

Article

Study on Dynamic Evaluation of Sci-tech Journals Based on Time Series Model

Yan Ma ^{1,*} , Yingkun Han ¹, Mengshi Chen ² and Yongqiang Che ¹¹ State Grid Shandong Electric Power Research Institute, Jinan 250003, China² School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China

* Correspondence: yanpony@126.com

Abstract: As science and technology continue to advance, sci-tech journals are developing rapidly, and the quality of these journals affects the development and progress of particular subjects. Whether sci-tech journals can be evaluated and predicted comprehensively and dynamically from multiple angles based on the current qualitative and quantitative evaluations of sci-tech journals is related to a rational adjustment of journal resource allocation and development planning. In this study, we propose a time series analysis task for the comprehensive and dynamic evaluation of sci-tech journals, construct a multivariate short-time multi-series time series dataset that contains 18 journal evaluation metrics, and build models based on machine learning and deep learning methods commonly used in the field of time series analysis to carry out training and testing experiments on the dataset. We compare and analyze the experimental results to confirm the generalizability of these methods for the comprehensive dynamic evaluation of journals and find the *LSTM* model built on our dataset produced the best performance (MSE: 0.00037, MAE: 0.01238, accuracy based on 80% confidence: 72.442%), laying the foundation for subsequent research on this task. In addition, the dataset constructed in this study can support research on the co-analysis of multiple short time series in the field of time series analysis.



Citation: Ma, Y.; Han, Y.; Chen, M.; Che, Y. Study on Dynamic Evaluation of Sci-tech Journals Based on Time Series Model. *Appl. Sci.* **2022**, *12*, 12864. <https://doi.org/10.3390/app122412864>

Academic Editor: Luigi Portinale

Received: 19 November 2022

Accepted: 13 December 2022

Published: 14 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: time series; machine learning; deep learning; sci-tech journals; dynamic evaluation; multivariate; multi-series; a new dataset

1. Introduction

With the continuous development of science and technology, sci-tech journals, as carriers of academic content, have also developed rapidly, and their importance as links in the national innovation system has increased. The quality of sci-tech journals affects the development and progress of subjects; thus, the development of subjects requires the scientific and systematic evaluation of journals. The evaluation and analysis of scientific journals can not only improve the quality and influence of scientific journals but can also facilitate researchers in understanding current research priorities and hot spots.

In sci-tech journal evaluation research, compared with qualitative evaluation [1] methods, which are more heavily influenced by subjective coloring, methods for calculating multiple metrics for quantitative evaluation based on the derivation of mathematical formulas have gradually become the mainstream direction of journal evaluation research. Quantitative evaluation methods initially focused on single metrics, with metrics such as the Price Index [2], the law of literature concentration [3], and the impact factor [4] being developed, which would then be combined with single metrics such as the H-index [5] and impact factor and assigned metric weights by principal component analysis [6] and the entropy weighting method [7] to obtain comprehensive evaluation metrics. At present, some studies focus on the scenario of “data-based evaluation at the time” and introduce artificial intelligence into journal evaluation, such as through the application of artificial intelligence to assign metric weights and for the effective selection of evaluation methods

to assist in journal evaluation tasks [8]. We focus on the task of “comprehensive dynamic evaluation of sci-tech journals based on historical journal evaluation metrics”, build a new sci-tech journal evaluation metrics dataset, and try to analyze the possible future influence of sci-tech journals by comprehensively analyzing various metrics through using artificial intelligence in journal evaluation studies to guide the future development of sci-tech journals.

In this study, we construct a new time series dataset with 18 journal evaluation metrics; propose applying machine learning and deep learning methods commonly used in the field of time series analysis to the task of sci-tech dynamic journal evaluation based on this dataset; construct models based on each of the nine mainstream methods in the field of time series analysis to conduct training and testing experiments on our multivariate short-time multi-series dataset; and provide evaluation metrics for these models. The final results confirm the generalizability of these methods for the comprehensive dynamic evaluation of sci-tech journals and show the LSTM model built on our dataset produces the best performance (MSE: 0.00037, MAE: 0.01238, accuracy based on 80% confidence: 72.442%). Based on the results of this study, a variety of practical application strategies can be extended, which can serve as a guide for rational resource allocation and for the subsequent development planning of sci-tech journals.

Contributions:

1. We propose a sci-tech journal evaluation task for the dynamic evaluation of scientific journals based on time series analysis.
2. Based on the collection and analysis of data from sci-tech journal publishing platforms, such as WanFang and ZhiWang, we construct a time series dataset of journal metrics, which is a multivariate short-time multi-series time series dataset. Our dataset provides data support and certain challenges to the field of multi-series time series data collaborative analysis.
3. We build models using nine mainstream time series data analysis methods of machine learning and deep learning to conduct various experiments on the task and dataset proposed in this study, and the results are compared and analyzed in detail.
4. We confirm that these nine methods are generalizable to the task of the comprehensive dynamic evaluation of scientific journals and find the LSTM model built on our dataset produces the best performance, laying the foundation for the subsequent optimization of algorithms and serving as a directional guide for this task.

The remainder of the article is organized as follows: Section 2 briefly introduces the development status of journal evaluation, classical time series datasets, and time series data analysis methods; Section 3 presents the details of the new time series dataset we propose in this study, the processing of experimental input data, and the metrics for our models; in Section 4, we introduce the details of the selected methods and the construction of the model for training and testing experiments. In Section 5, we present a detailed and comprehensive comparison and analysis of various results. Finally, in Sections 6 and 7, we discuss the possible applications and development directions of subsequent research and conclude the study.

2. Related Works

2.1. Evaluation of Sci-tech Journals

2.1.1. Development of Sci-tech Journals

After the first scientific journals were produced in 1665, the concept of core journals was first introduced by British bibliographers in 1934, giving strong impetus to the development and growth of sci-tech journals. The theoretical basis of sci-tech journal evaluation consists of three classical theories of bibliometrics: The “law of literature dispersion”, the “law of literature aging metric and citation peak”, and the “law of citation concentration”. The law of literature dispersion divides journals into core, relevant, and peripheral zones [9]; the metric of literature aging reflects the development rate of the subject in which the literature is located; the law of citation peaks provides important theoretical

support for constructing impact factors [10]; and the law of citation concentration verifies the law of document dispersion from the perspective of citation and further extends it to the distribution law of journals based on citations, thereby providing a key theoretical basis for the selection of core journals [11].

2.1.2. Evaluation Methods of Sci-tech Journals

At present, there are two main ways in which to evaluate sci-tech journals: Qualitative evaluation and quantitative evaluation.

Qualitative evaluation is often based on peer review, which ensures the subjective nature of the journal evaluation process. At present, UTD24 in the US, ABDC in Australia, ABS, FT50 in the UK, and CNRS in France are some of the more prominent lists of peer-reviewed journals.

Single metrics related to the publishing behavior of journals represent some of the most important quantitative evaluation methods. Common classifications include citation metrics represented by the journal impact factor (JIF), citation metrics represented by the H-index, citation metrics represented by page rank, citation metrics represented by 'altimetric' [12], etc. At present, the main evaluation metrics are the journal impact factor, the 5-year impact factor, the total number of citations, and the citation half-life of the journal. The final dependent variable of this study is the extended impact factor.

2.1.3. Current Status of Studies on Sci-tech Journal Evaluation Metric Analysis

Based on the current situation of the qualitative evaluation of scientific journals, the research on journal evaluation metric analysis mainly includes two categories: Individual journal metric research and overall journal metric research. For individual journal metrics, scholars at home and abroad currently pay more attention to the volume of articles, mainly focusing on the relationship between academic influence and trend changes. For example, the authors of [13–15] studied the relationship between article volume and the influence of different journal types, etc. As for the overall study of journal metrics, the research mainly focuses on the relationship with journal influence. On the other hand, it is mainly related to the application of source metrics to the evaluation system. The authors of [16] studied the relationship between the citation evaluation metrics of sci-tech journals from several perspectives through comparative analysis. The authors of [17] proposed a nonlinear academic evaluation research method based on the backpropagation artificial neural network, and their results better verified its applicability.

In summary, there are few journal evaluation metric analysis studies that use artificial intelligence methods to learn journal metric time series data in order to build dynamic evaluation models, and there are few datasets available in this field. To improve this situation, this study proposes the use of artificial intelligence methods in the field of journal evaluation to try to solve the task of the comprehensive dynamic evaluation of journals, build a time series dataset of journal metrics after collecting and analyzing journal metric data, select a variety of artificial intelligence methods to learn journal metric time series data to build dynamic evaluation models, and then test and evaluate the models in order to verify their feasibility. Finally, the feasibility and applicability of the research directions are verified.

2.2. Time Series Dataset

We define time series data as data received at different times, describing changes in one or more characteristics over events. There are two main types of datasets commonly used in the field of time series data analysis: Univariate time series datasets and multivariate time series datasets. Univariate time series datasets are simpler to understand as they can be easily plotted, and a method can be quickly tried and evaluated; multivariate time series datasets can carry richer information content for method mining analysis but are therefore more challenging. In the following, we will briefly describe several datasets for comparison with the new dataset constructed in this paper.

2.2.1. Univariate Time Series Datasets

Daily minimum temperature dataset. This dataset describes the minimum daily temperature in degrees Celsius for each day over the course of 10 years (1981–1990) in the city of Melbourne, Australia, with 3650 observations, comprising data from the Australian Bureau of Meteorology. The dataset has a strong seasonal component.

Shampoo sales dataset. This dataset describes the number of shampoos sold per month from sales counters over a three-year period, with 36 observations, and was provided by Makridakis, Wheelwright, and Hyndman (1998). The dataset as a whole shows an increasing upward trend and may contain a seasonal component.

Daily female birth dataset. This dataset describes the number of female births on each day in California in 1959, with 365 observations, and was provided by Newton (1988).

Monthly sunspot dataset. This dataset describes the monthly counts of the number of sunspots observed for the last 200 years, with 2820 observations, comprising data provided by Andrews and Herzberg (1985). This dataset has large differences between seasons and exhibits seasonality.

2.2.2. Multivariate Time Series Datasets

Room occupancy detection dataset. This dataset describes 20,560 min of observations of whether a room is occupied every minute, with seven attributes, including room lighting, climate, and other attributes, providing a classification prediction problem. The data were provided by Luis Candanedo of UMONS.

Ozone level detection dataset. This dataset describes ground-level ozone concentration observations for each day over the course of 6 years in order to predict whether a particular day is an “ozone day” or not. The dataset contains 2536 observations and 73 attributes and provides a classification prediction problem.

EEG eye state dataset. This dataset describes the EEG data of individuals and whether their eyes are open. The goal of the problem is to predict eye states based on EEG data, with a total of 14,980 observations and 15 covariates.

