EasyChair Preprint

№ 6716

A Comprehensive Evaluation of Statistical, Machine Learning and Deep Learning Models for Time Series Prediction

Ang Xuan, Mengmeng Yin, Yupei Li, Xiyu Chen and Zhenliang Ma

September 28, 2021

A comprehensive evaluation of statistical, machine learning and deep learning models for time series prediction

Ang Xuan*
Tsinghua-Berkeley Shenzhen Institute
Tsinghua University
Shenzhen, China
xa19@mails.tsinghua.edu.cn

Mengmeng Yin*
School of Aerospace Engineering
Beijing Institute of Technology
Beijing, China
yin_mengmeng@aliyun.com

Yupei Li*
School of Computer Science and Technology
Xi'an Jiaotong University
Xian, China
lady6838@ox.ac.uk

Xiyu Chen*
Qianweichang College
Shanghai University
Shanghai, China
chenxiyu@shu.edu.cn

Zhenliang Ma*
Department of Civil Engineering
Monash University
Melbourne, Australia
Mike.Ma@monash.edu

***Abstract*—How to choose the appropriate model to predict the time series is one of the most prominent activities of temporal data analysis. Empirical evidence is often adopted to select the most suitable model since there is no unified standard for matching data and models. Data characteristics affect model performance to a certain extent and maybe where the factors that determine the balance between prediction accuracy and model complexity are. In this article, Multi-Criteria Performance Measure method considering Mean of Absolute Value of the Residual Autocorrelation was adopted to address this problem. Case studies summarize the limitations and recommendations from the period, trend, stationarity and seasonality of datasets. The results show that the statistical models perform best for datasets with low stochasticity and high trend and seasonality, deep learning models specialize in forecasting fluctuant and long-term time series data, machine learning models could be candidates for datasets that possess numerical characters between the previous two categories. Conclusions could provide suggestions in selecting appropriate models and guide the research community in focusing the effort on more feasible or promising directions.**

Keywords—time series data prediction, statistical model, machine learning model, deep learning model, Multi-Criteria Performance Measure

I. INTRODUCTION

TIME series prediction problem nowadays has produced profound influences in great quantities of field, such as stock prices forecasting, weather forecasting, business planning, resources allocation and many others. Meanwhile, there are varieties of models to solve these problems, involving traditional statistical models, machine learning models and deep learning models. However, for certain prediction problems, what specific model is recommended to be used is still an outstanding question and quite worthy of working out for the extensive application.

Several scholars have carried out related researches in developing time series prediction methods. Early attempts to study time series prediction [1], particularly in the nineteenth century, were generally characterized by the view of a deterministic world. Scholars such as *Slutsky*, *Walker*, *Yaglom*, and *Yule* first proposed the concept of autoregressive (AR) and moving average (MA) models to formulate the ARMA model [2]. Along with the increase of computing power, statistical models such as regression and autoregressive integrated moving average (ARIMA) models according to Wold's decomposition theorem [3] popularized the use of their extensions in many areas of science for the years.

For the last two to three decades, with the advent of the data mining technique, there is a gradual concern for the applications of machine learning models for time series prediction. Literature [4] present a large-scale comparison study for the major machine learning models for time series forecasting applied in around a thousand-time series. Compared to classical statistical models, the machine learning techniques have established themselves as strong competitors due to their simplicity and comprehensibility. Two practices are introduced to alleviate the negative effect of the Artificial Neural Network (ANN) model in [5]. Research [6] provide insight into the applications using Support Vector Machines (SVM) for time series prediction and outline the advantages and challenges in using SVMs for time series prediction. The Random Forest approach is employed to explore the utility of the time series dataset compared to the ARIMA model [7].

Scientific progress has been undertaken to encourage the improvement of these algorithms as the development of new solutions. Meanwhile, with the sharp increase in the quantity and dimensionality of data, new challenges such as extracting deep features and recognizing deep latent patterns have emerged.

In recent years, deep learning techniques have developed at the forefront of artificial intelligence. Neural networks utilize multiple layers to represent latent features at a higher and more abstract level to describe models [8]. The interior relationships are learned from data itself rather than constructed by human engineers. Regarding the above model feature, deep learning-based models have been successfully applied in many areas to forecast time series data, including convolutional neural network (CNN) [9], Recurrent Neural Network (RNN) [10], Gated Recurrent Unit neural network (GRU) [11], Long Short-term Memory Neural Network (LSTM) [12].

Apart from the above-mentioned individual models, there are also some hybrids structures applied on time series prediction [13]-[18]. For instance, a hybrid model is proposed with the ARIMA methodology and neural network architecture to predict water quality in [15], the CNN-

* These authors contributed equally to this work and should be considered co-first authors.

BiLSTM-AM model demonstrated in [18] is adopted for the prediction of stock price and for providing investors to make investment decisions.

However, the prediction problem generally uses empirical evidence to select the most suitable model since no modelling method can be considered the best. According to the literature review of the last decade, few scientific publications rigorously expose the benefits, limitations and recommendations of the most popular and classical algorithms for time series prediction.

To solve the mentioned problems, we discuss three types of models mentioned above, and for each model, we pick three representatives to compare their performances. Furthermore, we chose four different datasets, whose characteristics are distinct, as our training dataset. Distinct from previous work, we precisely focus on the model application choice for a certain situation and dataset.

