SITEKIN: Jurnal Sains, Teknologi dan Industri, Vol. 22, No. 1, December 2024, pp. 92 - 100
ISSN 2407-0939 print/ISSN 2721-2041 online

Prediction of Anime Rating with Hybrid Artificial Neural Networks and Convolutional Neural Networks

M. Nurudduja Al Kautsar*[1], Violita Anggraini[2], Arif Reza Basirun[3], Achmad Pratama Rifai[4]
[1,2,3,4]Departemen Teknik Mesin dan Industri, Universitas Gadjah Mada
Jalan Grafika No. 2, Yogyakarta, 55281, Indonesia
Email: muchamadnurudduja alkautsar@mail.ugm.ac.id[1], violitaanggraini@mail.ugm.ac.id[2], arifrezabasirun@mail.ugm.ac.id[3], achmad.p.rifai@ugm.ac.id[4]

ABSTRACT

This study proposes an innovative approach to predict anime scores by leveraging a combination of Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN). Tabular data such as source, number of episodes, type, and genre are incorporated alongside the image representation of anime into a holistic model. Evaluation results on the test set show satisfactory performance, with an average loss value of 0.673, Mean Absolute Error (MAE) of 0.654, and Mean Absolute Percentage Error (MAPE) of 9.44%. Training and validation graphs reflect the model's convergence without significant signs of overfitting or underfitting. The integration of information from both data sources yields a model capable of providing accurate predictions of anime scores, contributing to an understanding of trends and preferences in the anime industry, and opening opportunities for the development of similar models in the field of score prediction or other quality evaluations.

Keywords: *Anime, Artificial Neural Networks, Convolutional Neural Networks, Rating score*

Introduction

In the last decade, advancements in Artificial Intelligence (AI) and Machine Learning (ML) have transformed numerous sectors, including the entertainment industry [1-2]. One critical aspect of this industry is the prediction of anime scores, which significantly influences the evaluation of an anime's popularity and acceptance [3].

Anime has become a widely appreciated form of visual art and a significant part of global popular culture [4]. However, preferences for anime genres vary across different societies, similar to the diverse tastes seen with movies. To ensure the continued development of anime, it is essential to understand the preferences of anime enthusiasts through their evaluations and feedback [5]. One effective approach to achieving this understanding is through recommendation systems, which analyze user ratings, sentiments, genres, and other factors to provide personalized recommendations.

Several studies have focused on developing recommendation systems for anime using machine learning techniques. Jena et al. [6] explored various machine learning methods, including content-based filtering, popularity filtering, and collaborative filtering, using K-Nearest Neighbour (KNN) and Singular Value Decomposition (SVD) to build an anime recommendation system. Ota et al. [7] introduced AniReco, a system capable of recommending anime and related content in a cross-sectional manner, reflecting users' potential preferences. Nuurshadieq and Wibowo [8] proposed a deep learning method that incorporates side information from both users and anime works into a hybrid model. This model learns embeddings for users and anime separately, incorporating a Long Short-Term Memory (LSTM) layer to extract information from long text features like synopses, which is then fed into a deep neural network to predict user ratings. Soni et al. [9] developed RikoNet, an innovative anime recommendation engine that combines content-based and collaborative filtering approaches into a robust and effective system.

Despite the progress made with these models, there has been limited research utilizing the visual information contained in anime posters. Posters contain rich visual information, including art style, characters, color schemes, and thematic elements. These visual cues can provide insights into the genre, mood, and type of anime, which are crucial for making accurate recommendations. In addition, the visual elements in posters often reflect the content and tone of the anime. For instance, a poster with vibrant colors and dynamic poses might indicate an action-packed series, while a more subdued and artistic poster might suggest a drama or slice-of-life genre.

To address this gap, this research focuses on developing an anime score prediction model using a hybrid approach that combines Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN). ANN and CNN are chosen for their superior ability to process and learn features from large and complex datasets, which are common in anime-related data [10]. The model aims to predict anime scores based on various parameters, including the source of adaptation, number of episodes, type of anime, genre, studio, and poster image. By leveraging the rich visual information in posters and integrating it with traditional text-based methods, this approach addresses data sparsity and enhances the accuracy and personalization of recommendations. Advanced deep learning techniques further enable the system to provide nuanced recommendations that cater to users' visual and content preferences.