It is worth noting that the commonly used datasets described above are single series, while the time series dataset of journal metrics constructed in this study provides multivariate, short-time, and multi-series data. Multivariate data can carry more information for mining and short-time data require the method employed to have a stronger implied information mining ability, while multi-series data require the method to have a certain data association analysis ability. Therefore, the dataset constructed in this study will pose a greater challenge than a multivariate time series dataset.

2.3. Time Series Analysis Methods

Based on time series datasets, the time series analysis method is intended to analyze and predict possible future data by mining and analyzing the implied correlation between data of multiple dimensions and their correlation characteristics over time, using historical data [18].

Time series analysis methods can all be regarded as regression problems, but there are some differences in regression methods, which can be divided into traditional statistical methods, machine learning methods, and deep learning methods. These three types of methods are briefly introduced below.

2.3.1. Traditional Methods

The traditional methods, represented by ARMA/ARIMA [19] methods, can only model and predict using smooth data and therefore need to differentiate the data first. The traditional methods are still simple linear models based on the derivation of mathematical formulas and statistical laws for time series analysis and prediction, which require the forms of the data themselves, which must also be high quality. When dealing with a univariate time series to analyze and predict problems, the traditional method performs well; however, if there are too many variables, it is not applicable.

2.3.2. Machine Learning Methods

Machine learning methods, represented by LightGBM [20] and XGBoost [21] methods, often convert the time series problem into a supervised learning problem and use machine learning methods to analyze and predict the time series after processing the data through feature engineering. These methods can solve most complex time series analysis problems and can support complex data modeling, multivariate co-regression, nonlinear problems, etc.; however, the more complex feature engineering aspect needs to be performed manually, and feature engineering determines the upper limit of machine learning methods. When feature engineering is completed, the machine learning method is fast and accurate in terms of computation. It does not possess high requirements for problems such as missing data values.

2.3.3. Deep Learning Methods

Deep learning methods are able to design a variety of deep learning method architectures based on practical problems due to the fact that the network structure can be freely designed. Among them, LSTM/GRU [22], as a variant of RNN [23], has been designed for solving time series problems, while WaveNet [24], Conv-1D [25], and other CNN-based methods have been developed and can also be applied to time series data. In addition, recent research has been carried out in relation to collaborative analysis and the prediction of multi-series data. In general, deep learning methods are more suitable for problems with large amounts of data, support online training, do not require manual feature engineering, and can learn deeper semantic and relative relationship information from data; however, their requirements with regard to input data are higher than machine learning methods, and require data pre-processing, such as missing value filling and normalization, in advance.

The time series dataset of journal metrics constructed in this study contains multiple series and multiple covariates, and each time series is short. Based on the characteristics of the dataset, it is difficult to apply traditional methods to the dataset, while machine learning methods and deep learning methods can be applied to it via their powerful adaptability to perform experimental analysis in relation to the dynamic evaluation task of journals proposed in this paper. In addition, the multivariate short-time multi-series dataset constructed in this study can also support the recent research on multiple-series co-analysis in the field of time series analysis.

3. Dataset and Metrics

3.1. Dataset

The journal metric time series dataset contains 5425 journals, and each journal corresponds to a time series consisting of five consecutive years of metric records for the same journal, with each metric record consisting of 21 categories of elements. To ensure diversity of the data sources in the dataset, metric records of journals for five consecutive years were collected from a total of 152 subdivisions, with 12 broad subject categories.

Data collection, cleaning, and collation. As shown in Figure 1, we collected the contents of the dataset from a total of 152 subdivisions (e.g., the medical field, the engineering field, etc.), with 12 disciplinary categories. The proportion of journals belonging to each of the 12 disciplinary categories in the dataset is shown in Figure 2. The dataset contains 5425 journals, each journal corresponds to a time series with five timesteps, and the contents comprise 21 types of metric records for each journal, for a total of 5 years, from 2017 to 2021. We first collected a large number of journal metric records by year from journal platforms such as WanFang and ZhiWang and used Excel records; then, we filtered the journals containing five consecutive years of records based on journal names and unique identifiers, stitched them according to journal names and unique identifiers, and sorted them according to journal names and years of records. Finally, we performed cleaning operations such as missing value filling and text transformation on the stitched data: For missing values, we used the more common mean-fill method, i.e., we used the mean value

of the category metric for the rest of the year of a journal, or directly filled it with 0 if all five years were missing; for text, we converted this into numerical elements as model inputs. The result of these steps is the time series dataset of journal metrics presented in this paper.

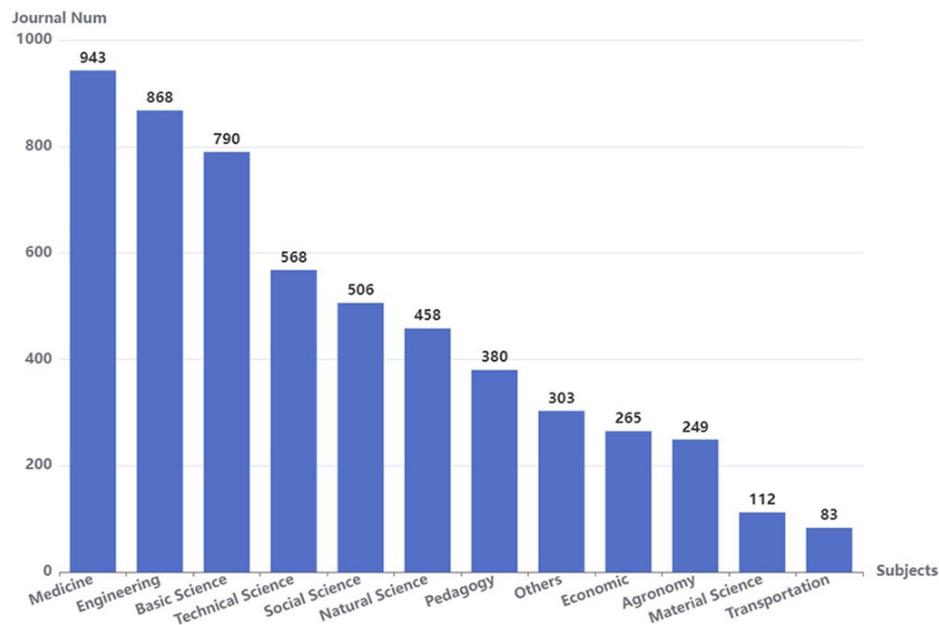


Figure 1. Bar of classes of subject statistics for the time series dataset of journal metrics.

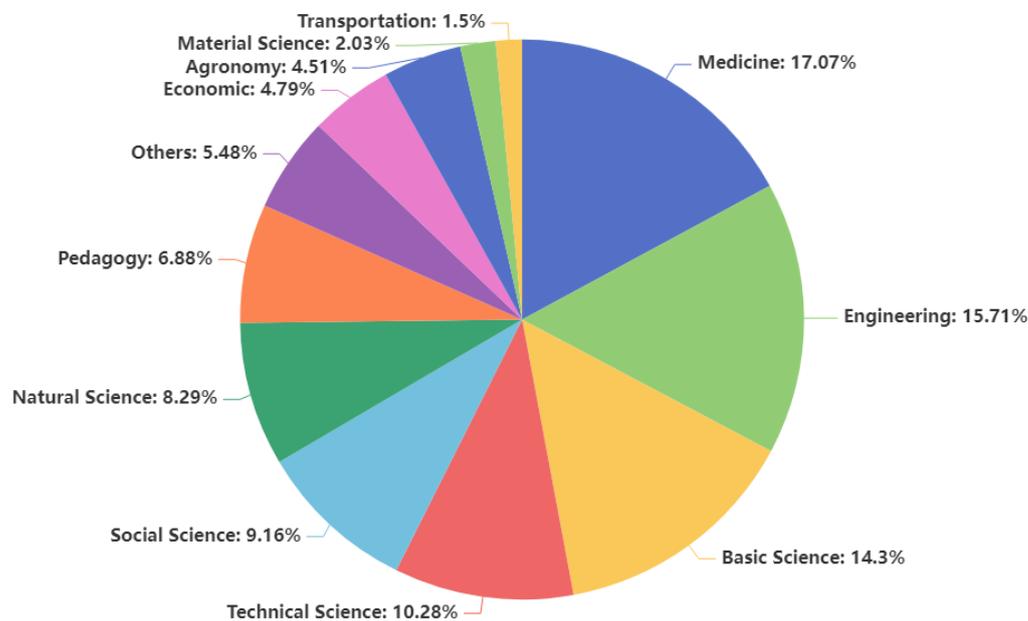


Figure 2. Pie chart of the proportion of journals belonging to each of the 12 disciplinary categories in the journal metric time series dataset.

Metric record categories. Each journal metric record contains three categories of journal overview attributes (journal name, journal subject classification, and metric record year) and 18 categories of journal evaluation metrics (regional distribution, grant ratio, the average number of authors, average number of citations, citation half-life, extended H metric, extended other citation rates, extended immediate-year metric, extended disciplinary impact metric, extended disciplinary diffusion metric, extended number of cited journals, extended total citation frequency, extended citation half-life, literature election rate, number

of institutional distribution, volume of source literature, overseas paper ratio, and extended impact factor). Among them, the overview attributes are used to distinguish between the data for detailed analyses in this study; the journal evaluation metrics are used as core time series data for training and testing the models constructed by various time series analysis methods.

Time series characteristics. Multiple multivariate, short-time series are included in the journal metrics time series dataset. Each journal in this dataset contains metric records for each of the last five years and thus can be viewed as a sample of 5425 time series of 5 time-step 18-dimensional variables, which can support experiments using multiple time series prediction methods. Since the variables in this dataset are evaluation metrics for journals by year, it is difficult to show obvious seasonality as a time series, and its multiple time series and multiple covariates make certain demands on the mining learning ability and association learning ability of the modeling algorithm. Therefore, there are some difficulties in the research of modeling algorithms for this dataset.

Normalization. To support both machine learning and deep learning time series prediction methods and to facilitate the comparison and weighting of metrics of different units or magnitudes, we changed the various metrics in the time series dataset of journal metrics from dimensioned expressions to dimensionless expressions, i.e., we normalized the data for each column according to Equation (1), a more comprehensive normalization equation referring to scikit-learn (a Python library) feature-scaling methods [26].

$$X_{scaled} = \frac{X - X.min}{X.max - X.min} \times (Max - Min) + Min \quad (1)$$

where X_{scaled} is the final normalized result, X represents the data to be normalized, $X.max / X.min$ are the maximum and minimum values of the columns in which the data are located, and Max / Min are the maximum and minimum values of the mapping interval. The essence of normalization is the operation of mapping data points to a given interval: For an element in a column, first, we must find the ratio of the difference between the minimum value and the maximum value of the column and then use the ratio to obtain the value in the given mapping interval. It is worth noting that if we set the normalization mapping range to $[0, 1]$ and set Max to 1 and Min to 0, then Equation (1) can be transformed into the most general normalization equation.