The two major contributions of this paper can be summarized as follows:

1). Experimentally explore the relationship between model performance and characteristics of the data.

2) Various real-time series datasets were selected, covering different time scales, seasonality, measurement granularity, etc. Meanwhile, nine current classical methods are covered;

3). Three evaluate indices Root Mean Square Error, correlation coefficient, and Prediction of Change in Direction were adopted to judge in more comprehensive aspects, and then Multi-Criteria Performance Measure Method was brought in to score the overall performance for each model.

The paper is organized as follows. Section II presents the mathematical fundamentals of prediction models. Section III describes the datasets and their statistical characteristics. Section IV illustrates the methodology of time series data prediction and evaluation systems. Section V presents the results, performance, limitations, recommendations. Finally, Section VI summarizes the conclusions and discuss directions for future work.

II. Mathematical Fundamentals

The time series prediction models have evolved over the years, passing from regression techniques to statistical and then to artificial intelligence algorithms. The following three subsections discuss nine approaches along with the most renowned algorithms for time series prediction.

A. Statistical models

1) Autoregressive Integrated Moving Average

The ARIMA models of parameters (p, d, q), i.e. ARIMA (p, d, q), result from the combination of three procedures: *1)* Autoregression (AR(p)), *2)* integration2, and *3)* Moving Averages (MA(q)).

When a time series is non-stationary, it can be transformed using a data differentiation procedure that ensures such property. This procedure added to the ARMA structure results in the ARIMA model with order (p, d, q), ARIMA (p, d, q), defined by equation (1) below.

$$I'_t = \delta + \sum_{i=1}^{p} \phi_i I'_{t-i} + \sum_{i=1}^{q} \theta_i e_{t-i} + e_t \qquad (1)$$

In equation (1), $I'_t = \Delta^d z_t = \Delta(\Delta^{d-1} z_t)$ (z_t represents the prediction value z for the period t), and d indicates the difference operator degree; ϕ_p and θ_q are, in this order, the parameters of the procedures; autoregressive with lag length p, and MA, with lag length q; δ reflects the initial level of the model (performs the same function as the intercept in linear regression).

The value range of parameters the models ARIMA (p, d, q) was defined by the value of ACF (Autocorrelation Function), PACF (Partial Autocorrelation Function), and the confidence interval. The exact parameters were defined using the minimization of the BIC (Bayesian information criterion); and e_t is the white noise in a distribution with zero average and constant variance σ_e^2 .

2) SARIMA

ARIMA exploits the autocorrelation between the time series values at successive instants, but when the data are observed in periods of less than one year, the series may also have autocorrelation for a seasonal station s. The seasonal ARIMA models, also known as SARIMA, have in their structure a non-seasonal part, with parameters (p, d, q), and a seasonal part, with parameters (P, D, Q, S) in equation (2).

$$I''_t = \delta + \sum_{i=1}^{P} \Phi_{is} I''_{t-is} + \sum_{i=1}^{Q} \Theta_{is} e_{t-is} + e_t \qquad (2)$$

In equation (2), $I''_t = \Delta^D z_t = \Delta(\Delta^{D-1} z_t)$ and D indicates the degree of the seasonal difference operator; the constant δ follows the same rules as those imposed on the ARIMA structure, but now considering D; Φ_P and Θ_Q are, in this order, the parameters of the procedures seasonal autoregressive, with lag length P, and of the seasonal MA, with lag length Q; and e_t is the white noise that cannot be explained by the model.

The SARIMA (p, d, q)×(P, D, Q, S) is denoted by equation (3) below, where the non-seasonal and seasonal parts are summed.

$$I_t = \delta + \sum_{i=1}^{p} \phi_i I'_{t-i} + \sum_{i=1}^{q} \theta_i e_{t-i} + \sum_{i=1}^{P} \Phi_{is} I''_{t-is} + \sum_{i=1}^{Q} \Theta_{is} e_{t-is} + e_t \qquad (3)$$

The value range of parameters the models SARIMA (p, d, q)×(P, D, Q, S) was defined by the amount of data in the dataset, from 0 to $\sqrt{\log(m-h)}$, where m is the size of the dataset, and h is 5% of the series size (usually $\sqrt{\log(m-h)} = 2$). The exact parameters were defined using the minimization of the AIC (Akaike information criterion).

3) LASSO

LASSO is Least Absolute Shrinkage and Selection Operator. The model is a compressed estimation. A more refined model is obtained by constructing a penalty function, which compresses some regression coefficients. That is, the sum of absolute values of the coercive coefficients is less than a fixed value. For the sake of introduction, we assume that the given n data sample points $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. This can also be expressed in matrix form $X = [x_1; x_2; \ldots; x_n]^T$ and $y = (y_1, y_2, \ldots, y_n)^T$. Generally speaking, regression problem is a function fitting process, and we need to add a regularization term whose coefficient is α, in order to avoid

the over-fitting phenomenon. The optimization objective of Lasso is shown in (4):

$$\beta^* = argmin_\beta \frac{1}{n} \|y - X\beta\|_2^2 + \alpha\|\beta\|_1 \quad (4)$$

B. Machine learning models

For the category of machine learning part, we chose three classical models to compare their behaviours, which are Artificial neural network (ANN), Decision Tree (DT) and Gradient Boosting Decision Tree (GBDT).