This study is particularly important given the continuously growing anime industry and its significant role in global pop culture [11]. By utilizing AI technology, specifically ANN and CNN, this research aims to lay a foundational framework for the analysis and prediction of trends in the anime industry, benefiting producers, content creators, and anime fans worldwide. By employing innovative methodologies, this research aspires to develop an accurate and efficient predictive model, significantly contributing to our understanding of anime fan preferences.

Research Methods

The primary objective of this research is to develop a robust anime score prediction model by leveraging both textual and visual data through the application of advanced machine learning techniques, specifically Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN). This section outlines the systematic approach employed to achieve this objective, detailing the key stages from data collection to model implementation and evaluation.

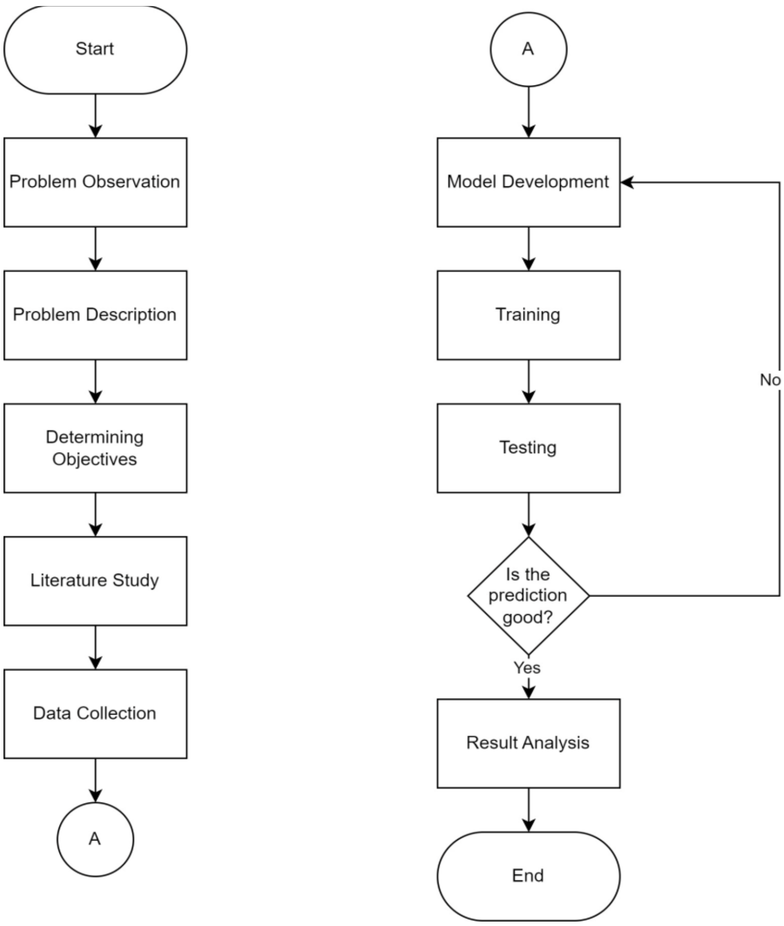


Figure 1. Research Flow

Anime recommendation systems traditionally rely on textual data such as user ratings, genres, and synopses to predict user preferences. However, these approaches often overlook the rich visual information contained in anime posters, which can provide significant insights into the content and aesthetic appeal of an anime. By integrating visual features extracted from posters with textual data, this research aims to enhance the accuracy and personalization of anime recommendations.

This research was meticulously designed and conducted through a series of structured process stages, which are illustrated in the Research Framework in Figure 1. This framework serves as an important guide in conducting each step of the research with the aim of achieving valid and accountable results. Here is a detailed description of each process stage:

[1] Problem Observation: Identify needs and consider problems in the context of anime score prediction. Conduct an initial analysis of the problem to be solved.

[2] Problem Description: Explain in detail about the existing problem, including factors that affect anime scores and the potential impact of accurate score predictions.

[3] Determining Objectives: Establish clear and measurable objectives for the research related to predicting anime scores based on specific features, such as genre, production studio, or episode duration using ANN and CNN.

[4] Literature Study: At this stage, conduct an in-depth study of various scientific articles and relevant journals related to the research topic. Review literature on anime score prediction using ANN and CNN techniques. Review related research, models used, evaluation metrics, and results achieved.

