themes and style.
2. **Collaborative Filtering**: It finds users with similar movie tastes and recommends what they liked.

Together, this **hybrid approach** uses both visual content and user behavior to generate smarter recommendations.

**How does it work?**

1. It uses the **MovieLens dataset**, which contains data on users, movies, and their ratings.
2. It processes and cleans the data.
3. Then it applies **VGG16 (a pre-trained CNN model)** to extract visual details from posters.
4. Using **cosine similarity**, it finds users who have similar tastes.
5. Based on those similarities, it creates a personalized movie list.

The model is constantly improved with **user feedback** and **new data** over time.

**Results:**

The model performed **significantly better** than traditional methods. It improved:

* **Prediction accuracy**
* **Personalization quality**
* **Cold-start handling** (by using movie posters even when ratings are sparse)

**Why it matters:**

This system is a step toward smarter, more human-like recommendations. It doesn’t just rely on numbers — it "looks" at the movie visually and blends that with what similar users liked. This kind of personalization can lead to **higher user satisfaction**, longer watch times, and more engagement on platforms like Netflix or Amazon Prime.

**Title:**

**Movie Rating Prediction Using Convolutional Neural Network Based on Historical Values**

**Authors:**

Rudy Aditya Abarja and Antoni Wibowo

**Summary:**

This paper proposes a deep learning method to **predict movie ratings *before* a movie is released**, using a **Convolutional Neural Network (CNN)** and **historical data** — no need for post-release reviews or user comments.

**What’s the problem?**

Most rating predictions rely on **user reviews or social media comments**, which only appear **after** the movie is out. That makes it hard for filmmakers or streaming platforms to plan ahead. Also, many models make **personalized predictions**, not general ones for everyone.

**The Solution:**

The authors introduce a **CNN-based model** that uses **historical metadata** — things like:

* Director
* Actors
* Genre
* Content rating
* Production company  
  This info is already available before release. They use **"historical values"**, which are stats from **similar past movies** (e.g., average ratings of movies with the same actor or director).

This allows the model to make **objective, pre-release rating predictions** — helpful for studios, marketers, and investors.

**How does it work?**

1. The system gathers metadata from sites like IMDb and TMDb.
2. It computes **historical features** (e.g., average past ratings of movies with the same crew).
3. These features are fed into a **CNN model** trained to predict the expected rating.
4. Dropout regularization is used to prevent overfitting.

**Results:**

* The CNN performed better than traditional methods (like linear regression or SVM).
* Using historical values made the predictions **more accurate and generalizable**.
* This approach worked well even **before** a movie had any public reviews.

**Why it matters:**

This method could let movie studios **predict box office success**, design better marketing campaigns, or choose smarter production strategies — all before a single audience member sees the film. It also avoids early ratings bias (like fan-influenced review bombs).

**Title:**

**Movie Recommendation Based on User Similarity of Consumption Pattern Change**

**Authors:**

Minjae Kim, Wonseok Choi, SungHwan Jeon, Haejin Chung, Heeseong Shin, and Yunmook Nah

**Summary:**

This paper proposes a **Recurrent Neural Network (RNN)**-based movie recommendation system that tracks **how users’ movie preferences change over time** — not just what they liked in the past.

**What’s the problem?**

Traditional recommendation systems usually:

* Treat all past preferences equally.
* Don’t consider **how tastes change** (e.g., someone might move from action movies to documentaries).
* Struggle with **data sparsity**, where not enough user ratings are available.

**The Solution:**

The authors created a system that:

1. **Groups users** with similar movie preferences (based on rating behavior).
2. Uses an **RNN** to learn **the sequence and evolution** of their movie-watching behavior.
3. Recommends movies by predicting what users with similar evolving tastes would want next.

**How does it work?**

* It starts by calculating **Pearson correlation** between users based on their movie ratings to group similar users.
* Then, it feeds the **watch history (as sequences)** of these user groups into a **modified RNN**.
* The RNN learns patterns in how preferences shift over time and uses this to recommend future movies.