1) ANN

ANN with multi-layer and single-layer, each layer contains a number of neurons, each with variable weights between neurons has to arc connection, by the repeated training to the known information network, the model to adjust the connection weights change neurons step by step, to deal with information, the purpose of the simulation of the relation between input and output. And in our research, we adjust the number of the layer and parameters and also activation to make the network behaved best

2) DT

DT is a Decision analysis model to obtain the probability that the expected value of net present value is greater than or equal to zero, evaluate project risk and judge its feasibility by forming a Decision Tree on the basis of the known probability of occurrence of various situations. It is a graphical model of intuitive use of probability analysis.

3) GBDT

GBDT are all regression trees. GBDT is used for regression prediction and can also be used for classification after adjustment (set a threshold value greater than the threshold value is a positive example, and vice versa). A variety of distinguishing features and feature combinations can be found. GBDT aims to sum up the conclusions of all trees to make the final conclusion. The core of GBDT is that each tree learns the residual (negative gradient) of the sum of all previous tree conclusions. This residual is a sum that can get the true value after adding the predicted value.

C. Deep learning models

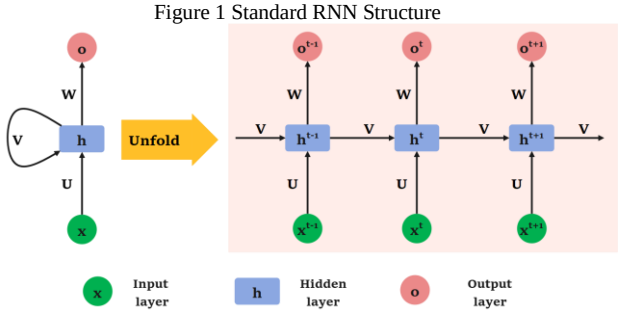
1) Convolutional Neural Network

As a class of artificial neural networks that have become dominant in various computer vision tasks, the Convolutional Neural Network (CNN) gradually attracts interest across various domains. CNN is designed to automatically and adaptively learn spatial hierarchies of features through backpropagation by adopting multiple interior blocks such as convolution layers, pooling layers, and fully connected layers.

2) Recurrent Neural Network

The multi-layer neural network models could be categorized as feedforward neural networks because the neuron-to-neuron signals flow only in one direction: from input to output. Differently in Recurrent Neural Networks (RNN), connections between neurons form a cycle, and the signals can move in different directions. For example, in a simple RNN, the state of the hidden layer at a given time is conditioned on its previous state by a context layer, as illustrated in Figure 1. This recursion implies a short-term memory, allowing the network to store complex signals arbitrarily. The ability to model temporal dependencies makes RNN especially suitable for the task as prediction, where input and output cover dependent data sequences.

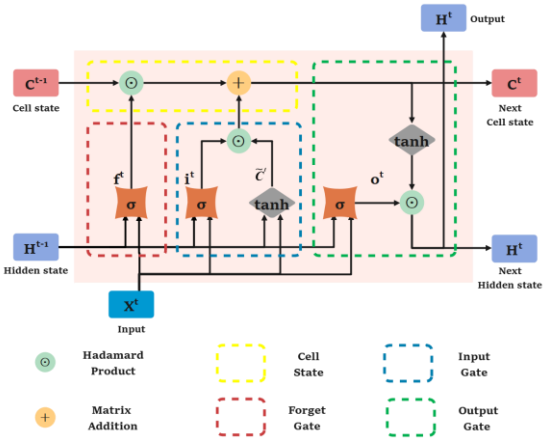
Figure 1 Standard RNN Structure



3) LSTM

In recent years, Long Short-term Memory (LSTM) was developed to deal with the vanishing gradient problem encountered when training traditional RNNs. LSTM is a special artificial RNN architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections, and due to this, it can process not only single data points (such as images) but also entire sequences of data (such as speech or video). A standard LSTM cell is shown in Figure 2 composed of a cell state sector, a forget gate, an input gate and an output gate. The cell remembers values over arbitrary time intervals, and the three gates regulate the flow of information into and out of the cell. Although LSTM is responsible for several models considered state-of-the-art in the literature, its performance depends heavily on the amount of available data and the choice of hyperparameters.

Figure 2 Standard LSTM Cell



III. DATASET

We use seven publicly available datasets to test the performance of the different prediction models. The statistics of these datasets are summarized in TABLE I.

TABLE I Units for Magnetic Properties*

Dataset	*m*	*T*	*L*	std	mean
Fly[19]	1	144	1 month	1.20×10^{2}	2.80×10^{2}
Bicycle[20]	16	833	1 hour	1.36×10^{2}	1.74×10^{2}
Exchange	1	1094	1 day	1.95×10^{-1}	6.67
SML2010[21]	24	2765	15 mins	2.85	1.92×10
Temperature[22]	1	3653	1 day	4.07	1.12×10
Solar Energy[23]	1	105121	5 mins	5.80×10	4.45×10
Electricity[24]	370	140256	15 mins	2.96×10^{16}	1.01×10^{17}

*The note *m* is the number of driving series. *T* is the length of the time series. *L* is the intervals of time series. Std is the standard deviation of the raw data. Mean is the average value of the target sequence.