[5] Data Collection: Select an appropriate anime dataset for score prediction, ensuring the dataset includes various information such as genre, studio, duration, popularity, and existing scores.

[6] Model Development: Design ANN and CNN models for anime score prediction, including variations in architecture and parameters to be explored. Design a CNN model useful for extracting features from posters or related anime images.

[7] Training: Build the ANN and CNN models according to previous analysis using the collected dataset. Then, train the models with optimization algorithms to improve prediction performance.

[8] Testing: Evaluate the performance of ANN and CNN models using a separate dataset for testing. Use appropriate evaluation metrics such as Mean Squared Error (MSE) to assess the accuracy of anime score predictions [12].

[9] Result Analysis: In this stage, the outcomes of the ANN and CNN models are thoroughly analyzed to interpret the accuracy and effectiveness of the score predictions. This involves comparing the predicted scores against the actual scores to assess the precision of the models. This step is crucial to identify areas for improvement in the models and to understand the strengths and limitations of the applied methodologies.

The combination of ANN and CNN in this architecture allows the model to process numerical and visual information simultaneously. Numerical data is input into the ANN model to process non-visual attributes, while visual data (such as images) is input into the CNN model to extract visual features. The combination of numerical and visual representations is done through concatenation and dense layers, allowing the model to learn more complex relationships between the two types of data, thus enabling better anime score predictions [13]. The combination of ANN (Artificial Neural Network) and CNN (Convolutional Neural Network) can be done with an architecture that takes advantage of the strengths of each model.

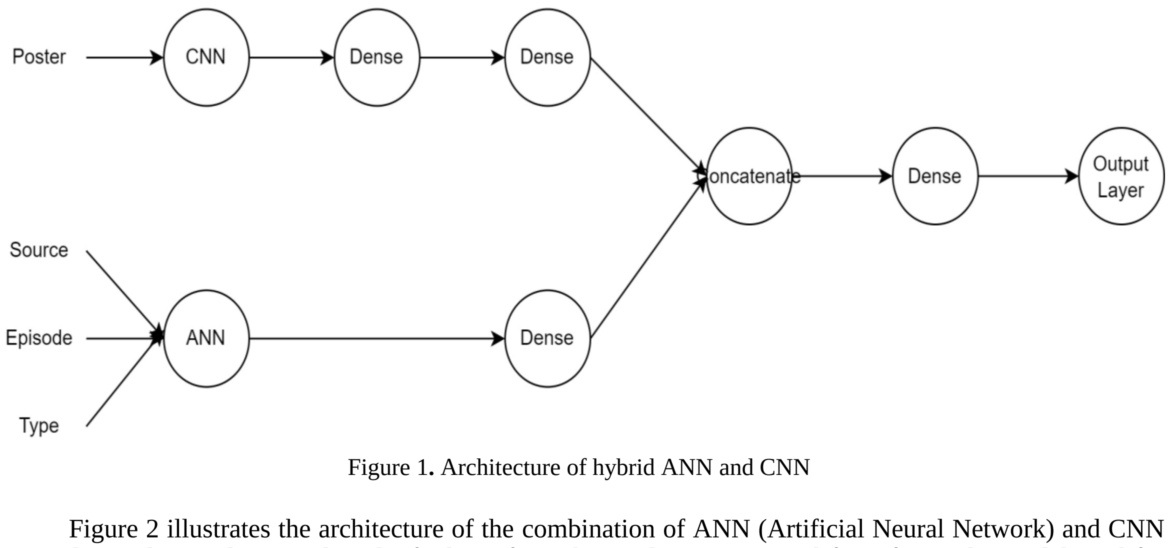


Figure 1. Architecture of hybrid ANN and CNN

Figure 2 illustrates the architecture of the combination of ANN (Artificial Neural Network) and CNN (Convolutional Neural Network). The findings from this analysis are critical for refining the models and for providing insights into the dynamics of anime score prediction. This comprehensive analysis contributes to enhancing the understanding of how different attributes of anime correlate with its popularity and acceptance, thereby offering valuable insights for producers, content creators, and the anime community at large. Here is an explanation of the architecture used:

[1] Input Layer: The first input enters the ANN model as a feature vector generated from numerical data or more general attributes related to anime. Anime data is taken from a CSV file which includes information about the source, number of episodes, type, score, genre, studio, and image file name. The second input enters the CNN model as images or structured data like anime posters.