Experimental data processing. To construct inputs that can be applied to the time series prediction method and to ensure a consistent data size, we processed the dataset in the form of $3 + 1$, with the extended impact factor as the target, i.e., the extended impact factor of the fourth year was predicted using the time series data of the last three years, and—finally—8680 training samples and 2170 test samples were generated.

For the machine learning method, which receives one-dimensional inputs of independent variables, as shown in Figure 3, we stitched 18 types of evaluation metrics of the same journal in three years in one dimension to obtain an input length of 54. For the ANN method in deep learning, the input is processed in the same way as machine learning.

For the rest of the deep learning methods, the received independent variable input is two-dimensional with the dimension of the timesteps \times and the number of features as shown in Figure 4. We used 18 categories of evaluation metrics from the same journal for three years for two-dimensional stitching to obtain the 3×18 input.

At this point, the training and test preparation for the experimental phase are complete.

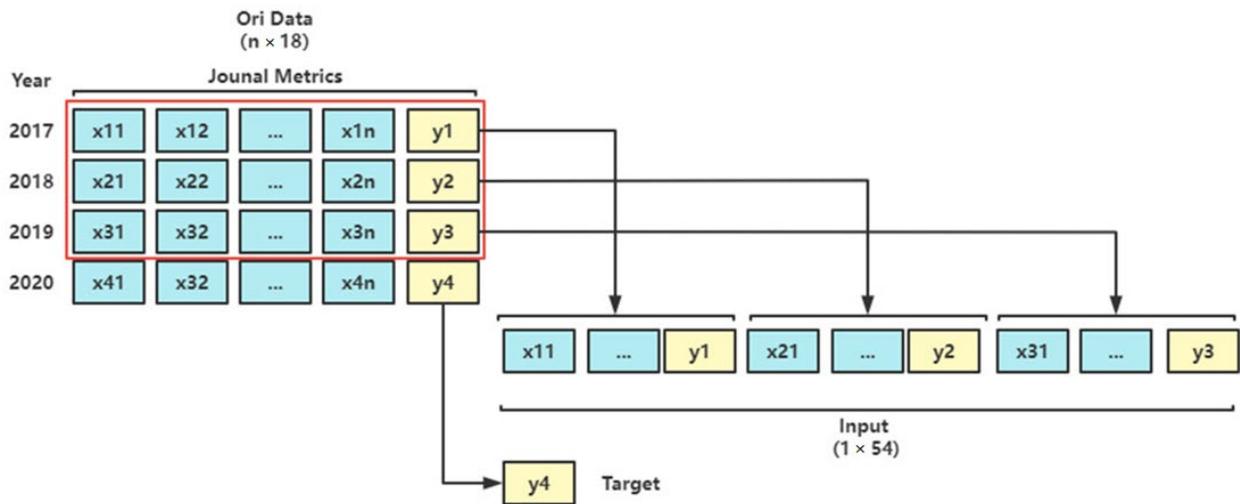


Figure 3. Machine learning method training for data processing.

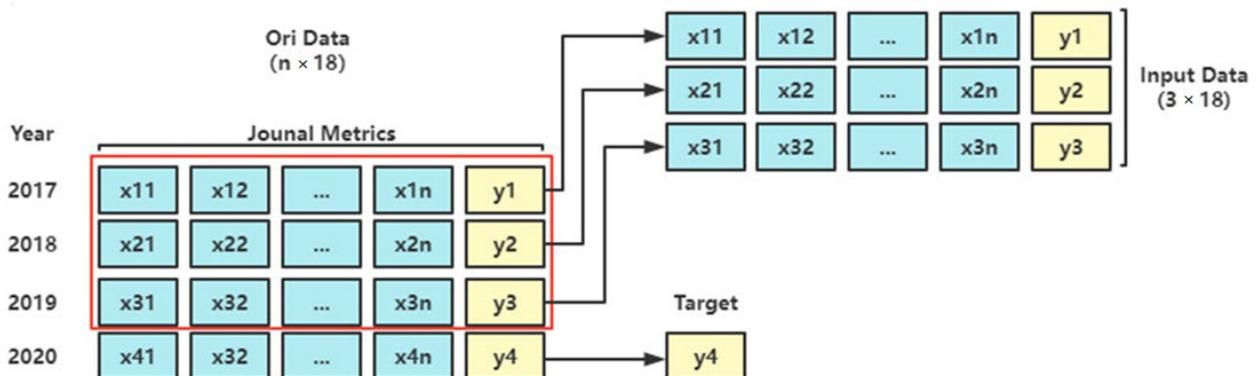


Figure 4. Deep learning method training for data processing.

3.2. Metrics

In time series prediction tasks, the diversity of evaluation metrics can provide a multi-faceted reference for assessing model performance. Considering that each method has its corresponding advantages and disadvantages, we chose five commonly employed evaluation metrics to evaluate the effectiveness of the time series prediction model constructed based on this study’s journal metrics dataset in the dynamic journal evaluation task: MSE (mean squared error) [27], MAE (mean absolute error) [28], RMSE (root mean square error) [29], MAPE (mean absolute percentage error) [30], and NRMSE (normalized root mean square error) [31]. The smaller the value, the higher the accuracy of the model. We evaluated each method in our experiments using all of the above-mentioned metrics and analyzed the applicability of the methods by comparing them with one another.

Among them, the mean absolute error (MAE) reflects the actual situation of the error of the predicted value but relies on a scale, while the mean absolute percentage error MAPE overcomes the disadvantage of relying on a scale, but we did not include a test sample with an actual value of 0 in the calculations during processing; the mean square error (MSE), root mean square error (RMSE), and standardized root mean square error (NRMSE) are all measures that reflect the degree of difference between the actual value and the predicted value, which are increasingly sensitive to outliers.

The evaluation metrics are calculated in Equations (2)–(6), where n is the number of samples in the test set, y_i is the actual value of the target for the i -th sample, \hat{y}_i is the predicted value of the target given by the prediction model after receiving the input of the

i -th sample, and y_{max} / y_{min} are the maximum and minimum values of the target columns in the test set samples, respectively.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (4)$$

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (5)$$

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}}{y_{max} - y_{min}} \quad (6)$$

In addition, we designed an accuracy metric based on the idea of confidence in order to support the analysis and allowed the number of samples in the test set to be n . For the x confidence level, when the absolute error range between the predicted value of i -th sample \hat{y}_i and the actual value y_i does not exceed $(1 - x)$, it should be considered that the sample is correctly predicted, and vice versa. Finally, we calculated the accuracy metric according to Equation (7).

$$\text{score} = \frac{\sum_{i=1}^n \begin{cases} 1, & \text{if } \frac{|\hat{y}_i - y_i|}{y_i} < 1 - x \\ 0, & \text{else} \end{cases}}{n} \quad (7)$$

4. Methods

We selected the most mainstream methods in the field of machine learning and deep learning for general time series analysis tasks, including multiple linear regression, random forest, XGBoost, LightGBM, ANN, Conv-1D CNN, WaveNet, LSTM, and GRU, and then built and trained the corresponding models to conduct tests on our dataset.

The predictions of traditional time series prediction methods are based on the exploration of mathematical–statistical laws for univariate long-time single-series time series data, which are difficult to apply to a multivariate short-time multi-series time series dataset of journal metrics, so traditional methods such as ARMA, ARIMA, and Prophet were not introduced or experimented upon in this study.

In Table 1, we briefly discuss how existing studies apply these methods to time series forecasting tasks.

In the subsequent part of this chapter, we will introduce the principle of each method and its modeling under our new time series forecasting task and new time series dataset.

4.1. Machine Learning Methods

As shown in Figure 3, we processed the journal metrics time series dataset into a one-dimensional input vector with a single target in the form of supervised learning in order to train the prediction model built by the machine learning method.

Table 1. Brief discussions of how existing studies apply these methods to time series forecasting tasks.

Methods	Existing Studies of Time Series Forecasting Tasks
Multiple Linear Regression	Ng K Y, Awang N use Multiple Linear Regression time series model to forecast PM10 concentrations in Peninsular Malaysia [32].
Random Forest	Kane M J, Price N, Scotch M, et al. use Random Forest time series model to predict avian influenza H5N1 outbreaks [33].
XGBoost LightGBM	Shehadeh A, Alshboul O, Al Mamlook R E, et al. use machine learning models (XGBoost and LightGBM) to predict the residual value of heavy construction equipment [34].
ANN	Alzain E, Alshebami A S, Aldhyani T H H, et al. use ANN to predict housing prices [35]; Ecer F, Ardabili S, Band S S, et al. train ANN to predict stock price index [36].
Conv-1D CNN	Cai M, Pipattanasomporn M, Rahman S. use Conv-1D CNN model in deep learning to forecast day-ahead building-level load [37].
WaveNet	Borovykh A, Bohte S, Oosterlee C W. use WaveNet to forecast conditional time series [38].
LSTM	Ahmed A A, Deo R C, Ghahramani A, et al. integrate LSTM and Random Forest to estimate soil moisture [39].
GRU	Yamak P T, Yujian L, Gadosey P K. use GRU to forecast time series data with Bitcoin's price dataset as their time series dataset [40].

4.1.1. Multiple Linear Regression

When there is a linear relationship between the independent and dependent variables, the method of building a prediction model for correlation analysis of two or more independent variables with one dependent variable is called linear regression. Its general form is as follows:

$$\hat{y}_i = \theta_0 + \theta_1 X_1^i + \theta_2 X_2^i + \dots + \theta_n X_n^i \quad (8)$$

In Equation (8), θ_0 is a regression constant, θ_n is the regression coefficient, and X_n^i is the n th variable of the i -th set of observations, for which exact measurements can be obtained from data collection. It is usually referred to as the independent variable or covariate in time series prediction tasks. \hat{y}_i is the predicted value of the i -th set of observations, which is called the dependent variable [41].

This method is arguably the simplest in machine learning, but it is computationally simple and effective, so we include it in our experiments with $\sum_{i=1}^n (y_i - \hat{y}_i)^2$ as the loss function, m as the number of samples, and y_i as the actual value of the i -th set of observations. The training is performed by using the journal metrics time series dataset. We consider the training of the multiple linear regression model complete when the value of the loss function is below the set threshold, or the reduction in multiple training periods is below the set threshold.

For the dynamic evaluation task of the journals in this study, the model is set as $y = f(x)$, the evaluation metrics of a journal for three years are stitched into a one-dimensional vector as the input, and the dynamic evaluation of the journal is completed when we obtain the corresponding \hat{y}_{i+1} .