We partition all datasets into training sets and test sets. The training set makes up the top 90% of the data set. The interpolation model is used to supplement the missing data in the datasets.

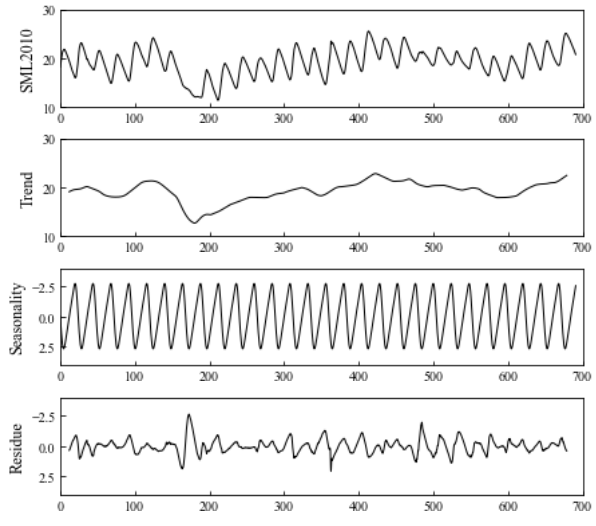
Figure 2(a) shows the target sequence in SML2010 and its three components including trend, seasonality and residue which are obtained through temperal data decomposition techniques. Trend *T* shows the long-term growth and decay of data. Seasonality *S* is the pattern of data cycles. Residue *Re* is short-term random fluctuations. In this research we define the residue of a period *p* of a series *Y*, as $Re_p = Y_p - (T_p + S_p)$ (with additive model) and $Re_p = Y_p / (T_p \times S_p)$ (with multiplicative model), respectively. Figure 2(b) shows the the serial correlation $R(k)$ for the residue. $R(k)$ is defined as follows, where n is the length of the series, X_t is the signals at the tth point, μ is the mean value of data, σ^2 is the variance of series. It gives the correlation of a series and a lagged version of itself. The smaller the autocorrelation coefficient of residue is, the more influenced raw data is by random factors that are not explicit. Therefore the mean of the absolute value of the amplitude of $R(k)$, which can be called $\overline{|A|}$, represent how stochastic the raw data is.

$$R(k) = \frac{\sum_{t=1}^{n-k}(X_t - \mu)(X_{t+k} - \mu)}{\sigma^2} \tag{5}$$

We can see in graph (a) of Figure 2, there are repetitive patterns in the stochastic series. Furthermore, we can observe a short-term daily pattern in the seasonality graph. In Figure 2(b), residue's autocorrelation suffers from attenuation. The decomposition properties of each dataset are summarized in

TABLE II.

(a)



(b)

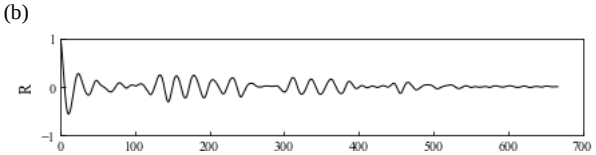


Figure 2. (a) Raw data and its three components;
(b) Serial residue correlation image of SML2010.

TABLE II DATASETS' DECOMPOSITION

Dataset	Term	Model	*p*	*k*	$\overline{\lvert A \rvert}$
Fly	Short	Multiplicable	12	131	0.094
Bicycle	Short	Additive	24	808	0.182
Exchange	Short	Additive	30	1062	0.049
SML2010	Medium	Additive	96	2667	0.073
Temperature	Medium	Additive	30	3621	0.012
Solar Energy	Long	Additive	288	104832	0.019
Electricity	Long	Additive	96	140159	0.007

IV. MODELOLOGY

A. Time Series Data and Its Prediction

Time series data is everywhere. As the real world gets increasingly instrumented, sensors and systems constantly emit a relentless stream of time series data applied across various industries. When the time series values could be synthesized by a mathematical function y=f(time), the series is classified as deterministic. In this work, we assume the researched time series data are discrete and deterministic.

Time series forecasting uses information regarding historical values and associated patterns to predict future activity. The Time series model involves original time-based data (years, days, hours, minutes, seconds) to derive hidden information. The time series prediction process generally covers six steps as follows.

The first step is to divide the time series into two sequences: the former part is intended for the model training, and another after that period, which is used to evaluate the quality of the fitted model. The second step chooses the predictive model according to data characteristics such as tendency, seasonality, and sequence length. The next step estimates the parameters of chosen models, the eventual prediction errors of the model reflect the effectiveness of the parameters, and such error could be amplified for a long time horizon. The final step is to evaluate the forecast results by comparing the predicted values to the test sequence, in which the evaluation indices are adopted to judge the predictive effects.