[2] Data Pre-Processing: After loading the data, the next step is data pre-processing. This stage includes steps like cleaning data, removing missing or irrelevant values, data transformation (such as normalization, standardization), data type conversion, or feature extraction. Numerical data like 'Episodes' are standardized, while categorical data like 'Type', 'Source', and 'Genres' are converted using OneHotEncoder to change categories into a format suitable for the model. Data is divided into three parts: training, validation, and testing sets in specific proportions to validate model performance and avoid overfitting.

[3] ANN Layer: ANN (Dense Layer): The feature vector from numerical data is input into the Dense layer, which may have several neurons to process numerical information. The ANN model is created to process pre-processed structured data from anime using dense layers. CNN (Convolutional Layer): For image data, convolutional layers in the CNN model extract visual features like patterns, lines, or textures from images. The CNN model is created to process anime images using convolutional and pooling layers.

[4] CNN Layer: CNN (Dense Layer): After convolutional and pooling layers, the output of CNN can be flattened and embedded into the Dense Layer in the ANN model to combine visual information with numerical information.

[5] Concatenate Layer: After the Dense layers of ANN and the Convolutional layers of CNN, the outputs of both models can be combined using the concatenate layer. This combines information from both paths, allowing the model to learn a combined representation of numerical and visual data.

[6] Dense Layer: After merging, the Dense layer can be used again to process the combined representation of the data. This Dense layer typically contains several neurons and may have specific activation functions. Model Compilation: The combined model is compiled with the 'adam' optimizer, 'mean_squared_error' loss, and MeanAbsoluteError and MeanAbsolutePercentageError metrics. Compiling the model allows the use of these parameters, which will control how the model learns during the training process. The chosen optimizer, loss function, and evaluation metrics will affect how the model adjusts its weights and how the model's performance is measured during and after training.
   a) Model Training: The model is trained using a data generator that automatically provides batches of image and structured data during the training process.
   b) Model Storage: After the model is trained, it is stored for use in further evaluation or applications.
   c) Image Pre-Processing for Testing: Images for the testing set are pre-processed according to the standards applied to the training images.
   d) Model Evaluation: The model is evaluated against the testing set to measure prediction accuracy on previously unseen data.
   e) Calculation of Evaluation Metrics: Metrics such as loss, MAE, and MAPE are calculated for each batch in the testing set and then averaged.

[7] Output Layer: The final layer is the output layer, producing the predicted anime score value. At this stage, one neuron is used to produce a single score. The output of the model represents the performance of the model after being trained and tested using a previously unseen dataset (testing set). The output will display performance indicators using the average loss, MAE, and MAPE from the testing set. These values provide information about how well or poorly the model performs predictions on new data. The smaller the values of loss, MAE, and MAPE, the better the performance of the model.

Results and Discussion

In this chapter, the results of the testing of the implementation of anime score prediction using artificial neural networks and convolutional neural networks will be explained. The results of the research and discussion represent data analysis that includes discussion based on the problem formulation and research objectives. This stage indicates the readiness of the system to be tested in real situations, thus evaluating whether the results obtained are in line with the expected objectives.

Experiments

Figure 3 provides Input-Process-Output diagram which illustrates the flow of a hybrid model designed to predict anime ratings by integrating both textual and visual data through Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN). In the input section, various types of data are considered: textual input includes categorical and numerical features such as the source of adaptation, number of episodes, type of anime, fan score, genre, and studio name; visual input consists of the anime poster image. The process section details how the ANN handles the textual inputs by learning patterns and relationships, while the CNN processes the visual inputs to extract relevant features. These features are then concatenated into a single vector and passed through additional layers to generate the final output. The output section presents the predicted anime rating and includes performance indicators such as Loss, Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) to evaluate the model's accuracy. This hybrid approach, by combining ANN and CNN, leverages a comprehensive set of features to enhance the precision and personalization of anime rating predictions.