4.1.2. Random Forest

The method is essentially an integrated learning method; it integrates many decision trees into a forest and uses the average values of the predicted results to predict the final outcome. The random forest method is insensitive to multivariate covariance, is more robust to missing data and unbalanced data, and can predict the effects of up to several thousand explanatory variables well, even when the input data are not normalized [42].

In the journal metrics time series dataset, the X vector is the independent variable of the input model (a journal with three years of metrics data stitched together with a length of 54), Y is the dependent variable to be predicted, the training set φ is randomly generated in (X, Y) , the predicted result is set to $f(X)$, and the mean square generalization error is

$E_{x,y}[Y - f(X)]^2$. The final regression function of the random forest method is shown in Equation (9).

$$Y = E_{\varphi} f(X, \varphi) \quad (9)$$

In our experiments, we set the maximum number of iterations of the weak learner to 500 and could use all of the feature values for partitioning when generating the decision tree. We also used the bootstrap method to randomly select the sample set with a release (the random forest is a bagging type algorithm), and each sample set grew into a single decision tree with each leaf node containing one prediction. We chose to use the mean square error (MSE) metric $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$ with the L2 regularization term as the loss function and a measure of branch quality, and the selection of features was based on the difference in the value of this function between the father and leaf nodes.

Considering that the dynamic journal evaluation task can be understood as a regression task under supervised learning, we divided the whole feature space X into many non-overlapping regions and used its mean value as the predicted result for all samples in the same region, hoping to find a way in which to divide the regions so that the loss was as low as possible. For the partitioning scheme, we used recursive slicing to exhaust the possible partitioning thresholds for each feature when branching in order to find the optimal slicing feature and the optimal slicing point threshold. We stopped when the branching reached the preset terminating conditions (e.g., the upper limit of the number of leaves, loss threshold, etc.). When making predictions, the means of all the results of the decision tree predictions on the input variables were used as the final prediction results.

As described above, we trained a regression model built by the random forest method based on a time series dataset of journal metrics for the dynamic evaluation task of journals.

4.1.3. EXtreme Gradient Boosting (XGBoost)

XGBoost (eXtreme gradient boosting) is an optimized distributed gradient boosting algorithm based on the boosting framework. Compared with random forest with a put-back random sampling bagging strategy, the boosting framework uses the whole dataset in generating each decision tree. This method is very powerful in terms of its parallel computational efficiency, missing value handling, control of overfitting, and prediction generalization performance, and it has wider applications in time series prediction tasks.

The basic component of XGBoost is also a decision tree, but unlike the random forest method, there is a before-and-after dependency between the decision trees in XGBoost, and the generation of the latter decision tree combines the deviation of the previous one so that the predicted value of the k th tree for the input sample is the sum of the predicted value of the previous $k-1$ decision trees and the predicted value of the k th tree [21].

In the experiment, for the k th decision tree, we set the loss function as the MSE metric between the predicted result and actual value of the dataset in the k th decision tree and the complexity of the k th tree (consisting of the number of leaf nodes and the L2 regularization factor of the node weights). In generating the decision tree, the leaf nodes were set to the value that minimized the objective function of the tree. For each input sample of journal metrics, we used the weighted partitioning method proposed by XGBoost to rank the 54 feature values to obtain the appropriate cut points, and then, based on the greedy algorithm, we attempted to use the cut points at each node to generate a new tree in order to obtain the optimal division with the smallest loss function value and with fewer overheads. At the same time, after one iteration, we multiplied the weights of the leaf nodes by a reduction factor to reduce the impact of each tree, in order to create more learning space. We set the maximum number of iterations of the weak learner to 1000, and the model was considered to be trained when the value of the loss function was below the set threshold.

In summary, we built and trained an XGBoost model that can perform regression prediction on a journal metrics time series dataset, which can be used for dynamic journal evaluation tasks.

4.1.4. LightGBM

LightGBM, proposed by Microsoft, is a lightweight gradient boosting algorithm that continues the integrated learning approach of XGBoost and optimizes XGBoost based on problems related to the existence of too many split points, too many samples, and too many features. This method can perform regression prediction for time series analysis tasks.

Compared with XGBoost, LightGBM uses the Histogram algorithm for feature selection, which constructs a histogram of width k by discretizing continuous floating-point features into k values, and then traverses the training data to calculate the cumulative statistics of each discretized value in the histogram. For the decision tree growth strategy, LightGBM adopts a leaf-wise strategy with a depth limit, finding the leaf with the largest splitting gain from all the current leaves each time to split, and then continues this process cyclically. Therefore, LightGBM has a smaller memory footprint and is faster but is also more prone to overfitting [20].

In our experiments, we used the same loss function for training as XGBoost in Section 4.1.3, and the model was considered trained when the value of the loss function was below a set threshold. We conducted time series data prediction experiments using this model on the journal metrics time series dataset and recorded the results.

4.2. Deep Learning Methods

As shown in Figure 4, we processed the journal metrics time series dataset into a two-dimensional vector of timesteps \times several eigenvalue dimensions with a single target to train the predicting model built by the deep learning methods.

4.2.1. Artificial Neural Network (ANN)

The Artificial Neural Network (ANN) is the most basic neural network, also called a multilayer perceptron. It is characterized by modeling and predicting through the process of learning training data without the need for complex mathematical models. Based on the fact that ANN can approximate arbitrary nonlinear mapping relations, it can be applied to the field of time series analysis for experiments [43].

As shown in Figure 5, an ANN network model containing 3 hidden layers was built for the dynamic journal evaluation task in this study. Both the ANN model and the machine learning method inputs were one-dimensional vectors obtained by the one-dimensional stitching of journal metric values in 3 timesteps, so the input layer had 54 neurons, the 3 hidden layers had 64, 128, and 64 nodes, and the output layer had only 1 node to output the predicted results of dynamic journal evaluation. Each layer is set with a corresponding partial coefficient. The network was optimized using the Adam algorithm, using the MSE metric as the loss function, and we set the initial learning rate of the model to 0.1, the decay rate of the first-order moment estimation to 0.9, the decay rate of the second-order moment estimation to 0.999, and the batch size to 32. In total, 90% of the data in the training set were taken for training, and 10% of the data were used to validate the network.

The ANN model is essentially a nonlinear mapping transformation of input samples to output samples, and the ANN model built in this study has the function of mapping past time series observations of journal metrics (each journal metric in the past three years) to future predicted values (possible future impact factors of journals). In this study, the ANN network model was constructed as a typical method of time series prediction in deep learning to perform one-step prediction and complete the task of the dynamic evaluation of journals.

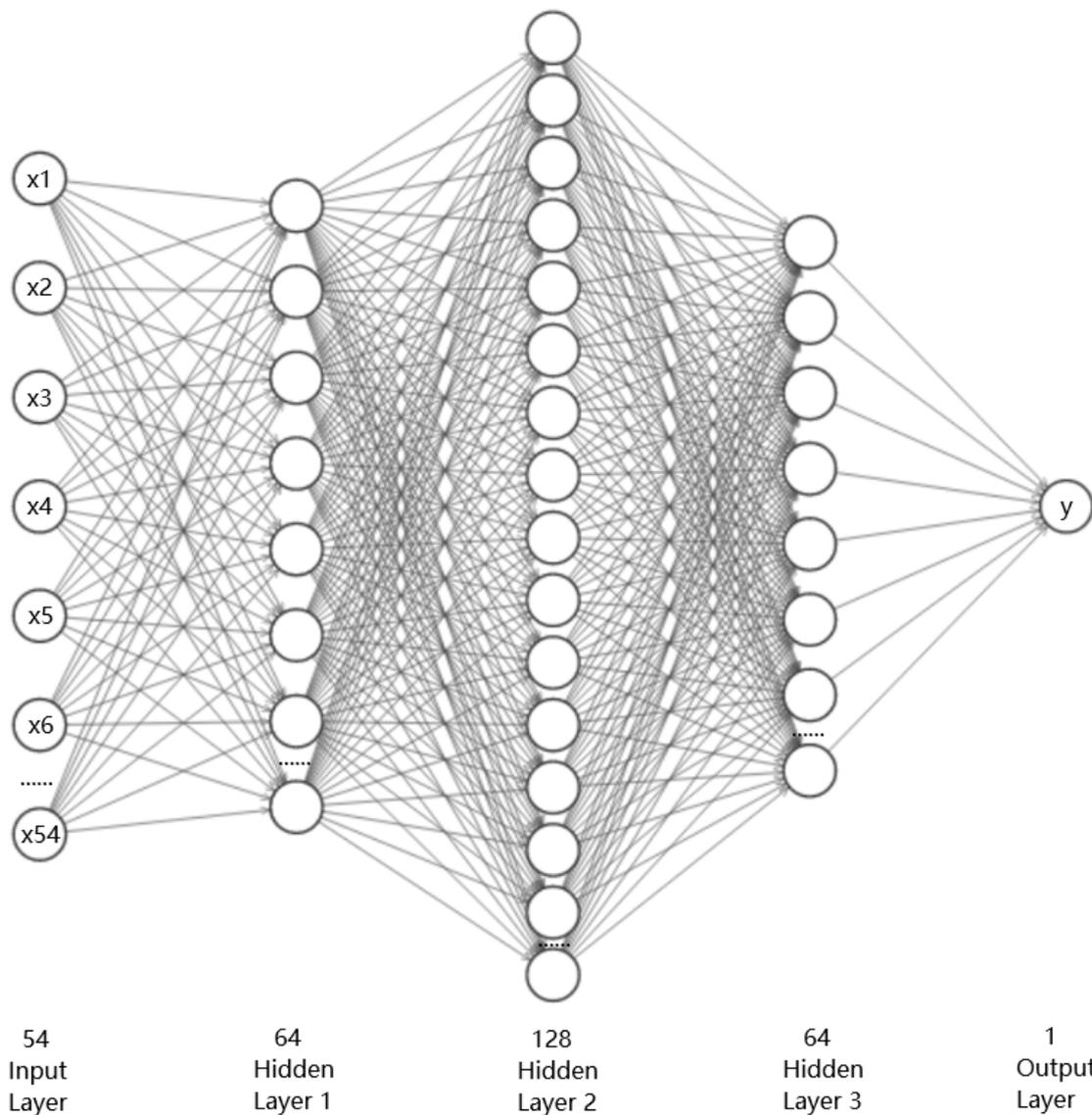


Figure 5. Structure of our ANN model for dynamic journal evaluation. The circles represent the neurons of each layer.