B. Predictive Performance

As mentioned in the previous subsection, the predictive data were compared with the test data using Root Mean Square Error (RMSE), correlation coefficient (R), and Prediction Of Change In Direction (POCID), where n is the sample size, $y_i, \hat{y}_i$ are the actual value and predicted value indexed with *i.*

RMSE is denoted by (6), where the error between y_i and $\hat{y}_i$. Its value is equal to the arithmetic square root of the quadratic sum of the prediction error divided by the number of observations.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{6}$$

The correlation coefficient in (7) measures how strong a relationship is between y_i and $\hat{y}_i$.

$$R = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}}_i)(y_i - \bar{y}_i)}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}}_i)^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}} \quad (7)$$

Another performance index considered is the POCID, which is formalized by (8). The term D_i acquires the value of 1 if $(\hat{y}_{i+1} - \hat{y}_i)(y_{i+1} - y_i) > 0$, and otherwise being 0. The purpose of POCID is to estimate the accuracy of direction's changes of the predictive data.

$$POCID = \frac{\sum_{i=1}^{n-1} D_i}{n\text{-}1} \quad (8)$$

It is a serious task to judge the best model by making a tradeoff for performance measurements. The Multi-Criteria Performance Measure (MCPM) developed in [25] are employed to combine the RMSE, R and POCID indices, in which the values of RMSE must be minimized, R and POCID must be maximized. The values of RMSE are normalized 0-1 and then reversed by one minus them, and the values of R and POCID are adopted to maximize the fitness.

A radar chart consisted of three axes, in which each one represents an individual performance measurement. The final value of MCPM is achieved by the area of each triangle. The lower values of MCPM correspond to better predictive performance for a specific model.

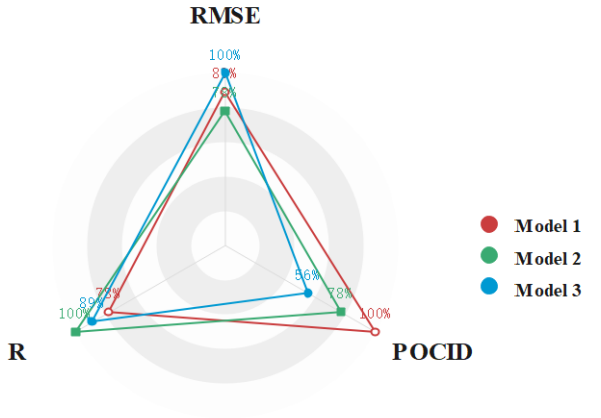


Figure 3. Multi-Criteria Performance Measure Radar Chart

V. Case Studies

The case studies contemplated the use of the programming languages MATLAB and Python, as well as their packages of functions for time series prediction.

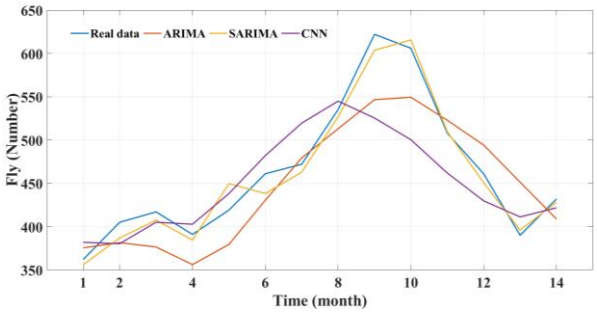
A. Prediction Results

The predictive results are calculated based on steps described in subsection IV.A. *Time Series Data and Its Prediction* for datasets illustrated in Section III. *Dataset*. The first 90% of all data sets are used for training, and the last 10% are used for prediction and verification. shows the nine models evaluated, as well as a description of their parameters and the range of values considered. All the codes and results are open-source and available on [26].

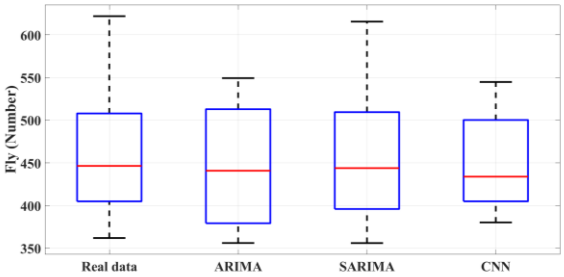
B. Overall Comparison

The comprehensive performance is fully considered by RMSE, R and POCID using MCPM. MCPM represents the overall performance of the model, and its value equals the area of each indices triangle. Several groups of models with the highest scores of MCPM in each dataset are selected to draw and observe the performance. It can be found in [26].Appendix for the specific evaluation indices of each model.

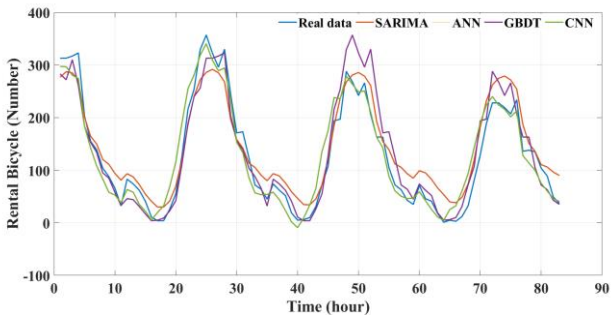
(a1)



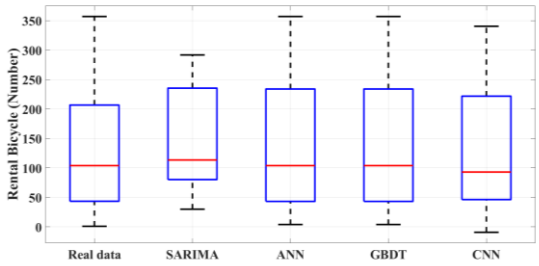
(a2)