**Results:**

* They tested the system using the **MovieLens Latest dataset** (over 1 million records).
* Compared to:
  + Traditional collaborative filtering (CF)
  + A basic RNN  
    → Their **modified RNN outperformed both**, showing lower error rates (RMSE, MSE, MAE).

**Why it matters:**

This approach mimics real life — our tastes change, and a smart recommendation system should keep up. By combining **user similarity** and **time-evolving behavior**, it delivers more relevant suggestions and improves user satisfaction over time.

**Title:**

**Movie Recommendation System Using RNN and Cognitive Thinking**

**Authors:**

Shubhada Labde, Vishesh Karan, Shubham Shah, and Dhruv Krishnan

**Summary:**

This paper presents a **hybrid movie recommendation system** that uses **Recurrent Neural Networks (RNNs)** along with **cognitive thinking** — especially considering **user age** and psychology — to generate more personalized suggestions.

**What’s the problem?**

Recommendation systems often:

* Ignore psychological factors (like how age affects genre preferences).
* Fail to adapt to evolving viewing patterns.
* Struggle with cold-start problems for new users.

**The Solution:**

The authors developed an **ensemble model** that combines multiple techniques:

1. **RNNs** to learn from the sequence of user preferences.
2. **Cognitive modeling** to tailor suggestions based on **age-related genre preferences**.
3. **Collaborative Filtering** and **Content-Based Filtering** for robustness.
4. **Popularity-Based models** for new users with no history.

**How does it work?**

The system is trained using **three datasets**:

* IMDb Bollywood dataset (15,000+ movies)
* A large Indian movie dataset (50,000+ rows)
* MovieLens 100K dataset

It uses:

* **One-hot encoding** for genres
* **SVD (Singular Value Decomposition)** to capture latent features
* **RNNs** for capturing temporal behavior
* Age-based logic to recommend genres (e.g., kids → animation, seniors → drama/comedy)

The model blends results from all individual learners into one **final recommendation list**.

**Results:**

* Each sub-model was optimized and validated separately.
* The hybrid approach resulted in **more accurate** and **more age-appropriate** recommendations.
* It also handled **cold-start users** better by using popularity and age heuristics.

**Why it matters:**

This system goes beyond just analyzing clicks and ratings — it mimics how humans think, especially how **age influences genre preference**. By blending multiple methods, it provides **personalized, psychologically aware recommendations** that better match the user’s mood and mindset.

**Title:**

**Multimodal Movie Recommendation System Using Deep Learning**

**Authors:**

Yongheng Mu and Yun Wu

**Summary:**

This paper introduces a **multimodal movie recommendation system** that uses **deep learning** to combine various data types — not just user ratings, but also movie posters, metadata, and user behaviors — to make smarter, more personalized suggestions.

**What’s the problem?**

Traditional recommendation systems suffer from:

* **Data sparsity** (not enough user ratings to work with)
* **Cold-start problems** (struggling to recommend new items or to new users)
* Limited understanding of movie content (e.g., they don’t “see” visuals like posters)

**The Solution:**

The authors propose a **deep learning model** that:

1. Analyzes **multimodal data** — text, visuals, and user behavior.
2. Learns **hidden features** of users and movies.
3. Predicts movie ratings more accurately, even when data is limited.

They use **MovieLens 100K and 1M datasets** to test their model.

**How does it work?**

* Inputs include **user ratings**, **movie metadata**, and **poster images**.
* Deep learning is used to **extract features** from all data types and build a **neural network**.
* The system is trained to **predict user ratings**, and those predictions are used to generate recommendations.

They compared their method to:

* User-based collaborative filtering (User-CF)
* Item-based content filtering (Item-CF)
* Singular Value Decomposition (SVD)

**Results:**

* The deep learning model **outperformed all traditional methods**.
* It achieved **RMSE scores of 0.9908 and 0.9096** on the 100K and 1M datasets, respectively.
* It handled cold-start and sparsity issues much better due to the use of **multiple data types**.