4.2.2. Conv-1D Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) can better identify simple features in data by convolution operations and can generate more complex features using these simple features by multi-layer convolution, pooling, and other operations, and finally, conduct regression prediction by using fully connected layers. Based on the characteristics of CNN, one-dimensional convolution (Conv-1D) is often applied to the analysis of time series, such as sensor data and signal data. For the journal metrics dynamic evaluation time series analysis task in this study, we implemented a Conv-1D CNN model for experiments on the journal metrics time series dataset [25].

As shown in Figure 6, a Conv-1D CNN model containing mainly 3 one-dimensional convolutional layers and 2 fully connected layers was built for this experiment. Each input data contained 18 journal evaluation metrics for each of the 3 years of a certain journal, so a 3×18 input matrix with 3 timesteps was obtained.

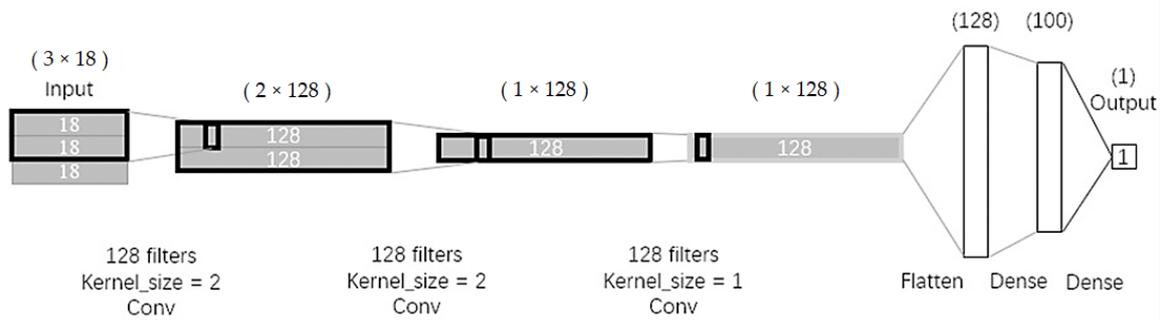


Figure 6. Structure of our Conv-1D CNN model for dynamic journal evaluation.

In this model, each convolutional layer was equipped with 128 convolutional kernels of size 2, 2, and 1, and we uniformly set the convolutional step size to 1 and used ReLU as the activation function. Since the convolution is intended to extract and learn features from the input of the layer, the simple features of the journal index time series can be gradually extracted and combined to form complex and comprehensive features after three layers of convolution. After obtaining the final extracted features from the convolution layer, they are fed into the fully connected layer for analysis, and the final output is a prediction of the possible impact factor values of the periodicals in the next time step.

We used the MSE metric as the loss function and trained the model using 90% of the data from the training set and 10% of the data for validation. We considered the training of the model complete when the loss value of a training epoch was below a set threshold or when the decrease in the loss value was too small after multiple training epochs, and we were able to complete the dynamic journal evaluation task using this model.

4.2.3. WaveNet

WaveNet is slightly different from ordinary one-dimensional convolution and is a model based on the atrous convolution style, which can also be called a temporal-based convolution, proposed by DeepMind in September 2016.

WaveNet's core architecture is atrous convolution, which ensures that the model is not exposed to information from future timesteps at each timestep. As shown in Figure 7, atrous convolution simply means skipping the input values by a certain step size and applying the convolution kernel to the region beyond its own size, thus achieving the goal of expanding the perceptual field and preserving the input resolution without increasing the computational cost significantly. Atrous convolution doubles the expansion factor of the convolution kernel in each layer until the cycle repeats from 1 after an expansion factor of 512 is reached [24].

In this study, we built a WaveNet model containing 9 convolutional layers, 1 flattened layer, and 1 fully connected layer structure, where the first 8 convolutional layers were atrous convolution modules with ReLU activation function, considering that each input data of the journal metrics time series contained only three timesteps containing 18 features, so each layer had 20 one-dimensional convolution kernels of size (2×18) and causal padding. We set the expansion factors to no more than 8. Each expansion factor was applied to 2 convolution layers; 10 one-dimensional convolution kernels of size (1×18) were set for the last convolution layer to extract the final features for input to the fully connected layer.

In the model training process, the training and validation sets were divided equally at 9:1, and the MSE metrics were used as the loss functions for convergence. The model had a large feeling field, and the network structure based on atrous convolution could automatically extract periodical information on different time scales, so it could extract time series features based on the journal metrics time series dataset to accomplish the task of dynamic journal evaluation.

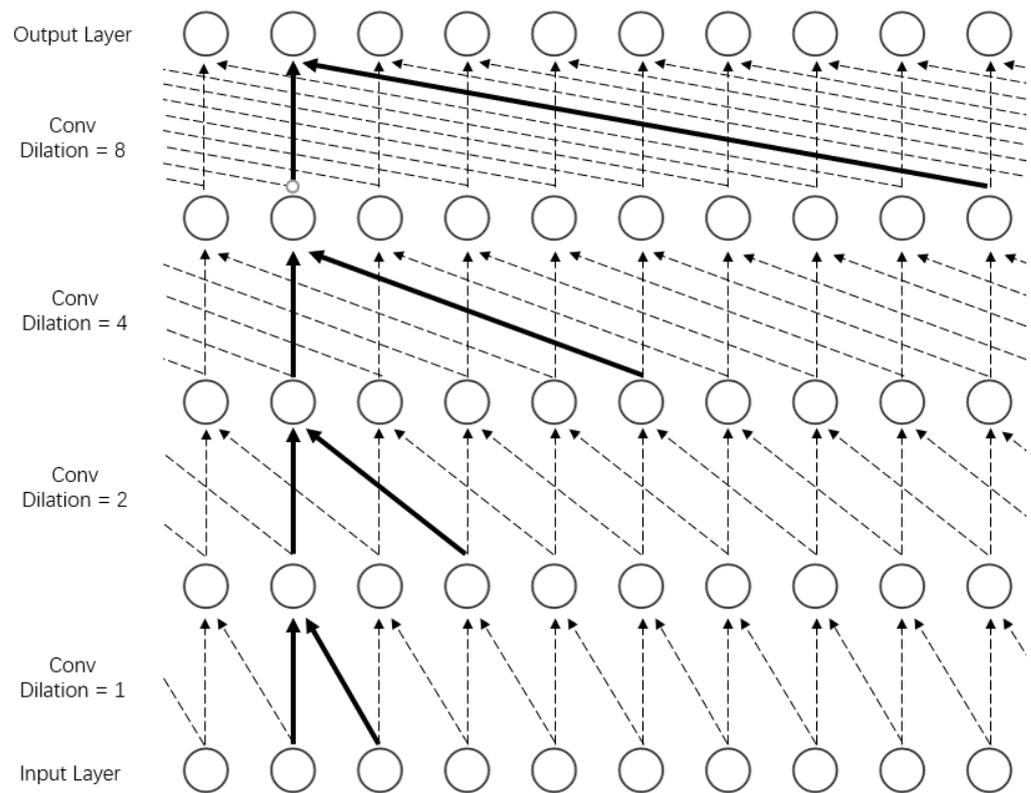


Figure 7. Example of atrous convolution. The circles represent the data of each layer, and the dotted lines represent the convolution operations between layers. The solid lines reflect atrous convolution, with different expansion factors between layers.

4.2.4. Long Short-Term Memory Network (LSTM)

The long short-term memory network (LSTM) is a recurrent neural network (RNN) specifically designed to solve the problem of gradient disappearance and gradient explosion caused by the existence of long-term dependence in general RNN, which introduces a gating unit with 3 gates (as shown in Figure 8), where h_{t-1} represents the output of the previous cell, C_{t-1} represents the previous cell state, x_t represents the input of the current cell, \tilde{C}_t represents the information to be updated into the cell state by the current cell, h_t represents the output of the current cell, C_t represents the current cell state, and σ represents the sigmoid layer (capable of mapping data in the 0–1 range and acting as a gating signal).

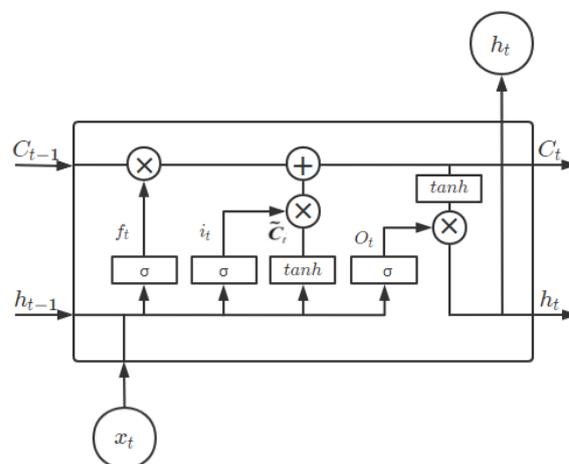


Figure 8. Example of LSTM cell, after inputting x_t and processing, obtaining the output h_t .

The LSTM model cell can be roughly divided into two lines, with the top line controlling long-time memory and the bottom line controlling short-time memory.

In the oblivion gate, f_t reads h_{t-1} and x_t and outputs a number from 0–1 with the previous cell state C_{t-1} in the previous cell, and the value determines which information is discarded from C_{t-1} .

In the input gate i_t , the results obtained by multiplying the results of the sigmoid layer with the vectors generated by the \tanh layer are used to determine which new information \tilde{C}_t is added to the cell state, and the final combination of the results of the forgetting gate and the input gate causes the cell state to be updated to C_t .

In the output gate o_t , the cell state is processed using the \tanh layer C_t and multiplied with the sigmoid layer result to obtain the final output of this cell, h_t ; then, h_t and C_t are passed to the next cell [44].

We can abstract the *LSTM* model as a function with three inputs and one output, as shown in Equation (10).

$$h_t = LSTM(x_t, C_{t-1}, h_{t-1}) \quad (10)$$

In this study, the dynamic journal evaluation task only required a one-step prediction of the possible future impact factors of journals. To simplify the network structure, we set only one layer of *LSTM* and one layer of fully connected layers when building the *LSTM* model. In the *LSTM* layer, the matrix dimension of the hidden layer was set to 50, the activation function was set to ReLU, the inputs were the spliced (3×18) journal metrics time series data, and only the output of the last time series was returned. The fully connected layer processed the 50-dimensional output of the *LSTM* layer into a single output value and this output was the one-step prediction of the input time series sample; we chose the MSE metric as the main loss function and the Adam algorithm as the optimizer. Finally, the training, validation, and testing of the *LSTM* model on the journal metrics time series dataset were completed, and the evaluation metrics results were derived.

LSTM controls the transmission state by gating the state and can remember information that needs to be remembered for a long period of time while forgetting unimportant information. Therefore, the *LSTM* model we built can be applied to the multivariate short-time multi-series data-based time series prediction task in this study.