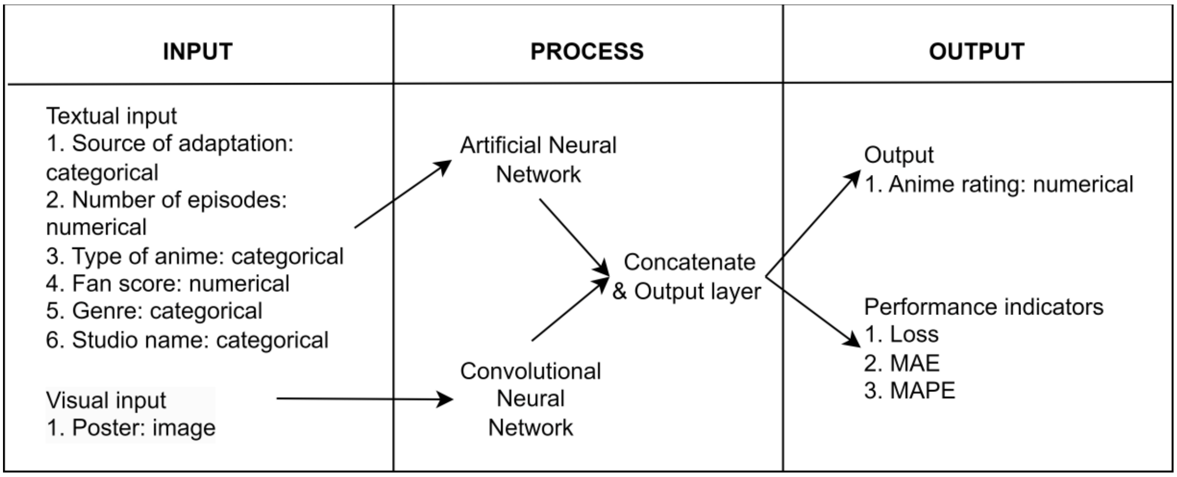


Figure 3. Input-Process-Output diagram

The implementation of a hybrid model that combines Convolutional Neural Network (CNN) and Artificial Neural Network (ANN) to predict the score values from anime data are executed using Tensorflow library written in Python. Here are the explanations of the steps and the purpose of each part of the coding:

[1] Load the CSV file: Using Pandas, read data from a CSV file containing information about anime, such as source (Source), number of episodes (Episodes), type (Type), score (Score), genre, studio, etc.
[2] Preprocess Data: Select the columns to be used for the model.
[3] Preprocessing Numerical and Categorical Data: Use make_column_transformer from scikit-learn to perform scaling on the 'Episodes' column and one-hot encoding on the categorical columns.
[4] Splitting Data into Train, Validation, and Test Sets: Divide the data into training, validation, and testing sets using train_test_split.
[5] CNN Model: Build a CNN model to process anime images.
[6] ANN Model: Build an ANN model to process numerical and categorical data.
[7] Combine CNN and ANN: Combine the outputs from the CNN model and ANN model using a concatenate layer.
[8] Compile the Model: Compile the model for regression tasks using mean squared error as the loss function and several other evaluation metrics.
[9] Custom Generator Function: Create a custom generator to provide batches of data consisting of images and ANN data on the training and validation sets.
[10] Create Training and Validation Generators: Create generators for the training and validation sets using the previously created generator function.
[11] Train the Model: Train the model using the training and validation generators for 100 epochs.
[12] Save the Model: Save the model to the file 'models_30y_v3.keras'.
[13] Convert the Test ANN Data to a Dense Array: Convert the ANN test data into a dense array.
[14] Prepare Test Images: Use ImageDataGenerator to prepare images from the test set.
[15] Evaluate the Model on the Test Set: Use the trained model to predict scores on the test set and calculate evaluation metrics such as mean absolute error (MAE) and mean absolute percentage error (MAPE).

The performance of the model is assessed based on three criteria: average test loss, MAE, and MAPE. Average Test Loss reflects how well or poorly the model can predict anime scores. The lower the loss value, the better the model is at making predictions. Average Test MAE measures the average absolute difference between the model's predictions and the actual values on the test dataset. The smaller the MAE value, the more accurate the model is in predicting anime scores. Average test MAPE measures the average relative error percentage between predictions and actual values on the test dataset. The smaller the MAPE value, the better the model is at providing predictions in terms of the percentage of relative error. The results of experiments are described in Table 1.