**Why it matters:**

This system mimics how humans choose movies — not just by reading text but by looking at posters, checking ratings, and browsing based on mood or interest. It shows that **combining different types of information leads to better recommendations**, especially for new users or content.

**Title:**

**Personalized Movie Recommendation Based on Convolutional Neural Network**

**Author:**

(Not listed in your extract; likely a single-author or unspecified team paper)

**Summary:**

This paper presents a **CNN-based personalized movie recommendation system** that improves suggestion accuracy by learning hidden patterns in user preferences and movie features — especially when traditional methods fall short.

**What’s the problem?**

Traditional recommendation systems (like collaborative filtering) often:

* Rely too heavily on user ratings.
* Struggle with **cold-start** and **data sparsity** (limited user input).
* Don’t deeply understand movie content beyond surface-level metadata.

**The Solution:**

This system uses **Convolutional Neural Networks (CNNs)** to:

* Extract **complex patterns** from both **user behavior** and **movie metadata**.
* Provide **personalized recommendations** by learning deeper features rather than relying on raw ratings alone.

The CNN helps analyze even limited data more meaningfully by recognizing subtle connections that users themselves might not even realize.

**How does it work?**

* The model uses a **user-movie matrix** as its input.
* A CNN processes this matrix to detect **nonlinear associations** between users and movies.
* It then ranks movies based on **predicted interest** levels for each individual user.

The system is also continuously updated as users interact with new content, which helps keep the recommendations fresh and relevant.

**Results:**

* The proposed CNN model performed **better than traditional methods** like:
  + Basic collaborative filtering
  + Content-based filtering
* It achieved higher **accuracy and personalization**, especially for users with fewer interactions (solving the cold-start issue).

**Why it matters:**

By leveraging CNNs — which are usually used in image recognition — this system provides a **deep, layered understanding of user behavior**, resulting in **smarter, more relevant suggestions**. It's a step toward recommendation engines that think more like humans by learning from context, not just numbers.

**Title:**

**Research on Movie Recommendation Model Based on LSTM and CNN**

**Authors:**

Wentao Wang, Chengxu Ye, Ping Yang, and Zhikun Miao

**Summary:**

This paper proposes a **hybrid deep learning model** that combines **LSTM (Long Short-Term Memory)** and **CNN (Convolutional Neural Network)** to **predict movie ratings** and generate more accurate recommendations. It takes advantage of both **time-based user behavior** and **movie content features**.

**What’s the problem?**

Most traditional systems can’t:

* Capture how user preferences change over time.
* Understand the actual content of the movie beyond metadata.
* Handle **cold-start** or **sparse data** effectively.

**The Solution:**

The authors design a model that fuses two powerful deep learning tools:

1. **LSTM**: To learn **time-based patterns** in user rating behavior (i.e., how your preferences shift over time).
2. **CNN**: To extract meaningful **features from movie titles** — such as keywords that indicate genre or tone.

These features are then merged to **predict ratings** and create a **personalized movie list**.

**How does it work?**

* **Input layer**: Takes in user details (ID, age, gender), movie metadata, and rating history.
* **LSTM module**: Learns how each user’s taste evolves based on the order and context of their previous ratings.
* **CNN module**: Analyzes movie titles using convolutional filters to pick out relevant patterns.
* **Fusion**: Combines both feature sets to predict how much a user would like a new movie.

**Results:**

* Tested on the **MovieLens dataset**.
* Achieved **4.4% to 18.7% improvement in Mean Squared Error (MSE)** and up to **52.2% improvement in Mean Absolute Error (MAE)** over traditional models.
* Proved to be **more accurate** than standalone LSTM, CNN, or older machine learning methods.