4.2.5. Gate Recurrent Unit (GRU)

The gate recurrent unit (GRU) is a kind of RNN and a variant of *LSTM*, which reduces the number of gates in the gating unit to 2, eliminates the second-order nonlinear function of the output, combines the forgetting gate and the input gate into an “update gate”, fuses the cell state and the hidden state, and makes the structure and parameters more concise, so it is easier to compute the structure and parameters while maintaining a similar effect as *LSTM*. Figure 9 shows an example cell of GRU, where h_{t-1} represents the output of the previous cell, x_t represents the input of the current unit, \tilde{h}_t represents the information to be updated to the hidden state of the current cell, h_t represents the output of the current cell, and σ represents the sigmoid layer (capable of mapping data in the 0–1 range and acting as a gating signal).

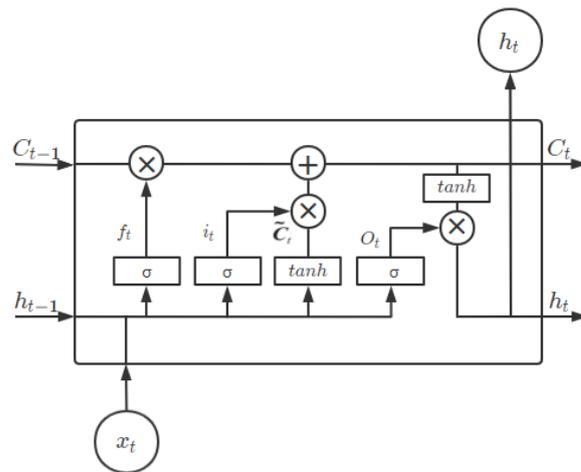


Figure 9. Example of GRU unit, after inputting x_t and processing, obtaining the output h_t .

In the GRU model cell, first, we combined the h_{t-1} and x_t sigmoid layers to obtain the reset gate r_t and update gate of z_t , where the reset gate combined signals r_t and h_{t-1} and multiplied them together to attain the “reset” data h'_{t-1} , which was then combined with the input x_t . Then, the “reset” data were obtained through the \tanh layer containing the current input data h' . The update gate performed both forgetting and remembering steps according to Equation (11) and used z_t for h_{t-1} selective forgetting and h' selective remembering; finally, the output of this unit is h_t [45].

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot h' \tag{11}$$

We can abstract the GRU model as a function with two inputs and one output, as shown in Equation (12), where the current unit receives the input x_t and combines the hidden state of the previous unit containing the information of the previous unit, h_{t-1} , to obtain the output of the current unit, h_t .

$$h_t = GRU(x_t, h_{t-1}) \tag{12}$$

In our experiments, we built a GRU model containing one layer of GRU with three fully connected layers. Among them, the hidden layer matrix dimension in the GRU layer was set to 64, the activation function between layers was set to LeakyReLU, the inputs were the spliced (3×18) journal metric time series data, and only the output of the last time series was returned. The neurons in the fully connected layer were 32, 16, and 1, and the 64-dimensional output of the GRU layer was processed into a single output value layer by layer in order to complete the one-step prediction of the 3 timesteps for the journal metrics time series data. We chose the MSE metric as the main loss function, and the ratio of the training set to the validation set was 9:1. Finally, we finished the model training, validation, and experiments on the test set and obtained the evaluation metric results, which proved that the GRU model can be applied to the dynamic journal evaluation task in this study.

5. Results

In this study, we conducted experiments using a single RTX 3090 GPU, used sklearn for model construction, training, and testing based on machine learning methods, and TensorFlow and Keras for model construction, training, and conducted testing based on deep learning methods. In the experiments, we built corresponding models using each of the nine methods introduced in Chapter 4 and then trained, tested, and compared the performance of these models on the journal metrics time series dataset and utilized various evaluation metrics to assess the experimental results in order to enable comprehensive analysis. We normalized the data before inputting them into the models and uniformly used

the MSE metric as the main loss function of the model. We uniformly set the deep learning model to use 80 epochs to try to improve the reasonableness of the metric comparison.

5.1. Training Situation Analysis

The model training situation of the nine methods is shown in Table 2. In the experiments, we used instrumentation timing to time the training of the models built by each method.

Table 2. Training situation analysis table. The best data in each column are displayed in bold.

	Methods	Training Time	Params	MSE
Machine Learning	Multiple Linear Regression	0.36618 s	-	0.00044
	Random Forest	134.23013 s	-	0.00047
	XGBoost	3.37131 s	-	0.00058
	LightGBM	2.57614 s	-	0.00076
Deep Learning	ANN	32.43376 s	20,161	0.00047
	Conv-1D CNN	51.61289 s	67,145	0.00047
	WaveNet	73.93935 s	6721	0.00050
	LSTM	108.72789 s	13,851	0.00037
	GRU	120.08700 s	18,561	0.00046

In machine learning, the multiple linear regression model benefits from its simple algorithm and obtains the fastest training speed and superior results. On the other hand, the random forest model is the slowest method, but its performance is fair. Presumably, its model pre-construction process is more complex, and we did not conduct any pruning, which resulted in the algorithm becoming too complex, resulting in increased time consumption. The XGBoost and LightGBM models are based on the optimization method of the boosting tree, which is faster than the random forest model, but the effect is relatively poor. In terms of the dynamic journal evaluation task, using the unpruned random forest model is more expensive, but the prediction fit of the journal metric time series dataset is better than that of the XGBoost and LightGBM optimization models.

The training speed of deep learning models is generally affected by the network structure, the number of epochs, and the number of parameters. Regarding the network structure, the general time consumption in the case of a close number of layers means that ANN is the fastest, CNN is the second fastest, and RNN is the slowest. In terms of the number of epochs, we uniformly conducted 80 epochs of iterative training during the experiment.

Among the five deep learning models built in this study, the ANN model has only four fully connected layers in its network structure, so it has the fastest training speed despite its large number of parameters. Alternatively, the Conv-1D CNN model has the largest number of trainable parameters (10 times that of WaveNet) among the five methods because it uses more convolutional kernels, but its network structure only contains three convolutional layers and one fully connected layer. The LSTM model and the GRU model are both RNNs, both of which take the longest amount of time to train among the deep learning methods. Although GRU is an optimized and simplified version of the LSTM model, we added two more fully connected layers to the GRU model structure compared with the LSTM model. The relative relationship between the five models' training situations is consistent with our inferences regarding the factors affecting the training speed.

On the whole, the training time for building models by machine learning methods is shorter than that of deep learning. Considering that machine learning generally requires artificial feature engineering of the input in advance, it is normal that the training time is shorter than that of deep learning models.

For the models built using the deep learning methods, the variations in the loss for 80 epochs during training are shown in Figure 10. It can be seen in the figure that the loss convergence curves of several models are relatively close, and the LSTM model has the best convergence, while the ANN, Conv-1D CNN, and GRU models have similar convergence effects, and WaveNet has the worst convergence, with large fluctuations at 20–30 epochs.



Figure 10. Comparison of deep learning methods (80 epochs; loss descent).

5.2. Evaluation Metrics Results

In Section 4.1, we introduce the mainstream evaluation metrics for time series prediction selected for this experiment, and the results of the evaluation metrics for the models built by the nine methods are shown in Table 3.

Among the models built by machine learning methods, the multiple linear regression model and the random forest model have the best metrics. The multiple linear regression model exhibits low training time consumption and high metric results with regard to the journal metrics time series dataset applied in this study and has the best overall performance among the machine learning models in this study, indicating that this dataset can be used to mine better linear dependencies.

The random forest model is able to learn the possible relationships between a large amount of time series data via the use of a forest composed of decision trees—without the use of a pruning strategy—and although the training period is long, time-consuming, and more complex in structure, it also obtains almost the best results in terms of the evaluation metrics in the journal time series dataset. However, its overall performance is not as good as the multiple linear regression model due to the long training period and excessive memory usage.

Table 3. Evaluation metric record table. The best data in each column are displayed in bold.

	Methods	MSE	MAE	RMSE	MAPE	NRMSE	80%	85%	90%
Machine Learning	Multiple Linear Regression	0.00044	0.01393	0.02109	22.143%	0.02516	68.433%	55.899%	40.645%
	Random Forest	0.00047	0.01342	0.02174	19.921%	0.02594	71.152%	59.447%	43.687%
	XGBoost	0.00058	0.01456	0.02418	21.963%	0.02884	68.203%	56.359%	39.862%
	LightGBM	0.00076	0.01561	0.02764	27.961%	0.03298	66.129%	53.963%	38.018%
Deep Learning	ANN	0.00047	0.01388	0.02165	23.472%	0.02583	68.525%	57.281%	42.074%
	Conv-1D CNN	0.00047	0.01369	0.02169	21.560%	0.02588	69.032%	58.479%	42.258%
	WaveNet	0.00050	0.01401	0.02225	22.983%	0.02655	67.604%	56.866%	41.567%
	LSTM	0.00037	0.01238	0.01914	19.704%	0.02283	72.442%	61.521%	46.083%
	GRU	0.00046	0.01304	0.02154	19.972%	0.02570	71.521%	60.046%	43.548%

The XGBoost and LightGBM models are based on the optimization method of boosting trees, and both models are slightly worse than the random forest model. LightGBM is the model with the worst evaluation metrics in terms of machine learning, and although it optimizes XGBoost from a theoretical point of view, it exhibits poor prediction generalization in this study, which is also consistent with its possible disadvantage of easy overfitting. Both XGBoost and LightGBM greatly improve the speed of training with a slight decrease in accuracy compared to the random forest model.

The performance of the test evaluation metrics of the four machine learning models after training on the journal metrics time series dataset reflects that the dataset constructed in this study achieved superior feature engineering and can be better applied to many time series analysis methods in the field of machine learning.

Among the models built by deep learning methods, the LSTM model is optimal in all metrics and is the best deep learning model when considering the training situation. However, the GRU model outperforms the other three deep learning models in terms of overall metrics, considering that both are special RNNs with gating units added, which can indicate that the ability of RNNs to mine implied relationships through short-time series under the journal metrics time series dataset is stronger, and the ability to link covariates with other time series for learning is more suitable for the dynamic journal evaluation task proposed in this paper.

The ANN and CNN models perform similarly on the evaluation metrics. ANN models are able to fit nonlinear relationships due to the presence of activation functions, and their logic and performance in the field of time series data analysis are similar to the multiple linear regression model used in machine learning. The large number of neurons contained in multiple hidden layers introduces a large number of parameters to be learned but also improves the models' ability to mine the implicit relationships in time series data.

The Conv-1D CNN model contains a large number of one-dimensional convolutional kernels and thus has the largest number of parameters, but it is able to extract multimodal hidden features among the time series of journal metrics compared to the ANN model, which contains only fully connected layers; therefore, evaluation metrics perform better with the Conv-1D CNN model than the ANN model overall.