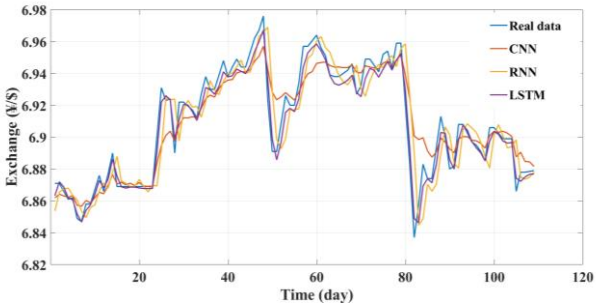
(b1)



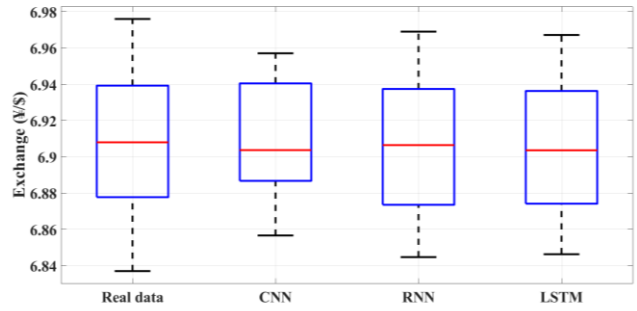
(b2)



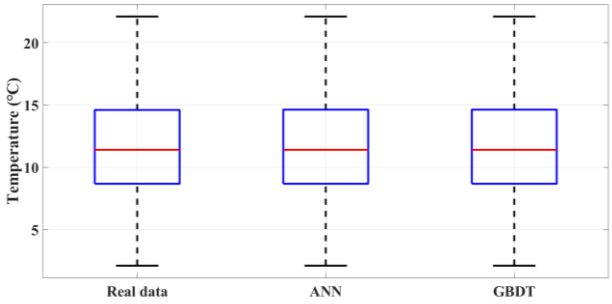
(c1)



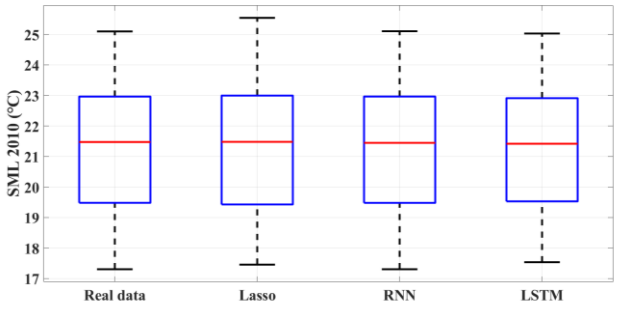
(c2)



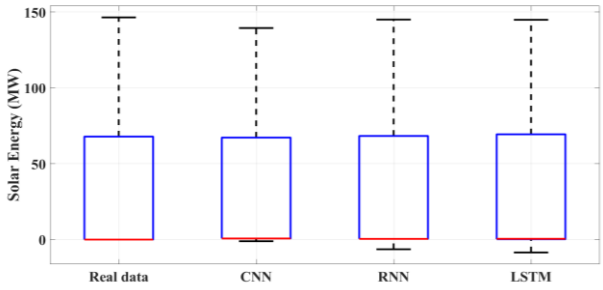
(d)



(e)



(f)



(g)

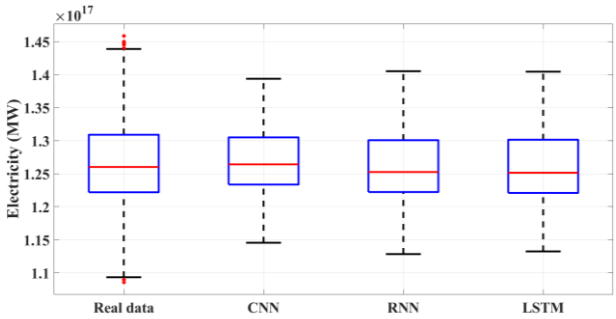


Figure 4. Comparison of most prominent models:
(a) Fly; (b) Bicycle; (c) Exchange;(d) SML2010;
(e) Temperature; (f) Solar energy; (g) Electricity.

C. *Discussions and Recommendations*

1) Statistical models

ARIMA is suitable for trend prediction. As shown in Figure 4 (a1). The increase in the number of dataset and test data is easy to lead to a straight line in the prediction results because the predict difference tends to be zero. Therefore, ARIMA can tell the trend of the dataset, but it cannot do the precise prediction. Using the ARIMA model to predict takes a short time because, with the help of ACF and PACF diagrams, we can quickly select (p, d, q) parameters.

SARIMA is suitable for the prediction of the dataset that contains a small amount of data, especially for the data with periodic changes (Bicycle). When the amount of data is small, parameters (p, d, q)×(P, D, Q, S) can be determined quickly. When the amount of data is large, the prediction requires high running memory and large amount of calculation. As a result, using SARIMA is time-consuming and difficult to get a precise prediction.