**Why it matters:**

This model reflects **how people actually think and watch movies** — our preferences shift over time, and titles often hint at whether a movie’s up our alley. By combining the strengths of **sequence learning (LSTM)** and **content analysis (CNN)**, this system delivers **smart, adaptable, and human-like recommendations**.

**Title:**

**A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis**

**Authors:**

Anwar Ur Rehman, Ahmad Kamran Malik, Basit Raza, Waqar Ali

**Summary:**

This paper proposes a **hybrid deep learning model** that combines **Convolutional Neural Networks (CNN)** and **Long Short-Term Memory (LSTM)** to perform **sentiment analysis on movie reviews** — essentially determining whether a review is positive or negative.

**What’s the problem?**

With the rise of online reviews (especially on platforms like IMDb and Amazon), there's **a huge amount of opinion data**, but it's **unstructured** and hard to analyze at scale. Traditional methods either:

* Miss the **context** behind word sequences, or
* Struggle with capturing **local text patterns**.

**The Solution:**

The authors created a **CNN-LSTM hybrid model** that combines the best of both worlds:

* **CNN** extracts important **local features** (like phrases such as “really loved” or “extremely boring”).
* **LSTM** handles **long-term dependencies** in text (like understanding that “Although the beginning was slow, the ending was fantastic” is overall positive).

They also use:

* **Word2Vec** for word embedding (turns text into meaningful numerical vectors).
* **Global max-pooling**, **dropout**, and **ReLU activation** to boost performance and reduce overfitting.

**How does it work?**

1. Convert reviews into word vectors using **Word2Vec**.
2. Pass them through several **convolutional layers** to capture patterns.
3. Feed the CNN output into an **LSTM layer** to capture context over time.
4. Final layers perform classification (positive or negative).

**Results:**

* Tested on **IMDb** and **Amazon movie review** datasets.
* The model achieved **high precision, recall, F1 score, and accuracy**, beating traditional machine learning and standalone deep learning models.
* Especially good at handling **long reviews** and **complex sentence structures**.

**Why it matters:**

By combining CNN and LSTM, this model more closely mirrors how **humans understand language** — recognizing key phrases while also getting the broader meaning. It’s a powerful, scalable way to automatically analyze large volumes of reviews and extract public opinion.

**Title:**

**A Personalized Movie Recommendation System Based on LSTM-CNN**

**Authors:**

Haili Wang, Zhenlin Chao, and Nana Lou

**Summary:**

This paper introduces a **deep learning–powered movie recommendation system** that uses a **combination of LSTM (Long Short-Term Memory)** and **CNN (Convolutional Neural Networks)** to provide **highly personalized suggestions** based on user behavior and movie metadata.

**What’s the problem?**

In the age of **information overload**, people often have too many choices. Traditional recommendation systems:

* Rely on shallow models like collaborative filtering,
* Struggle with **cold-start problems** (e.g., when there's not enough data on new users or movies),
* And can’t capture complex patterns in behavior.

**The Solution:**

The authors combine two deep learning models:

1. **LSTM**: Tracks how a user’s preferences evolve over time by looking at **sequential behavior** (like watching habits).
2. **CNN**: Extracts deeper features from **movie titles** and other structured data.

Together, they build a powerful hybrid system that:

* Learns both **temporal patterns** in behavior,
* And **content-level patterns** in the movies themselves.

**How does it work?**

1. User metadata (ID, gender, age, occupation) is embedded into a feature matrix.
2. Movie data (e.g., titles) is passed through the LSTM layer to **preserve word order and context**.
3. Output from LSTM goes into a CNN to **extract hierarchical features**.
4. The final output is a **Top-N list of movie recommendations**.

They use the **MovieLens dataset** for training and testing, applying standard train/test splits.

**Results:**

* The LSTM-CNN model **outperformed traditional methods**, especially in:
  + **Personalization quality**
  + **Handling sparse data**
  + **Adapting to user preferences over time**
* The combination of behavioral and content-based learning made it **robust and flexible**.