Table 1. Results of experiment

No.	Indicator	Value
1.	Average Test Loss	0.6732
2.	Average Test MAE	0.6537
3.	Average Test MAPE	9.4409%

The model's performance metrics, as detailed in the results, indicate the effectiveness of the hybrid ANN and CNN approach in predicting anime ratings. The key performance indicators are as follows:

- Average Test Loss: The model's average test loss is 0.6732. Loss is a measure of the difference between the predicted and actual ratings, with lower values indicating better performance. A loss of 0.6732 suggests that the model is fairly accurate in its predictions, though there is still room for improvement.
- Average Test MAE (Mean Absolute Error): The average test MAE is 0.6537. MAE represents the average magnitude of errors in the model's predictions, without considering their direction. An MAE of 0.6537 means that, on average, the model's predictions are off by about 0.6537 units on the rating scale, which is relatively precise given typical rating scales.
- Average Test MAPE (Mean Absolute Percentage Error): The average test MAPE is 9.4409%. MAPE indicates the average percentage error between the predicted and actual ratings. A MAPE of 9.4409% suggests that, on average, the model's predictions deviate from the actual ratings by about 9.44%. This is a strong performance, implying that the model can predict ratings with less than 10% error, which is generally considered acceptable for many predictive tasks.

Overall, these metrics indicate that the hybrid model is performing well, providing reasonably accurate predictions of anime ratings based on the combined textual and visual data inputs. The relatively low values of loss, MAE, and MAPE reflect the model's ability to capture the underlying patterns in the data and make reliable predictions.

Discussion

The analysis progresses with the objective of creating three charts to illustrate the evolution of loss and the evaluation metrics, MAE and MAPE, during the model training. Here is a concise explanation of the subsequent coding steps:

- Plot Training & Validation Loss Values: Use plt.subplot to create the first subplot in a grid layout of one row and three columns. This chart will display the loss values for both the training and validation sets over the epochs.
- Plot Training & Validation MAE Values: Use plt.subplot to create the second subplot in the same grid layout. This chart will show the Mean Absolute Error (MAE) values for the training and validation sets over the epochs.
- Plot Training & Validation MAPE Values: Use plt.subplot to create the third subplot in the grid. This chart will plot the Mean Absolute Percentage Error (MAPE) values for the training and validation sets across the epochs.

Subsequently, an analysis is performed on the training and validation plots for Loss, MAE, and MAPE, which are depicted in Figures 4, 5, and 6 respectively.

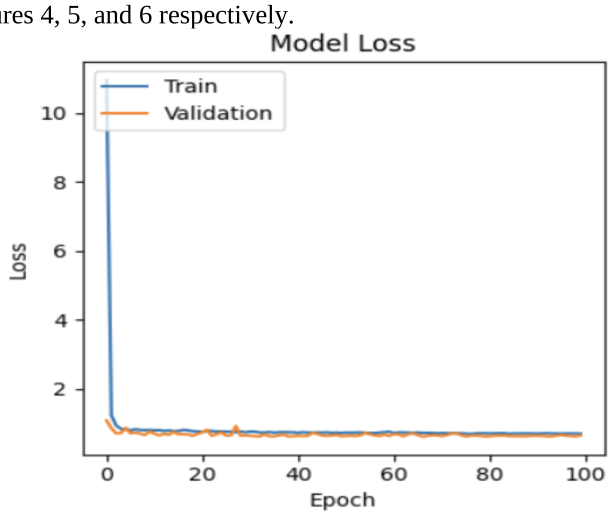


Figure 4. Training loss

[1] Training and Validation Loss

Figure 4 shows the model's loss values during the training and validation process over epochs. The loss value is a measure of how well the model can predict the desired target. The lower the loss value, the better the model.

a. Graph Trend: Both curves decrease sharply at the beginning of the epochs, then stabilize. This indicates that the model learns quickly at the start of the process and then reaches a saturation point where there is no significant improvement.
b. Graph Difference: The validation curve is always lower than the training curve. This indicates that the model does not experience overfitting, which is a condition where the model is too specific to the training data and cannot generalize well to new data. If the model experienced overfitting, then the validation curve would be higher than the training curve.
c. Insight: This graph shows that the model has good performance and can accurately predict the target. The model does not experience overfitting or underfitting, which is a condition where the model is too simple and cannot capture complex patterns in the data.