When Lasso forecasts dollar exchange rates, neither trends nor fluctuations fit correctly. When Lasso predicted multi-dimensional data, such as the Bicycle dataset, it does a good job of fitting trend and seasonality but did a poor job of randomness. In addition, when Lasso predicted SML2010, the first half of the prediction is almost perfect in Figure 4 (e), while the second half of the prediction is slightly inconsistent. Finally, when Lasso predicted Electricity, compared to machine learning and artificial intelligence algorithms, it takes only tens of seconds. The conclusion is that this model cannot predict chaotic data at all. For multi-dimensional time series, regardless of length, the stronger the autocorrelation is, the weaker the randomness is, the better the prediction effect is.

2) Machine Learning Models

The machine learning model is fitful for mid-term time series prediction (Figure 4 (d)), for the short training time and the high accuracy. And for ANN, since it needs appropriate networks, it is quite significant to choose the most effective layer units, optimizer and activation. If chosen well, its performance is quite good on account of elaborate training and forward feedback. In addition, the DT is also recommended if the depth is carefully set. It has the best performance and least training time for a reasonable dataset. In comparison, GDBT needs more time training and is easily overfit because it has actually too complicated calculation. So, it sometimes is able to deal with the dataset that the characters and the variables needed predicting are vaguely related or the cause and effect are not clearly known by human beings (Figure 4 (b), (d)). That is probably because it has a complex gradient equation to minimize the loss.

3) Deep Learning Models

For medium-term and long-term time series datasets, the line graph does not reflect the difference well, it may be better to adopt boxplots to observe the overall performance. Based on the predicted boxplots in Figure 4 (f)-(g), it can be seen that the deep learning models, especially LSTM, have the best prediction ability on the long-term prediction among nine models. Deep learning models denote a better prediction effect than statistical and machine learning algorithms due to the merits of extraction ability for the input feature vector.

TABLE III. MODELS AND THEIR PARAMETERS

Models	Parameters	Value
ARIMA	Order of the autocorrelation function (p)	p = 0: 1: t (t<2 times standard deviation)
	Degree of the differentiation operator (d)	d = 0: 1: 2
	Order of the partial autocorrelation function (q)	q = 0: 1: t (t<2 times standard deviation)
SARIMA	Order of the autoregression procedure (p)	p = 0: 1: $\sqrt{\log(m-h)}$
	Degree of the differentiation operator (d)	d = 0: 1: 2
	Order of the average moving procedure (q)	q = 0: 1: $\sqrt{\log(m-h)}$
	Order of the seasonal autoregression procedure (P)	P = 0:1: $\sqrt{\log(m-h)}$
	Degree of the seasonal differentiation operator (D)	D = 0: 1: 2
	Order of the seasonal moving average procedure (Q)	Q = 0: 1: $\sqrt{\log(m-h)}$
	Number of observations that make up a seasonal period (S)	S = 12
Lasso	Coefficient of the regularization term (α)	α = 0.1
ANN	Optimizer	Adam
	Activation function	Relu
	Learning rate	1e-5
DT	Depth	300
GBDT	Loss	Deviance
	Learning rate	0.005
	max depth	3
CNN	Convolution Layer Size	3*3
	ReluLayer Size	Relu
	Pooling Layer Size	2*2
	Dropout Layer Size	20%
	Optimizer	Adam
	MiniBatchSize	8
	MaxEpochs	20-300
	Initial Learn Rate	0.005
	LearnRateDrop Factor	0.01
	LearnRate Drop Period	20
	Validation Frequency	1000
RNN	trainfunction	Traingdx
	trainParam.epochs	20-300
	trainParam.goal	0.00001
	trainParam.max_fail	5
LSTM	MaxEpochs	20-300
	GradientThreshold	1
	InitialLearnRate	0.005
	LearnRateDropPeriod	125
	LearnRateDropFactor	0.2

In a nutshell, Deep learning models overcome the shortcomings of statistical and machine learning algorithms, such as the time series dependencies that cannot be described, weak data feature learning ability, and is prone to overfitting. Therefore, they indicate better prediction effects for capturing the non-linear dynamic characteristics of time series data, which is more accurate for datasets with non-stationary and time-dependent characteristics.

Overall, the conclusions and recommendations are summarized as follows:

a). For short-term time series prediction for datasets with obvious trends and seasonalities such as Bicycle and Exchange, it is recommended that ARIMA or SARIMA could be adopted. The reason traced back to the algorithm principles of the traditional statistical models, and explicit functions are adopted to achieve parameters and build the statistical model, thus avoid overfitting and elaborate calculation.

b). For medium-term time series prediction such as SML2010 and Temperature, ANN and GBDT are recommended replacing statistical models. It is worth mentioning that the Lasso model performs excellent for mid-term forecasts with the stationary property.

c). For long-term time series prediction, it is recommended that a deep learning model should be used. The deep learning model has strong potential to deal with large-scale datasets due to the nerual network embedded.

TABLE IV suggested the potential prediction models for diverse datasets according to time interval, volatility, trend, and seasonality.