**Why it matters:**

This approach reflects how people actually choose movies — combining **past behavior**, **personal characteristics**, and **movie features**. The LSTM-CNN hybrid gives **smarter, more relevant suggestions** than either method alone, especially in real-world, data-heavy platforms like Netflix or YouTube.

**Title:**

**Analytical Approach for Sentiment Analysis of Movie Reviews Using CNN and LSTM**

**Authors:**

Arushi Garg, Soumya Vats, Garima Jaiswal, Arun Sharma

**Summary:**

This paper offers a deep dive into using **CNN (Convolutional Neural Networks)** and **LSTM (Long Short-Term Memory)** for **sentiment analysis of movie reviews**, aiming to determine whether a review is **positive or negative**. It’s a comparative and analytical study covering several model combinations.

**What’s the problem?**

With massive amounts of movie reviews online, there’s a **need to automatically understand public opinion**. Traditional machine learning models often:

* Fail to capture deep semantic meaning.
* Don’t handle **sequence** or **context** well in language.
* Are limited by handcrafted features.

**The Solution:**

This study reviews and implements **multiple hybrid and standalone deep learning models** for sentiment classification, focusing especially on:

* **CNNs**: Great for spotting **important local features** (e.g., key phrases).
* **LSTMs**: Great for handling **long-range dependencies and context** in text.

They use **Word2Vec embeddings** for word representation and test various deep learning combinations, including:

* CNN only
* LSTM only
* CNN + LSTM (hybrid)
* Bi-LSTM
* CNN-BiLSTM
* CNN with attention mechanisms

**How does it work?**

* Preprocesses raw reviews (cleaning, tokenizing, encoding).
* Converts text into word vectors using Word2Vec.
* Runs the data through **hybrid architectures**, often with dropout layers and ReLU for better learning.
* Models are trained and evaluated on standard datasets like **IMDb** and **Amazon reviews**.

**Results:**

* CNN+LSTM models achieved up to **91% accuracy**.
* CNN alone performed well but missed long-term dependencies.
* LSTM alone was slower and less efficient with short reviews.
* **Hybrid models consistently outperformed individual models** in terms of accuracy, recall, and F1-score.
* Models using **Bi-LSTM** or **attention** showed the best balance between precision and generalization.

**Why it matters:**

This work shows that **hybrid deep learning models** can better handle real-world language, especially in movie reviews which often involve nuanced sentiment. By combining CNN and LSTM strengths, these models are **faster, more accurate**, and better at understanding natural text than older machine learning techniques.

**Title:**

**Convolutional Neural Network-Based Personalized Program Recommendation System for Smart Television Users**

**Authors:**

Khasim Vali Dudekula, Hussain Syed, Mohamed Iqbal Mahaboob Basha, Sudhakar Ilango Swamykan, Purna Prakash Kasaraneni, Y.V.P. Kumar, Aymen Flah, Ahmad Taher Azar

**Summary:**

This paper introduces a smart, **CNN-powered personalized recommendation system** for **smart TV users**, using **facial recognition** to match viewers with their preferences — then recommending shows accordingly. It’s one of the first to combine **computer vision + deep learning + hybrid filtering** in a smart home setting.

**What’s the problem?**

Traditional smart TV recommendation systems:

* Don’t account for **individual users** in multi-user households.
* Can’t **verify who’s watching** the screen.
* Mostly rely on **ratings or watch history**, which isn't always available or accurate.

**The Solution:**

The authors propose a system that:

1. Uses a **smart TV camera** to **capture a user’s face**.
2. Uses **CNN (Convolutional Neural Network)** to **identify the user** via face recognition.
3. Pulls up the user’s personalized data and applies a **hybrid filtering algorithm** (combining content-based and collaborative filtering) to recommend programs.

This way, the TV shows **custom recommendations** the moment a person sits down — no login or manual selection needed.