The WaveNet model is an optimized version of the Conv-1D CNN model that uses atrous convolution to expand the perceptual field with a low number of parameters, but in the case of the journal metrics time series dataset, containing only three timesteps per input, the Conv-1D CNN model is able to obtain a more comprehensive perceptual field and features as well, although the network structure is shallower and has more parameters.

Therefore, the WaveNet model has a worse combined effect than the Conv-1D CNN model and is the worst-performing deep learning method.

Overall, for the dynamic journal evaluation task and the journal metrics time series dataset, the models built by machine learning perform better than the models built by multiple linear regression and random forest.

The two models built based on RNN perform the best among the models built using deep learning and can still obtain better evaluation metric results than the machine learning models, even under the premise that the feature engineering of the dataset in this study is better. On the other hand, ANN can maintain better evaluation metric results under the premise of the shortest training time consumption and is also capable of the task proposed in this paper. The overall comparison of the nine models in this study shows that the two deep learning models based on CNN are less effective, but the gap between them and other models is not particularly large.

In this study, a model was built based on nine methods in two fields, the model was trained on our journal metrics time series dataset for dynamic journal evaluation tasks, and experiments and evaluations were conducted. The results demonstrate, to a certain extent, that the current common and mainstream methods in the field of time series analysis can be generalized to the new tasks proposed in this paper. After analyzing the results of various metrics, the LSTM model is currently the best model for the comprehensive evaluation of the dynamic journal evaluation task, and the following subsection provides a more detailed analysis of the experimental results based on this method.

5.3. Dynamic Evaluation Analysis of Sci-tech Journals Based on LSTM

This subsection analyzes the performance of LSTM in the experiment in detail.

Figure 11 shows the comparison between the real results in the test set and the predicted results of the LSTM model. Blue dashes represent the actual values and green dashes represent the predicted values, revealing that the predicted results are well fitted. Considering that the difference between the metrics of other experiments and LSTM is not disparate, it can be demonstrated that the mainstream methods in the field of time series analysis are generalizable to the task of dynamic journal evaluation.

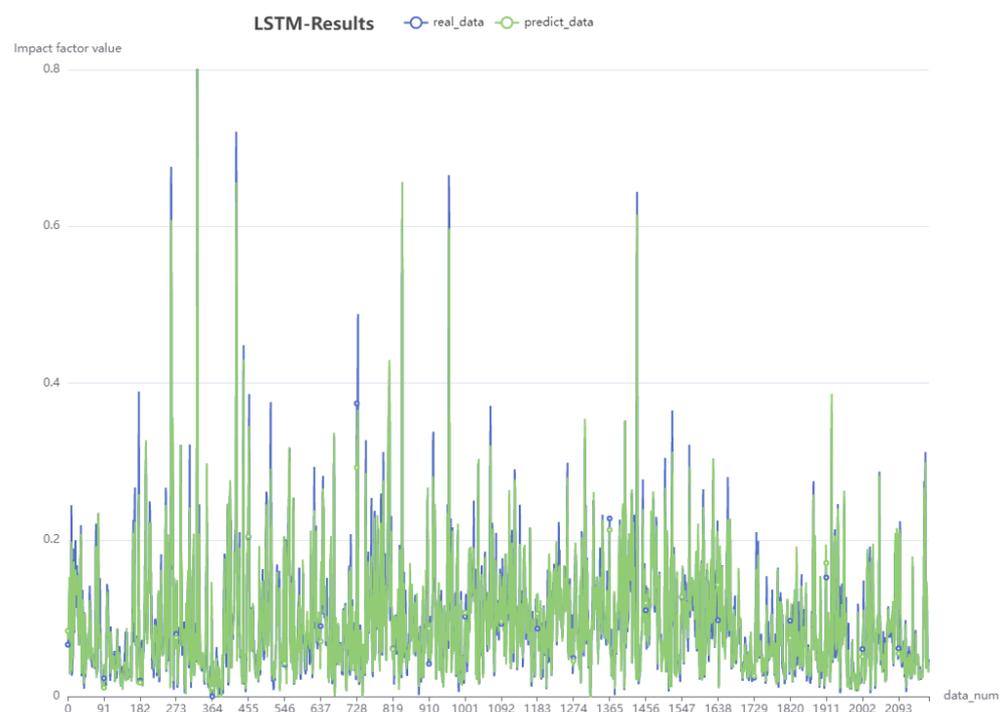


Figure 11. Prediction results of the LSTM model for the test set compared with the actual results. Blue dashes represent the actual values and green dashes represent the predicted values.

Table 4 shows the results of the evaluation metrics of the trained LSTM model in the test set for dynamic evaluation of journals in different classes of subjects. It can be seen that the dynamic evaluation of journals in the field of medicine is optimal, followed by agriculture; however, the overall dynamic evaluation of the subjects of economics and transportation is poor.

Table 4. LSTM results' evaluation metrics of different classes of objects. The best data in each column are displayed in bold.

Subject Class	MSE	MAE	RMSE	MAPE	NRMSE	80%	85%	90%
Medicine	0.00040	0.01372	0.01990	14.032%	0.03096	82.656%	72.847%	55.622%
Engineering	0.00034	0.01020	0.01844	20.840%	0.04779	69%	57%	44%
Pedagogy	0.00045	0.01311	0.02123	27.958%	0.03149	71.622%	59.459%	44.595%
Technical Science	0.00018	0.00902	0.01330	19.659%	0.03335	69.412%	58.235%	44.706%
Economic	0.00086	0.01806	0.02924	35.588%	0.04092	53.947%	43.421%	31.579%
Transportation	0.00015	0.00969	0.01237	17.177%	0.07558	61.538%	50%	30.769%
Social Science	0.00030	0.01213	0.01743	17.112%	0.02764	68.868%	53.774%	36.792%
Natural Science	0.00036	0.01278	0.01887	22.791%	0.06288	61.667%	52.5%	38.333%
Materials Science	0.00026	0.01225	0.01599	43.718%	0.12383	55.263%	47.368%	34.211%
Agronomy	0.00014	0.00869	0.01180	14.211%	0.05090	79.310%	63.793%	48.276%
Basic Science	0.00022	0.00993	0.01477	27.944%	0.05378	59.483%	48.276%	34.914%
Others	0.00077	0.01635	0.02775	27.026%	0.03310	62.745%	53.922%	37.255%

Through detailed analysis of the journal metric time series dataset, we found that journals in the medical field are the most numerous and most completely and comprehensively recorded samples in the dataset, while journals in the field of agronomy are smaller in number, but their metric records are more comprehensive, with fewer missing values and a high degree of metric diversity; thus, both have better evaluation metric results under the trained LSTM model. In contrast, journals in the field of economics and transportation are smaller in number in the dataset, and it is verified that the original data of these two types of journals contain more outliers and missing values, and although they are processed through data cleaning, poorer evaluation metric results are still obtained under the LSTM model.

The above analysis proves that the authenticity, completeness, and comprehensiveness of metric records are very important for the dynamic journal evaluation task, and the number of training samples is essentially positively correlated with the performance effect. Although the problem of abnormal data can be solved by data pre-processing, it still affects the final training effect.

6. Discussion

Sci-tech journals currently play an important role in the innovation systems of each country as a carrier of academic content, and effective journal evaluation and prediction models will play a crucial role in the development of sci-tech journals, so the purpose of this study was to dynamically evaluate the possible future development of the journal through journal historical metrics, which will enable us to allocate the right resources and develop appropriate strategies for journal development. In this chapter, we will discuss the methods and results of this paper and analyze the theoretical and practical implications based on the above content.

6.1. Methods and Results Discussion

In this paper, we introduced time series data prediction methods using artificial intelligence into the dynamic journal evaluation scenario and built and tested nine models based on various metrics.

In the experimental process of this study, we found that although machine learning methods are not necessary to dimension the process of the data, normalizing the data and then training the model can still improve the evaluation metrics. WaveNet based on atrous convolution performs no better than Conv-1D CNN on the dynamic journal evaluation task because the dataset is a multivariate short-time multi-series and has few time-steps. Although multiple linear regression is a simple algorithm, it performs well on the dataset in this study, and its model can be quickly implemented to obtain the basic values of evaluation metrics when performing the time series analysis task. The models built with RNN methods (LSTM and GRU) obtained the best performance on the dynamic journal evaluation task.

In recent years, there have been many papers conducted on the analysis of time series data, for example, Alzain E, et al. used artificial intelligence to predict housing prices [35], Shehadeh A, et al. used machine learning models to predict the residual value of heavy construction equipment [34], Ecer F, et al. trained a multilayer perceptron to predict the stock price index [36], Ahmed A A, et al. integrated LSTM and random forest to estimate soil moisture [39], etc.

Similar to these papers, we also propose new scenarios, new tasks, and new datasets, and build models. In addition, our time series dataset of journal metrics is more challenging than traditional time series datasets, because it is a multivariate short-time multi-series dataset; our experiments cover the two major fields of machine learning and deep learning; therefore, we are able to provide a relatively reliable result after a comprehensive comparison of various mainstream time series analysis models.

Through this study, we found that, currently, on journal platforms such as WanFang and ZhiWang, there are more journals in the medical field and comprehensive records of metrics, while economics journals are relatively fewer in number and have more metric anomalies. To achieve the orderly and healthy development of sci-tech journals, the dissemination media of sci-tech journals should be expanded, diverse dissemination channels should be opened, and annual metrics should be comprehensively recorded and calibrated.

6.2. Theoretical and Practical Implications

Practical Implication. The model built, trained, and experimented with in this study can be extended with a variety of application strategies in future journal evaluation and analysis tasks and can serve as a guide for resource allocation and the subsequent development planning of journals. For example, when trying to make decisions on arranging the allocation of journal resources, the model built in this study can be used to outline the possible future development status of journals, which can help guide the reasonable allocation of journal resources to a certain extent. For instance, when there are certain planning expectations for the development of journals in a given year, they can be passed into the models as input to obtain dynamic evaluation results, and different dynamic evaluation results can be obtained through the continuous adjustment of the planning, which can, in turn, assist in adjusting journals' development.

Theoretical Implication. The comprehensive conclusions of the experiments conducted in this study can provide methodological references for other multivariate short-time multi-series time series analysis studies, and the dataset proposed in this paper can also provide data support for similar tasks.

7. Conclusions

At present, sci-tech journals are becoming increasingly important as part of the national innovation system. The quality of journals is directly related to the development and progress of subjects, and the scientific and systematic evaluation of journals can help

improve their quality and influence, as well as facilitate researchers to understand the current research priorities and hot spots. The development of a dynamic evaluation model for journals based on journal metrics and time series data and the verification of its rationality and usability can play an important role in the development of sci-tech journals.