TABLE IV. MODEL RECOMMENDATIONS

Dataset	$\overline{\lvert A\rvert}$>0.09	0.05<$\overline{\lvert A\rvert}$<0.09	$\overline{\lvert A\rvert}$<0.05	Trend	Seasonality
Short-term	ARIMA, SARIMA	-	CNN, RNN, LSTM	ARIMA	SARIMA
Medium-term	-	LASSO, ANN, DT	CNN, RNN, LSTM	ANN, DT	ANN, DT
Long-term	-	-	CNN, RNN, LSTM	CNN	RNN, LSTM

- N/A

VI. Conclusion

The paper aims to answer questions that whether complex prediction models are needed and how to select appropriate models based on data characteristics and prediction tasks. We empirically compared the performance of three categories of prediction models (statistical, machine learning and deep learning) using seven time series datasets to answer these questions. The findings show that the model prediction performance varies depending on prediction horizon, as well as time interval, volatility, trend and seasonality of the dataset. Case studies show that the statistical models perform better for datasets with low stochasticity and high trend and seasonality; machine learning models work outstandingly on coping with medium-term time-series data prediction; As for deep learning models, they specialise in forecasting fluctuant time series data due to their information extraction abilities of neural network structure. The results can guide the practitioner in selecting appropriate models and thus the research community in focusing the effort to more feasible or promising directions.

The future work will focus on developing a more scientific evaluation system. Moreover, the connection between the result and the model structure is also worth researching.

References

[1] De Gooijer, Jan G., and Rob J. Hyndman. "25 years of time series forecasting." International journal of forecasting 22.3 (2006): 443-473.

[2] Kolmogorov, Andreĭ Nikolaevich. Stationary sequences in Hilbert space. John Crerar Library] National Translations Center, 1978.

[3] Hamilton, James Douglas. Time series analysis. Princeton university press, 2020.

[4] Ahmed, Nesreen K., et al. "An empirical comparison of machine learning models for time series forecasting." Econometric Reviews 29.5-6 (2010): 594-621.

[5] Yin, Chungen, Lasse Rosendahl, and Zhongyang Luo. "Models to improve prediction performance of ANN models." Simulation Modelling Practice and Theory 11.3-4 (2003): 211-222.

[6] Sapankevych, Nicholas I., and Ravi Sankar. "Time series prediction using support vector machines: a survey." IEEE Computational Intelligence Magazine 4.2 (2009): 24-38.

[7] Kane, Michael J., et al. "Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks." BMC bioinformatics 15.1 (2014): 1-9.

[8] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436-444.

[9] Hoseinzade, Ehsan, and Saman Haratizadeh. "CNNpred: CNN-based stock market prediction using a diverse set of variables." Expert Systems with Applications 129 (2019): 273-285.

[10] Zhang, Jia-Shu, and Xian-Ci Xiao. "Predicting chaotic time series using recurrent neural network." Chinese Physics Letters 17.2 (2000): 88.

[11] Li, Chaoshun, et al. "Short-term wind speed interval prediction based on ensemble GRU model." IEEE transactions on sustainable energy 11.3 (2019): 1370-1380.

[12] Hua, Yuxiu, et al. "Deep learning with long short-term memory for time series prediction." IEEE Communications Magazine 57.6 (2019): 114-119.

[13] Du, Shengdong, et al. "Deep air quality forecasting using hybrid deep learning framework." IEEE Transactions on Knowledge and Data Engineering (2019).

[14] Chen, Jie, et al. "Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization." Energy conversion and management 165 (2018): 681-695.

[15] Faruk, Durdu Ömer. "A hybrid neural network and ARIMA model for water quality time series prediction." Engineering applications of artificial intelligence 23.4 (2010): 586-594.

[16] Eswaran, C., and R. Logeswaran. "An enhanced hybrid model for time series prediction using linear and neural network models." Applied Intelligence 37.4 (2012): 511-519.

[17] Cai, Xindi, et al. "Time series prediction with recurrent neural networks trained by a hybrid PSO–EA algorithm." Neurocomputing 70.13-15 (2007): 2342-2353.

[18] Lu, Wenjie, et al. "A CNN-BiLSTM-AM model for stock price prediction." Neural Computing and Applications 33.10 (2021): 4741-4753.

[19] https://max.book118.com/html/2017/0507/105050278.shtm

[20] H. Fanaee, "Bike sharing dataset data set," University of Porto, December, 2013. Accessed on: August, 2021. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset

[21] F. Zamora et al., "On-line learning of indoor temperature forecasting models towards energy efficiency," Energy and Buildings, vol 83, pp. 162-172, Nov. 2014.

[22] GitHub. 2021. Datasets/daily-min-temperatures.names at master · jbrownlee/Datasets. [online] Available at: <https://github.com/jbrownlee/Datasets/blob/master/daily-min-temperatures.names> [Accessed 5 September 2021].

[23] Y. C. Zhang, "Solar power data for integration studies," NREL, 2016. Accessed on: August, 2021. [Online]. Available. https://www.nrel.gov/grid/assets/downloads/fl-pv-2006.zip

[24] A. Trindade, "Electricityloaddiagrams20112014 data set," Elergone, August 2015. Accessed on: August, 2021. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

[25] Parmezan A R S, Lee H D, Wu F C. Metalearning for choosing feature selection algorithms in data mining: Proposal of a new framework[J]. Expert Systems with Applications, 2017, 75: 1-24.

[26] "Github - MITAI-Lyft/Codes: Statistical Model, Machine Learning Model And Deep Learning Model". Github, 2021, https://github.com/MITAI-Lyft/Codes.git.