**How does it work?**

* CNN is trained on **CelebA** and **LFW face datasets** for high-accuracy facial recognition.
* A **synthetic dataset** stores each user’s:
  + Face image
  + Program preferences
  + Watch duration per channel
* When someone turns on the TV, the system:
  + Captures their face,
  + Matches it to the dataset,
  + Recommends personalized content based on hybrid filtering.

The model also uses **SMOTE** to handle class imbalance, **morphological image preprocessing**, and **local storage** to protect privacy.

**Results:**

* CNN face detection achieved ~**95% accuracy**.
* The hybrid recommendation system reached:
  + ~**85% accuracy** for single users
  + ~**81% accuracy** for multi-user environments
* Outperformed traditional filtering systems in both personalization and speed.

**Why it matters:**

This system brings **Netflix-style personalization** to shared environments like living rooms — without needing accounts, logins, or remote-control navigation. By combining face recognition with recommendation tech, it creates a **hands-free, user-aware entertainment experience**, especially useful in smart homes.

### **Title:**

**An Analysis and Comparison of Deep-Learning Techniques and Hybrid Model for Sentiment Analysis for Movie Review**

**Authors:**  
 Swapnil Sinha, Abishek Jayan, Rishabh Kumar

### **Summary:**

This paper compares several deep learning models — **CNN, LSTM, GRU, BERT**, and **hybrid combinations** like BERT-LSTM and BERT-CNN — to determine the best approach for **sentiment analysis of movie reviews**.

#### **What’s the problem?**

Sentiment analysis models often struggle with:

* Capturing both **word meaning** and **context**.
* Choosing the right **embedding technique** (Word2Vec, GloVe).
* Balancing **speed vs accuracy** for large datasets like IMDb.

#### **The Solution:**

They conducted a **comprehensive benchmark**:

* Preprocessed IMDb data
* Applied different word embedding methods (Word2Vec, GloVe)
* Evaluated deep learning models and hybrids using **precision, recall, F1-score**

#### **Results:**

* **BERT-based hybrid models (BERT-LSTM and BERT-CNN)** outperformed all others.
* These models better captured **contextual and semantic information** in reviews.
* Hybrid models had the highest **accuracy and generalization**.

#### **Why it matters:**

The study provides a **practical guide** for choosing the right deep learning model for sentiment analysis. It confirms that **hybrid approaches** using pre-trained models like BERT offer **superior results** on real-world datasets.

### **Title:**

**Integrating User and Item Reviews in Deep Cooperative Neural Networks for Movie Recommendation**

**Authors:**  
 Aristeidis Karras and Christos Karras**Summary:**

This paper introduces **DeepCoNN**, a deep learning model that learns from both **user reviews and movie reviews** to make better movie recommendations.

#### **What’s the problem?**

* Traditional collaborative filtering can't deal well with **cold-start** and **sparse ratings**.
* They **ignore valuable text data** from user-written reviews.

#### **The Solution:**

**DeepCoNN** has two parallel neural networks:

* One learns from **reviews written by a user**.
* The other learns from **reviews written about a movie**.

The outputs are combined using a **shared layer** similar to **factorization machines**, allowing the system to connect user and movie traits meaningfully.

#### **How it works:**

* Uses **Amazon Instant Video Reviews** dataset.
* Reviews are processed with **GloVe embeddings**.
* CNNs extract text features.
* Final predictions are made via a **dot product of the two networks’ outputs**.

#### **Results:**

* Outperformed traditional methods in **cold-start and sparse data scenarios**.
* Achieved high accuracy in **rating prediction** and **user-item matching**.

#### **Why it matters:**

DeepCoNN shows how **textual reviews** can dramatically enhance personalization. It’s a **real-world scalable model** for platforms like Amazon or Netflix that collect rich user feedback.