Contributions. In this study, we proposed a new task: Dynamic evaluation task for journals and construct a new time series dataset of 18 journal evaluation metrics. Based on this, we built nine models based on the mainstream machine learning and deep learning methods in the field of time series analysis and conducted training and experiments on the dataset to provide comprehensive and diverse evaluation metric results, then compared and analyzed the results to confirm the generalizability of these methods for the comprehensive dynamic evaluation of journals, and found the LSTM model we built achieved high prediction accuracy in the task proposed in this paper, laying the foundation for subsequent research on this task.

Limitations. We built the dataset by collecting journal evaluation metrics data in the past five years from the Wanfang and ZhiWang platforms, but there are still many journal platforms and search websites that can provide more data or proofread existing data. We built models using nine mainstream time series forecasting methods to dynamically evaluate sci-tech journals, and although these models are universal for the task of dynamic evaluation of sci-tech journals, they have not been specifically optimized for our multivariate short-time multi-series dataset. These limitations also provide directions for future studies.

Future studies. To improve the accuracy and credibility of the models built in this study for dynamic journal evaluation, more journal metric time series data could be added from different data sources. From the perspective of the feature value, we could consider selectively choosing some covariates for model training based on the importance of the journal metrics. With regard to the algorithm, we could consider improving the network structure and the parameter settings of the model built based on LSTM and GRU methods in this study, for example, adding a transformer or Seq2Seq structure, etc.

Author Contributions: Conceptualization, Y.M., Y.H., and M.C.; data curation, Y.M., Y.H., and M.C.; formal analysis, Y.M. and M.C.; funding acquisition, Y.M., Y.H., and Y.C.; investigation, M.C.; methodology, Y.M. and M.C.; project administration, Y.M., Y.H., and Y.C.; resources, Y.M., Y.H., and Y.C.; software, M.C.; supervision, Y.M., Y.H., and Y.C.; validation, M.C.; visualization, M.C.; writing—original draft, M.C.; writing—review and editing, M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by a scientific project of the State Grid, Shandong Electric Power Research Institute, grant number ZY-2022-07; and a scientific project of the State Grid, grant number 520626220042.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: If any researcher requires data support for this article, please contact the author of this article via email to obtain our dataset.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Thomas, D.R. A general inductive approach for analyzing qualitative evaluation data. *Am. J. Eval.* **2006**, *27*, 237–246. [[CrossRef](#)]
2. Derek de solla Price. *Little Science, Big Science*; Columbia Press: New York, NY, USA, 1965; p. 53.
3. Birkle, C.; Pendlebury, D.A.; Schnell, J.; Adams, J. Web of Science as a data source for research on scientific and scholarly activity. *Quant. Sci. Stud.* **2020**, *1*, 363–376. [[CrossRef](#)]
4. Garfield, E. The history and meaning of the journal impact factor. *Jama* **2006**, *295*, 90–93. [[CrossRef](#)] [[PubMed](#)]
5. Alonso, S.; Cabrerizo, F.J.; Herrera-Viedma, E.; Herrera, F. h-Index: A review focused in its variants, computation and standardization for different scientific fields. *J. Informetr.* **2009**, *3*, 273–289. [[CrossRef](#)]
6. Abdi, H.; Williams, L.J. Principal component analysis. *Wiley Interdiscip. Rev. Comput. Stat.* **2010**, *2*, 433–459. [[CrossRef](#)]

7. Zhu, Y.; Tian, D.; Yan, F. Effectiveness of entropy weight method in decision-making. *Math. Probl. Eng.* **2020**, *2020*, 3564835. [[CrossRef](#)]
8. He, L.; Xingye, D. Research on intelligent evaluation for the content innovation of academic papers. *Libr. Inf. Serv.* **2020**, *64*, 93.
9. Goffman, W.; Morris, T.G. Bradford's law and library acquisitions. *Nature* **1970**, *226*, 922–923. [[CrossRef](#)]
10. Clermont, M.; Krolak, J.; Tunger, D. Does the citation period have any effect on the informative value of selected citation indicators in research evaluations? *Scientometrics* **2021**, *126*, 1019–1047. [[CrossRef](#)]
11. Chi, P.S. Differing disciplinary citation concentration patterns of book and journal literature? *J. Informetr.* **2016**, *10*, 814–829. [[CrossRef](#)]
12. Guz, A.N.; Rushchitsky, J.J. Scopus: A system for the evaluation of scientific journals. *Int. Appl. Mech.* **2009**, *45*, 351–362. [[CrossRef](#)]
13. Zhang, Y.; Yu, L. A comparative study on the relationship between article volume and impact factor of humanities and social sci-tech journals—The example of art and design and intelligence journals. *J. Intell.* **2019**, *38*, 151–157.
14. Tang, K.Y.; Lu, J.; Li, H. Research on the influence of journalism and communication journals' article volume on academic influence. *Media Watch* **2021**, *38*, 91–97.
15. Li, C.; Ding, Z. Research on the relationship between the number of articles and the influence of excellence action plan journals based on the M-K trend test. *Technol. Publ.* **2021**, *40*, 78–84.
16. Wu, T.; Shi, J.; Yang, Y.; Chen, C.; Sun, J. A comparative study of core indicators for citation evaluation of science and technology journals. *China J. Sci. Technol. Res.* **2014**, *25*, 1058–1062.
17. Yu, L. A study on the selection of nonlinear academic evaluation methods based on neural networks. *Intell. Theory Pract.* **2021**, *44*, 63–70, 56.
18. Esling, P.; Agon, C. Time-series data mining. *ACM Comput. Surv. (CSUR)* **2012**, *45*, 1–34. [[CrossRef](#)]
19. Newbold, P. ARIMA model building and the time series analysis approach to forecasting. *J. Forecast.* **1983**, *2*, 23–35. [[CrossRef](#)]
20. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3146–3154.
21. Chen, T.; He, T.; Benesty, M.; Khotilovich, V.; Tang, Y.; Cho, H.; Chen, K. Xgboost: Extreme gradient boosting. *R Package Version 0.4-2* **2015**, *1*, 1–4.
22. Althelaya, K.A.; El-Alfy ES, M.; Mohammed, S. Stock market forecast using multivariate analysis with bidirectional and stacked (LSTM, GRU). In Proceedings of the 2018 21st Saudi Computer Society National Computer Conference (NCC), Riyadh, Saudi Arabia, 25–26 April 2018; pp. 1–7.
23. Medsker, L.R.; Jain, L.C. Recurrent neural networks. *Design and Applications* **2001**, *5*, 64–67.
24. Oord, A.V.D.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv* **2016**, arXiv:1609.03499.
25. Harper, C.A.; Lyons, L.; Thornton, M.A.; Larson, E.C. Enhanced Automatic Modulation Classification using Deep Convolutional Latent Space Pooling. In Proceedings of the 2020 54th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 1–4 November 2020; pp. 162–165.
26. Bisong, E. Introduction to Scikit-learn. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*; Apress: Berkeley, CA, USA, 2019; pp. 215–229.
27. Das, K.; Jiang, J.; Rao, J.N.K. Mean squared error of empirical predictor. *Ann. Stat.* **2004**, *32*, 818–840. [[CrossRef](#)]
28. Willmott, C.J.; Matsuura, K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Clim. Res.* **2005**, *30*, 79–82. [[CrossRef](#)]
29. Karunasingha, D.S.K. Root mean square error or mean absolute error? Use their ratio as well. *Inf. Sci.* **2022**, *585*, 609–629. [[CrossRef](#)]
30. Chicco, D.; Warrens, M.J.; Jurman, G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Comput. Sci.* **2021**, *7*, e623. [[CrossRef](#)]
31. Shcherbakov, M.V.; Brebels, A.; Shcherbakova, N.L.; Tyukov, A.P.; Janovsky, T.A.; Kamaev, V.A.E. A survey of forecast error measures. *World Appl. Sci. J.* **2013**, *24*, 171–176.
32. Ng, K.Y.; Awang, N. Multiple linear regression and regression with time series error models in forecasting PM10 concentrations in Peninsular Malaysia. *Environ. Monit. Assess.* **2018**, *190*, 1–11. [[CrossRef](#)]
33. Kane, M.J.; Price, N.; Scotch, M.; Rabinowitz, P. Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks. *BMC Bioinform.* **2014**, *15*, 1–9. [[CrossRef](#)]
34. Shehadeh, A.; Alshboul, O.; Al Mamlook, R.E.; Hamedat, O. Machine learning models for predicting the residual value of heavy construction equipment: An evaluation of modified decision tree, LightGBM, and XGBoost regression. *Autom. Construct.* **2021**, *129*, 103827.
35. Alzain, E.; Alshebami, A.S.; Aldhyani TH, H.; Alsubari, S.N. Application of Artificial Intelligence for Predicting Real Estate Prices: The Case of Saudi Arabia. *Electronics* **2022**, *11*, 3448. [[CrossRef](#)]
36. Ecer, F.; Ardabili, S.; Band, S.S.; Mosavi, A. Training multilayer perceptron with genetic algorithms and particle swarm optimization for modeling stock price index prediction. *Entropy* **2020**, *22*, 1239. [[CrossRef](#)] [[PubMed](#)]
37. Cai, M.; Pipattanasomporn, M.; Rahman, S. Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques. *Appl. Energy* **2019**, *236*, 1078–1088. [[CrossRef](#)]

38. Borovykh, A.; Bohte, S.; Oosterlee, C.W. Conditional time series forecasting with convolutional neural networks. *arXiv* **2017**, arXiv:1703.04691.
39. Ahmed, A.A.; Deo, R.C.; Ghahramani, A.; Raj, N.; Feng, Q.; Yin, Z.; Yang, L. LSTM integrated with Boruta-random forest optimiser for soil moisture estimation under RCP4. 5 and RCP8. 5 global warming scenarios. *Stoch. Environ. Res. Risk Assess.* **2021**, *35*, 1851–1881. [[CrossRef](#)]
40. Yamak, P.T.; Yujian, L.; Gadosey, P.K. A comparison between arima, lstm, and gru for time series forecasting. In Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence, Sanya, China, 20–22 December 2019; pp. 49–55.
41. Uyanık, G.K.; Güler, N. A study on multiple linear regression analysis. *Procedia-Soc. Behav. Sci.* **2013**, *106*, 234–240. [[CrossRef](#)]
42. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
43. Wang, S.C. Artificial neural network. In *Interdisciplinary Computing in Java Programming*; Springer: Boston, MA, USA, 2003; pp. 81–100.
44. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
45. Dey, R.; Salem, F.M. Gate-variants of gated recurrent unit (GRU) neural networks. In Proceedings of the 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS), Boston, MA, USA, 6–9 August 2017; pp. 1597–1600.