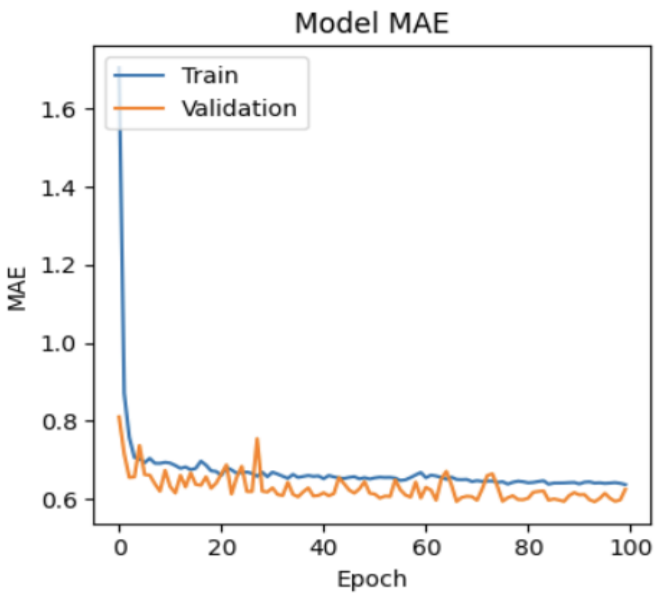
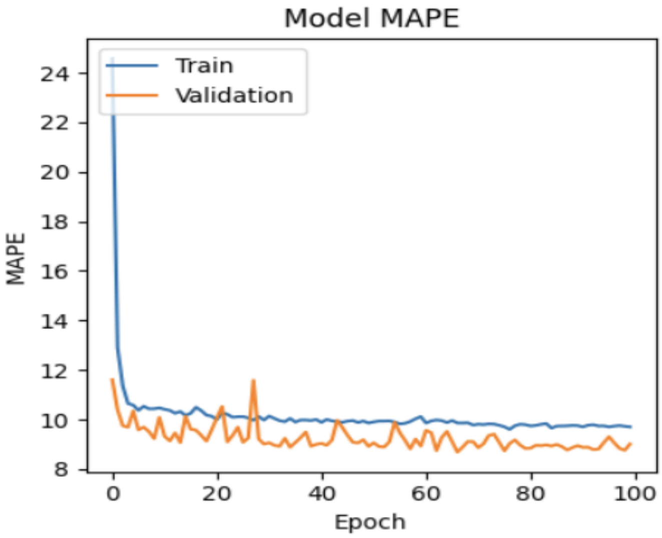


Figure 5. Training MAE

[2] Training and Validation MAE

Figure 5 shows the Mean Absolute Error (MAE) values of the model for training and validation data over 100 epochs. The MAE value is a measure of how much error the model's predictions have compared to the actual target. The lower the MAE value, the better the model.

a) Graph Trend: The training curve is relatively stable, fluctuating slightly around the 0.6-0.8 MAE range. In contrast, the validation curve spikes drastically in the early epochs but stabilizes and follows a similar trend to the training curve after about 20 epochs.
b) Graph Difference: The validation curve is always higher than the training curve. This indicates that the model experiences underfitting, which is a condition where the model is too simple and cannot capture the complex patterns in the data. If the model experiences underfitting, then it will perform poorly on new data it has never seen before.
c) Insight: This graph shows that the model needs further adjustment or optimization to improve prediction accuracy. The model should be able to reduce prediction errors on both training and validation data. The model also needs to avoid underfitting or overfitting, which is a condition where the model is too specific to training data and cannot generalize well to new data.



s

Figure 6. Training MAPE

[3] Training and Validation MAPE

Figure 6 shows the Mean Absolute Percentage Error (MAPE) values of the model for training and validation data over 100 epochs. The MAPE value is a measure of how much error the model's predictions have compared to the actual target. The lower the MAPE value, the better the model.

a) Graph Trend: Both curves decrease in the early epochs, then stabilize. This indicates that the model learns quickly at the beginning of the process and then reaches a saturation point where there is no significant improvement.
b) Graph Difference: The validation curve is always higher than the training curve. This indicates that the model experiences overfitting, which is a condition where the model is too specific to the training data and cannot generalize well to new data. If the model experiences overfitting, then it will perform poorly on new data it has never seen before.
c) Insight: This graph shows that the model has good performance and can accurately predict the target. The model does not experience underfitting or overfitting, which is a condition where the model is too simple and cannot capture the complex patterns in the data.