### **Title:**

**Movie Recommendation System Based on Tweets Using Switching Hybrid Filtering with Recurrent Neural Network**

**Authors:**  
 Berlian Muhammad Galin Al Awienoor, Erwin Budi Setiawan

### **Summary:**

This paper builds a movie recommendation system using **Twitter data** and combines **Switching Hybrid Filtering (SHF)** with a **Recurrent Neural Network (RNN)** for better prediction.

#### **What’s the problem?**

* People struggle to pick a movie from thousands of options.
* Ratings are often **sparse**, especially for new movies.
* Twitter is underused as a source of user feedback.

#### **The Solution:**

1. Use **SHF**: If collaborative filtering doesn’t work (due to sparse data), switch to **content-based filtering** using RoBERTa.
2. Apply **RNN** for classification of tweets into movie sentiment scores.

#### **How it works:**

* Crawled tweets about 855 Netflix movies.
* Extracted sentiment labels (0–5 ratings) and classified recommendations (yes/no).
* Used **Nadam optimization** for the best training results.

#### **Results:**

* **Accuracy:** 86.11% (RNN)
* **MAE:** 0.0617
* **RMSE:** 0.1178
* RNN significantly outperformed SVM and Naive Bayes.

#### **Why it matters:**

By combining Twitter data and hybrid filtering, this system offers **real-time, relevant recommendations**, even for **new or lesser-known content**. It’s especially effective on **sparse, dynamic social data**.

### **Title:**

**Sentiment Analysis of Movie Reviews Based on LSTM-Adaboost**

**Authors:**  
 Ling Zhang, Miao Wang, Ming Liu, Haozhan Li

### **Summary:**

This study introduces an enhanced sentiment classifier by combining **LSTM** (for sequence learning) with **Adaboost** (for boosting weak learners).

#### **What’s the problem?**

* LSTM is good at learning from sequences but **can still be weak** under certain conditions (e.g., noisy text or long reviews).
* Most methods use single models rather than ensembles.

#### **The Solution:**

The authors combine:

* **LSTM**, which handles context in reviews.
* **Adaboost**, which builds multiple weak LSTM classifiers and boosts them into a stronger ensemble

**How it works:**

* Used **IMDb reviews dataset**
* Compared **LSTM-Adaboost** against vanilla CNN and LSTM models
* Evaluated based on accuracy and other classification metrics

#### **Results:**

* **6% improvement in accuracy** over standalone LSTM
* Better handling of **mixed, long, and ambiguous reviews**

#### **Why it matters:**

This method combines the best of **deep learning and ensemble learning**, resulting in a more **robust and accurate sentiment classifier** — perfect for platforms analyzing huge volumes of user feedback.

### **Title:**

**Sentiment Analysis of Movie Reviews Using Improved Word2Vec and CNN**

**Authors:**  
 Azhar Imran, Muhammad Imran Khan, Khattab M Ali Alheeti, Abdulkareem Alzahrani

### **Summary:**

This paper proposes a unique method combining **Word2Vec + LDA (Latent Dirichlet Allocation)** with a **CNN** to improve sentiment analysis on movie reviews.

#### **What’s the problem?**

* Traditional Word2Vec captures **only semantic** meaning, missing **topic or contextual info**.
* Sentiment classification models often ignore the **hierarchical structure** in language.

#### **The Solution:**

* Enhance Word2Vec by integrating **LDA** to capture both **semantic and topical context**.
* Feed the improved embeddings into a **7-layer CNN** for binary sentiment classification.

#### **How it works:**

* Used Stanford movie review dataset (50,000 reviews).
* Created new feature representations using Word2Vec + LDA.
* Used CNN for final sentiment prediction.

#### **Results:**

* **Accuracy:** 92.4%
* **Sensitivity:** 93.2%
* **Specificity:** 92.4%
* Outperformed all baseline models (SVM, CNN, LSTM)

#### **Why it matters:**

This model shows how **improving word embeddings** can lead to better sentiment understanding. It's especially good for nuanced movie reviews where both content and tone matter.