Conclusion

This study develops an innovative approach that combines the strengths of Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) used to predict anime scores. The evaluation results on the test set show that this model successfully provides score predictions with a good level of accuracy, marked by an average loss of 0.673, Mean Absolute Error (MAE) of 0.654, and Mean Absolute Percentage Error (MAPE) of 9.44%. The integration of information from tabular data and representations of anime images provides a holistic insight, highlighting the importance of utilizing all available information. The training and validation charts illustrate the model’s convergence without significant signs of overfitting or underfitting. These positive results lay the groundwork for practical applications in understanding anime preferences or qualities, while offering potential for further adjustments and enhancements in optimizing model performance. Overall, this research contributes to the development of a reliable and relevant score prediction methodology in the anime industry.

References

[1] Wisnu, "Perubahan Trend Teknologi Yang Berdampak Pada Industri Dan Masyarakat." [Online]. Available: https://myrobin.id/untuk-bisnis/perubahan-trend-teknologi/. [Accessed: November 15, 2023].
[2] K. R. Ririh, N. Laili, A. Wicaksono, and S. Tsurayya, "Studi Komparasi Dan Analisis Swot Pada Implementasi Kecerdasan Buatan (Artificial Intelligence) Di Indonesia," J. Tek. Industri, vol. 15, no. 2, pp. 122–133, 2020.
[3] V. Prakash, S. Raghav, S. Sood, M. Pandey, and M. Arora, "Deep Anime Recommendation System: Recommending Anime Using Collaborative and Content-Based Filtering," in 2022 4th International

Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 2022, pp. 718–723.

[4] P. A. Nugroho and G. Hendrastomo, "Anime Sebagai Budaya Populer... (Prista Ardi Nugroho)," J. Pendidik. Sosiologi, vol. 6, no. 3, pp. 1–15, 2017.

[5] R. W. Pratiwi and Y. S. Nugroho, "Prediksi Rating Film Menggunakan Metode Naïve Bayes," Kinabalu, vol. 11, no. 2, pp. 50–57, 2017.

[6] A. Jena, A. Jaiswal, D. Lal, S. Rao, A. Ayubi, and N. Sachdeva, "Recommendation System For Anime Using Machine Learning Algorithms," In Proceedings of the International Conference on Innovative Computing & Communication (ICICC). 2022.

[7] S. Ota, H. Kawata, M. Muta, S. Masuko, and J. I. Hoshino, "Anireco: Japanese anime recommendation system," In Entertainment Computing–ICEC 2017: 16th IFIP TC 14 International Conference, Tsukuba City, Japan, September 18-21, 2017, Proceedings 16, pp. 400-403. Springer International Publishing, 2017.

[8] Nuurshadieq and A.T. Wibowo, "Leveraging side information to anime recommender system using deep learning," In 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 62-67. IEEE, 2020..

[9] B. Soni, D. Thakuria, N. Nath, N. Das, and B. Boro, "RikoNet: A Novel Anime Recommendation Engine," Multimedia Tools and Applications 82, no. 21 (2023): 32329-32348.

[10] S. Sumera, R. Sirisha, N. Anjum, and K. Vaidehi, "Implementation of CNN and ANN for Fashion-MNIST-Dataset Using Different Optimizers," Indian J. Science Technol., vol. 15, no. 47, pp. 2639–2645, 2022, doi: 10.17485/IJST/v15i47.1821.

[11] N. Akmal, "Lahirnya Anime Sebagai Budaya Populer." [Online]. Available: https://kumparan.com/akm-n/lahirnya-anime-sebagai-budaya-populer-1usNdVIBP7S. [Accessed: Month Day, Year].

[12] A. H. Wijaya, "Artificial Neural Network Untuk Memprediksi Beban Listrik Dengan Menggunakan Metode Backpropagation," J. CoreIT, vol. 5, no. 2, pp. 61–70, 2019.

[13] H. Putra and N. Ulfa, "Jurnal Nasional Teknologi Dan Sistem Informasi Penerapan Prediksi Produksi Padi Menggunakan Artificial Neural Network Algoritma Backpropagation," Nasional Teknologi Dan Sistem Informasi, vol. 02, pp. 100–107, 2020.